

LinuxQMI SDK-Lite
SLQS04.00.23

Generated by Doxygen 1.8.6

Thu Nov 21 2019 08:29:34

Contents

1	Welcome to the Sierra Wireless Linux QMI SDK API Reference Guide	1
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
2	Module Index	3
2.1	Modules	3
3	Namespace Index	5
3.1	Namespace List	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	29
5.1	File List	29
6	Module Documentation	31
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
7	Namespace Documentation	33
7.1	Tables Namespace Reference	33
7.1.1	Detailed Description	33
8	Data Structure Documentation	35
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	persoState	41
8.5.2.8	persoUnblockRetries	42
8.5.2.9	pin1Retries	42
8.5.2.10	pin1State	42
8.5.2.11	pin2Retries	42
8.5.2.12	pin2State	42
8.5.2.13	puk1Retries	42
8.5.2.14	puk2Retries	42
8.5.2.15	univPin	42
8.6	audio_RXAGCList Struct Reference	42
8.6.1	Detailed Description	42
8.6.2	Field Documentation	42
8.6.2.1	pRXAIG	43
8.6.2.2	pRXComprSlope	43
8.6.2.3	pRXComprThres	43
8.6.2.4	pRXExpSlope	43
8.6.2.5	pRXExpThres	43
8.6.2.6	pRXStaticGain	43
8.7	audio_RXAVCList Struct Reference	43
8.7.1	Detailed Description	43
8.7.2	Field Documentation	43
8.7.2.1	pAVRXAVCHeadroom	43
8.7.2.2	pAVRXAVCSens	43
8.8	audio_RXPCMIIRFtr Struct Reference	43
8.8.1	Detailed Description	43
8.8.2	Field Documentation	45
8.8.2.1	pFlag	45
8.8.2.2	pStage0Val	45
8.8.2.3	pStage1Val	45
8.8.2.4	pStage2Val	45
8.8.2.5	pStage3Val	45
8.8.2.6	pStage4Val	45
8.8.2.7	pStageCnt	45
8.9	audio_TXAGCList Struct Reference	45
8.9.1	Detailed Description	45
8.9.2	Field Documentation	45
8.9.2.1	pTXAIG	46
8.9.2.2	pTXComprSlope	46
8.9.2.3	pTXComprThres	46
8.9.2.4	pTXExpSlope	46

8.9.2.5	pTXExpThres	46
8.9.2.6	pTXStaticGain	46
8.10	audio_TXPCMIIRFtr 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	CatAlphaIdtfr	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
8.109.2.11pIPv6AddrPref	112
8.109.2.12pPassword	112
8.109.2.13pPasswordSize	112
8.109.2.14pPcscfAddrUsingDhcp	112
8.109.2.15pPcscfAddrUsingPCO	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 pIPv6AddPref	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_BdsSV Struct Reference	149
8.121.1 Detailed Description	149
8.121.2 Field Documentation	149
8.121.2.1 id	149
8.121.2.2 mask	149
8.122loc_BdsSVInfo Struct Reference	149
8.122.1 Detailed Description	149
8.122.2 Field Documentation	150

8.122.2.1 len	150
8.122.2.2 pSV	150
8.123loc_CellDb Struct Reference	150
8.123.1 Detailed Description	150
8.123.2 Field Documentation	150
8.123.2.1 mask	150
8.124loc_ClkInfo Struct Reference	150
8.124.1 Detailed Description	151
8.124.2 Field Documentation	151
8.124.2.1 mask	152
8.125loc_GnssData Struct Reference	152
8.125.1 Detailed Description	152
8.125.2 Field Documentation	153
8.125.2.1 mask	153
8.126loc_gpsTime Struct Reference	153
8.126.1 Detailed Description	154
8.126.2 Field Documentation	154
8.126.2.1 gpsTimeOfWeekMs	154
8.126.2.2 gpsWeek	154
8.127loc_gyroAcceptReady Struct Reference	154
8.127.1 Detailed Description	154
8.127.2 Field Documentation	155
8.127.2.1 batchPerSec	155
8.127.2.2 injectEnable	155
8.127.2.3 samplesPerBatch	155
8.128loc_gyroTempAcceptReady Struct Reference	155
8.128.1 Detailed Description	155
8.128.2 Field Documentation	155
8.128.2.1 batchPerSec	155
8.128.2.2 injectEnable	155
8.128.2.3 samplesPerBatch	156
8.129loc_IPv4Config Struct Reference	156
8.129.1 Detailed Description	156
8.129.2 Field Documentation	156
8.129.2.1 IPv4Addr	156
8.129.2.2 IPv4Port	156
8.130loc_IPv4Info Struct Reference	156
8.130.1 Detailed Description	156
8.130.2 Field Documentation	157
8.130.2.1 address	157

8.130.2.2 port	157
8.130.2.3 TlvPresent	157
8.131 loc_IPv6Config Struct Reference	157
8.131.1 Detailed Description	157
8.131.2 Field Documentation	157
8.131.2.1 IPv6Addr	157
8.131.2.2 IPv6Port	157
8.132 loc_IPv6Info Struct Reference	157
8.132.1 Detailed Description	157
8.132.2 Field Documentation	158
8.132.2.1 address	158
8.132.2.2 port	158
8.132.2.3 TlvPresent	158
8.133 loc_LocApplicationInfo Struct Reference	158
8.133.1 Detailed Description	158
8.133.2 Field Documentation	159
8.133.2.1 appNameLength	159
8.133.2.2 appProviderLength	159
8.133.2.3 appVersionLength	159
8.133.2.4 appVersionValid	159
8.133.2.5 pAppName	159
8.133.2.6 pAppProvider	159
8.133.2.7 pAppVersion	159
8.134 loc_precisionDilution Struct Reference	159
8.134.1 Detailed Description	159
8.134.2 Field Documentation	160
8.134.2.1 HDOP	160
8.134.2.2 PDOP	160
8.134.2.3 VDOP	160
8.135 loc_satelliteInfo Struct Reference	160
8.135.1 Detailed Description	160
8.135.2 Field Documentation	162
8.135.2.1 azimuth	162
8.135.2.2 elevation	162
8.135.2.3 gnssSvId	162
8.135.2.4 healthStatus	162
8.135.2.5 snr	162
8.135.2.6 svInfoMask	162
8.135.2.7 svListLen	162
8.135.2.8 svStatus	162

8.135.2.9 system	162
8.135.2.10 validMask	162
8.136 loc_sensorDataUsage Struct Reference	162
8.136.1 Detailed Description	162
8.136.2 Field Documentation	163
8.136.2.1 aidingIndicatorMask	163
8.136.2.2 usageMask	163
8.137 loc_SV Struct Reference	163
8.137.1 Detailed Description	163
8.137.2 Field Documentation	164
8.137.2.1 id	164
8.137.2.2 mask	164
8.137.2.3 system	164
8.138 loc_SVInfo Struct Reference	164
8.138.1 Detailed Description	164
8.138.2 Field Documentation	164
8.138.2.1 len	164
8.138.2.2 pSV	164
8.139 loc_svUsedforFix Struct Reference	164
8.139.1 Detailed Description	164
8.139.2 Field Documentation	165
8.139.2.1 gnssSvUsedList	165
8.139.2.2 gnssSvUsedList_len	165
8.140 loc_urlAddr Struct Reference	165
8.140.1 Detailed Description	165
8.140.2 Field Documentation	165
8.140.2.1 address	165
8.140.2.2 TlvPresent	165
8.141 loc_URLAddrInfo Struct Reference	165
8.141.1 Detailed Description	166
8.141.2 Field Documentation	166
8.141.2.1 urlAddr	166
8.142 lteSSInfo Struct Reference	166
8.142.1 Detailed Description	166
8.142.2 Field Documentation	166
8.142.2.1 rsrp	167
8.142.2.2 rsrq	167
8.142.2.3 rssi	167
8.142.2.4 snr	167
8.143 messageModeTlv Struct Reference	167

8.143.1 Detailed Description	167
8.143.2 Field Documentation	167
8.143.2.1 MessageModelInfo	167
8.143.2.2 TlvPresent	167
8.144nas_acqOrderPref Struct Reference	167
8.144.1 Detailed Description	167
8.144.2 Field Documentation	168
8.144.2.1 acqOrdeLen	168
8.144.2.2 pAcqOrder	168
8.145nas_AcqOrderPrefTlv Struct Reference	168
8.145.1 Detailed Description	168
8.145.2 Field Documentation	168
8.145.2.1 acqOrdeLen	168
8.145.2.2 pAcqOrder	169
8.145.2.3 TlvPresent	169
8.146nas_ActPilotPNElement Struct Reference	169
8.146.1 Detailed Description	169
8.146.2 Field Documentation	169
8.146.2.1 ActSetPilotPN	169
8.146.2.2 ActSetPilotPNStrength	169
8.147nas_AddCDMASysInfo Struct Reference	169
8.147.1 Detailed Description	169
8.147.2 Field Documentation	170
8.147.2.1 geoSysIdx	170
8.147.2.2 regPrd	170
8.148nas_AddSysInfo Struct Reference	170
8.148.1 Detailed Description	170
8.148.2 Field Documentation	170
8.148.2.1 cellBroadcastCap	170
8.148.2.2 geoSysIdx	170
8.149nas_BandPrefInfoTlv Struct Reference	170
8.149.1 Detailed Description	170
8.149.2 Field Documentation	171
8.149.2.1 bits_129_192	171
8.149.2.2 bits_193_256	171
8.149.2.3 bits_1_64	171
8.149.2.4 bits_65_128	171
8.149.2.5 TlvPresent	171
8.150nas_BandPrefTlv Struct Reference	171
8.150.1 Detailed Description	171

8.150.2 Field Documentation	173
8.150.2.1 BandPref	173
8.150.2.2 TlvPresent	173
8.151nas_CallBarringSysInfo Struct Reference	173
8.151.1 Detailed Description	173
8.151.2 Field Documentation	173
8.151.2.1 csBarStatus	173
8.151.2.2 psBarStatus	173
8.152nas_callBarStatus Struct Reference	173
8.152.1 Detailed Description	174
8.152.2 Field Documentation	174
8.152.2.1 csBarStatus	174
8.152.2.2 psBarStatus	174
8.153nas_CDMAChannel Struct Reference	174
8.153.1 Detailed Description	174
8.153.2 Field Documentation	175
8.153.2.1 priChA	175
8.153.2.2 priChB	175
8.153.2.3 secChA	175
8.153.2.4 secChB	175
8.154nas_CDMAECIOThresh Struct Reference	175
8.154.1 Detailed Description	175
8.154.2 Field Documentation	175
8.154.2.1 CDMAECIOThreshListLen	175
8.154.2.2 pCDMAECIOThreshList	176
8.155nas_CDMAInfo Struct Reference	176
8.155.1 Detailed Description	176
8.155.2 Field Documentation	176
8.155.2.1 baselId	176
8.155.2.2 baseLat	176
8.155.2.3 baseLong	176
8.155.2.4 nid	177
8.155.2.5 refpn	177
8.155.2.6 sid	177
8.156nas_CDMARSSIThresh Struct Reference	177
8.156.1 Detailed Description	177
8.156.2 Field Documentation	177
8.156.2.1 CDMARSSIThreshListLen	177
8.156.2.2 pCDMARSSIThreshList	177
8.157nas_CDMA SysInfo Struct Reference	177

8.157.1 Detailed Description	178
8.157.2 Field Documentation	180
8.157.2.1 baseId	180
8.157.2.2 baseLat	180
8.157.2.3 baseLong	180
8.157.2.4 bsInfoValid	180
8.157.2.5 bsPRev	180
8.157.2.6 bsPRevValid	180
8.157.2.7 ccsSupported	180
8.157.2.8 ccsSupportedValid	180
8.157.2.9 cdmaSysIdValid	180
8.157.2.10 isSysPrIMatch	180
8.157.2.11 isSysPrIMatchValid	180
8.157.2.12 MCC	180
8.157.2.13 MNC	180
8.157.2.14 networkId	180
8.157.2.15 networkIdValid	181
8.157.2.16 packetZone	181
8.157.2.17 packetZoneValid	181
8.157.2.18 pRevInUse	181
8.157.2.19 pRevInUseValid	181
8.157.2.20 sysInfoCDMA	181
8.157.2.21 systemID	181
8.158 nas_CDMASysInfoExt Struct Reference	181
8.158.1 Detailed Description	181
8.158.2 Field Documentation	181
8.158.2.1 imsi_11_12	181
8.158.2.2 MCC	181
8.159 nas_cellParams Struct Reference	181
8.159.1 Detailed Description	182
8.159.2 Field Documentation	182
8.159.2.1 pci	182
8.159.2.2 rsrp	182
8.159.2.3 rsrq	182
8.159.2.4 rssi	182
8.159.2.5 srxlev	182
8.160 nas_ciotAcqOrderPref Struct Reference	182
8.160.1 Detailed Description	182
8.160.2 Field Documentation	183
8.160.2.1 ciotAcqOrderLen	183

8.160.2.2 pCiotAcqOrder	183
8.161nas_CiotAcqOrderPrefTlv Struct Reference	183
8.161.1 Detailed Description	183
8.161.2 Field Documentation	184
8.161.2.1 ciotAcqOrderLen	184
8.161.2.2 pCiotAcqOrder	184
8.161.2.3 TlvPresent	184
8.162nas_CiotLteOpModePrefTlv Struct Reference	184
8.162.1 Detailed Description	184
8.162.2 Field Documentation	184
8.162.2.1 ciotLteOpModePref	184
8.162.2.2 TlvPresent	184
8.163nas_CommInfo Struct Reference	184
8.163.1 Detailed Description	185
8.163.2 Field Documentation	186
8.163.2.1 imsRegState	186
8.163.2.2 modemMode	186
8.163.2.3 psState	186
8.163.2.4 systemMode	186
8.163.2.5 temperature	186
8.164nas_CSGID Struct Reference	186
8.164.1 Detailed Description	186
8.164.2 Field Documentation	187
8.164.2.1 id	187
8.164.2.2 mcc	187
8.164.2.3 mnc	187
8.164.2.4 mncPcsDigits	187
8.164.2.5 rat	187
8.165nas_Csgld Struct Reference	187
8.165.1 Detailed Description	187
8.165.2 Field Documentation	187
8.165.2.1 csgld	187
8.165.2.2 TlvPresent	187
8.166nas_currentPLMN Struct Reference	187
8.166.1 Detailed Description	188
8.166.2 Field Documentation	188
8.166.2.1 MCC	188
8.166.2.2 MNC	188
8.166.2.3 netDescr	188
8.166.2.4 netDescrLength	188

8.167nas_dataSrvCapabilities Struct Reference	188
8.167.1 Detailed Description	188
8.167.2 Field Documentation	189
8.167.2.1 dataCapabilities	189
8.167.2.2 dataCapabilitiesLen	189
8.168nas_DataStatusDetail Struct Reference	189
8.168.1 Detailed Description	189
8.168.2 Field Documentation	190
8.168.2.1 IPAddress	190
8.168.2.2 LastErrCode	190
8.169nas_detailSvcInfo Struct Reference	191
8.169.1 Detailed Description	191
8.169.2 Field Documentation	192
8.169.2.1 hdrHybrid	192
8.169.2.2 hdrSrvStatus	192
8.169.2.3 isSysForbidden	192
8.169.2.4 srvCapability	192
8.169.2.5 srvStatus	192
8.170nas_DeviceConfigDetail Struct Reference	192
8.170.1 Detailed Description	192
8.170.2 Field Documentation	193
8.170.2.1 Chipset	193
8.170.2.2 HWVersion	193
8.170.2.3 QLIC	193
8.170.2.4 Technology	193
8.171nas_dirNum Struct Reference	193
8.171.1 Detailed Description	193
8.171.2 Field Documentation	193
8.171.2.1 dirNum	193
8.171.2.2 dirNumLen	193
8.172nas_DRCPParams Struct Reference	193
8.172.1 Detailed Description	194
8.172.2 Field Documentation	194
8.172.2.1 DRCCover	194
8.172.2.2 DRCValue	194
8.173nas_ecioListElement Struct Reference	194
8.173.1 Detailed Description	194
8.173.2 Field Documentation	194
8.173.2.1 ecio	195
8.173.2.2 radiolf	195

8.174nas_ECIOThresh Struct Reference	195
8.174.1 Detailed Description	195
8.174.2 Field Documentation	195
8.174.2.1 ECIOThresListLen	195
8.174.2.2 pECIOThresList	195
8.175nas_EdrxCiotLteMode Struct Reference	195
8.175.1 Detailed Description	195
8.175.2 Field Documentation	196
8.175.2.1 lteOpMode	196
8.175.2.2 TlvPresent	196
8.176nas_EdrxCycleLength Struct Reference	196
8.176.1 Detailed Description	196
8.176.2 Field Documentation	196
8.176.2.1 cycleLength	196
8.176.2.2 TlvPresent	196
8.177nas_EdrxEnableType Struct Reference	196
8.177.1 Detailed Description	197
8.177.2 Field Documentation	197
8.177.2.1 edrxEnabled	197
8.177.2.2 TlvPresent	197
8.178nas_EdrxPagingTimeWindow Struct Reference	197
8.178.1 Detailed Description	197
8.178.2 Field Documentation	197
8.178.2.1 edrxPtw	197
8.178.2.2 TlvPresent	197
8.179nas_EdrxRatType Struct Reference	197
8.179.1 Detailed Description	198
8.179.2 Field Documentation	198
8.179.2.1 edrxRatType	198
8.179.2.2 TlvPresent	198
8.180nas_EmerModeTlv Struct Reference	198
8.180.1 Detailed Description	198
8.180.2 Field Documentation	199
8.180.2.1 EmerMode	199
8.180.2.2 TlvPresent	199
8.181nas_errorRateListElement Struct Reference	199
8.181.1 Detailed Description	199
8.181.2 Field Documentation	200
8.181.2.1 errorRate	200
8.181.2.2 radiolf	200

8.182nas_ForbiddenNetworks3GPP Struct Reference	200
8.182.1 Detailed Description	200
8.182.2 Field Documentation	201
8.182.2.1 forbiddenNwInstLen	201
8.182.2.2 MCC	201
8.182.2.3 MNC	201
8.182.2.4 TlvPresent	201
8.183nas_GERANInfo Struct Reference	201
8.183.1 Detailed Description	201
8.183.2 Field Documentation	202
8.183.2.1 arfcn	202
8.183.2.2 bsic	202
8.183.2.3 cellID	202
8.183.2.4 insNmrCellInfo	202
8.183.2.5 lac	202
8.183.2.6 nmrlnst	202
8.183.2.7 plmn	202
8.183.2.8 rxLev	202
8.183.2.9 timingAdvance	202
8.184nas_geranInstInfo Struct Reference	202
8.184.1 Detailed Description	203
8.184.2 Field Documentation	203
8.184.2.1 geranArfcn	203
8.184.2.2 geranBsicBcc	203
8.184.2.3 geranBsicNcc	203
8.184.2.4 geranRssi	203
8.185nas_gsmCellInfo Struct Reference	203
8.185.1 Detailed Description	203
8.185.2 Field Documentation	204
8.185.2.1 arfcn	204
8.185.2.2 band1900	204
8.185.2.3 bsicld	204
8.185.2.4 cellldValid	204
8.185.2.5 rssi	204
8.185.2.6 srxlev	204
8.186nas_GSMRSSIthresh Struct Reference	204
8.186.1 Detailed Description	204
8.186.2 Field Documentation	205
8.186.2.1 GSMRSSIthreshListLen	205
8.186.2.2 pGSMRSSIthreshList	205

8.187nas_GSMSrvStatusInfo Struct Reference	205
8.187.1 Detailed Description	205
8.187.2 Field Documentation	206
8.187.2.1 isPrefDataPath	206
8.187.2.2 srvStatus	206
8.187.2.3 trueSrvStatus	206
8.188nas_GSMSysInfo Struct Reference	206
8.188.1 Detailed Description	206
8.188.2 Field Documentation	208
8.188.2.1 cellId	208
8.188.2.2 cellIdValid	208
8.188.2.3 dtmSupp	208
8.188.2.4 dtmSuppValid	208
8.188.2.5 egprsSupp	208
8.188.2.6 egprsSuppValid	208
8.188.2.7 lac	208
8.188.2.8 lacValid	208
8.188.2.9 MCC	208
8.188.2.10MNC	208
8.188.2.11networkIdValid	208
8.188.2.12regRejectInfoValid	208
8.188.2.13rejCause	209
8.188.2.14rejectSrvDomain	209
8.188.2.15sysInfoGSM	209
8.189nas_GWAcqOrderPrefTlv Struct Reference	209
8.189.1 Detailed Description	209
8.189.2 Field Documentation	209
8.189.2.1 GWAcqOrderPref	209
8.189.2.2 TlvPresent	209
8.190nas_HDRECIOThresh Struct Reference	209
8.190.1 Detailed Description	209
8.190.2 Field Documentation	210
8.190.2.1 HDRECIOThreshListLen	210
8.190.2.2 pHDRECIOThreshList	210
8.191nas_HDRIOTThresh Struct Reference	210
8.191.1 Detailed Description	210
8.191.2 Field Documentation	210
8.191.2.1 HDRIOTThreshListLen	210
8.191.2.2 pHDRIOTThreshList	210
8.192nas_HDRPersonality_Ind_Data Struct Reference	210

8.192.1 Detailed Description	211
8.192.2 Field Documentation	211
8.192.2.1 pCurrentPersonality	211
8.192.2.2 pPersonalityListLength	211
8.192.2.3 pProtocolSubtypeElement	211
8.193nas_HDRRSSIThresh Struct Reference	211
8.193.1 Detailed Description	211
8.193.2 Field Documentation	211
8.193.2.1 HDRRSSIThreshListLen	211
8.193.2.2 pHDRRSSIThreshList	212
8.194nas_HDRSINRThresh Struct Reference	212
8.194.1 Detailed Description	212
8.194.2 Field Documentation	212
8.194.2.1 HDRSINRThresListLen	212
8.194.2.2 pHDRSINRThresList	212
8.195nas_HDRSINRThreshold Struct Reference	212
8.195.1 Detailed Description	212
8.195.2 Field Documentation	213
8.195.2.1 HDRSINRThreshListLen	213
8.195.2.2 pHDRSINRThreshList	213
8.196nas_HDRSysInfo Struct Reference	213
8.196.1 Detailed Description	213
8.196.2 Field Documentation	215
8.196.2.1 hdrActiveProt	215
8.196.2.2 hdrActiveProtValid	215
8.196.2.3 hdrPersonality	215
8.196.2.4 hdrPersonalityValid	215
8.196.2.5 is856SysId	215
8.196.2.6 is856SysIdValid	215
8.196.2.7 isSysPrIMatch	215
8.196.2.8 isSysPrIMatchValid	215
8.196.2.9 sysInfoHDR	215
8.197nas_homeSIDNID Struct Reference	215
8.197.1 Detailed Description	215
8.197.2 Field Documentation	215
8.197.2.1 numInstances	215
8.197.2.2 SidNid	215
8.198nas_infoInterFreq Struct Reference	215
8.198.1 Detailed Description	216
8.198.2 Field Documentation	216

8.198.2.1 cell_resel_priority	216
8.198.2.2 cellInterFreqParams	216
8.198.2.3 cells_len	216
8.198.2.4 earfcn	216
8.198.2.5 threshXHigh	217
8.198.2.6 threshXLow	217
8.199nas_IOTresh Struct Reference	217
8.199.1 Detailed Description	217
8.199.2 Field Documentation	217
8.199.2.1 IOTreshListLen	217
8.199.2.2 pIOTreshList	217
8.200nas_lteBandPrefExt Struct Reference	217
8.200.1 Detailed Description	217
8.200.2 Field Documentation	218
8.200.2.1 bits_129_192	218
8.200.2.2 bits_193_256	218
8.200.2.3 bits_1_64	218
8.200.2.4 bits_65_128	218
8.201nas_LTEBandPrefTlv Struct Reference	218
8.201.1 Detailed Description	218
8.201.2 Field Documentation	219
8.201.2.1 LTEBandPref	220
8.201.2.2 TlvPresent	220
8.202nas_LteCiotOpModeTlv Struct Reference	220
8.202.1 Detailed Description	220
8.202.2 Field Documentation	220
8.202.2.1 campedCiotLteOpMode	220
8.202.2.2 TlvPresent	220
8.203nas_lteEARFCN Struct Reference	220
8.203.1 Detailed Description	220
8.203.2 Field Documentation	221
8.203.2.1 earfcn0	221
8.203.2.2 earfcn1	221
8.203.2.3 status	221
8.204nas_lteGsmCellInfo Struct Reference	221
8.204.1 Detailed Description	221
8.204.2 Field Documentation	222
8.204.2.1 cellReselPriority	222
8.204.2.2 cells_len	222
8.204.2.3 GsmCellInfo	222

8.204.2.4 nccPermitted	222
8.204.2.5 threshGsmHigh	222
8.204.2.6 threshGsmLow	222
8.205nas_LTEInfo Struct Reference	222
8.205.1 Detailed Description	222
8.205.2 Field Documentation	224
8.205.2.1 band	224
8.205.2.2 bandwidth	224
8.205.2.3 emmConnState	224
8.205.2.4 emmState	224
8.205.2.5 emmSubState	224
8.205.2.6 RXChan	224
8.205.2.7 TXChan	224
8.206nas_LTEInfoInterfreq Struct Reference	225
8.206.1 Detailed Description	225
8.206.2 Field Documentation	225
8.206.2.1 freqsLen	225
8.206.2.2 InfoInterfreq	225
8.206.2.3 ueInIdle	225
8.207nas_LTEInfoIntrafreq Struct Reference	225
8.207.1 Detailed Description	226
8.207.2 Field Documentation	227
8.207.2.1 CellParams	227
8.207.2.2 cellReselPriority	227
8.207.2.3 cellsLen	227
8.207.2.4 earfcn	227
8.207.2.5 globalCellId	227
8.207.2.6 plmn	227
8.207.2.7 servingCellId	227
8.207.2.8 sIntraSearch	227
8.207.2.9 sNonIntraSearch	227
8.207.2.10tac	227
8.207.2.11threshServingLow	227
8.207.2.12ueInIdle	227
8.208nas_LTEInfoNeighboringGSM Struct Reference	227
8.208.1 Detailed Description	228
8.208.2 Field Documentation	228
8.208.2.1 freqsLen	228
8.208.2.2 LteGsmCellInfo	228
8.208.2.3 ueInIdle	228

8.209nas_LTEInfoNeighboringWCDMA Struct Reference	228
8.209.1 Detailed Description	228
8.209.2 Field Documentation	229
8.209.2.1 freqsLen	229
8.209.2.2 LTEWCDMACellInfo	229
8.209.2.3 ueInIdle	229
8.210nas_LteM1BandPrefTlv Struct Reference	229
8.210.1 Detailed Description	229
8.210.2 Field Documentation	230
8.210.2.1 lteM1BandPref	231
8.210.2.2 TlvPresent	231
8.211nas_LteNb1BandPrefTlv Struct Reference	231
8.211.1 Detailed Description	231
8.211.2 Field Documentation	232
8.211.2.1 lteNb1BandPref	233
8.211.2.2 TlvPresent	233
8.212nas_LTEOperationMode Struct Reference	233
8.212.1 Detailed Description	233
8.212.2 Field Documentation	233
8.212.2.1 pLTEOperationMode	233
8.212.2.2 TlvPresent	233
8.213nas_LteOpMode Struct Reference	233
8.213.1 Detailed Description	233
8.213.2 Field Documentation	234
8.213.2.1 lteOpMode	234
8.213.2.2 TlvPresent	234
8.214nas_LteOpModeTlv Struct Reference	234
8.214.1 Detailed Description	234
8.214.2 Field Documentation	235
8.214.2.1 lteOpMode	235
8.214.2.2 lteOpModeLen	235
8.214.2.3 MCC	235
8.214.2.4 MNC	235
8.214.2.5 TlvPresent	235
8.215nas_LtePCI Struct Reference	235
8.215.1 Detailed Description	235
8.215.2 Field Documentation	235
8.215.2.1 earfcn	235
8.215.2.2 pci	235
8.215.2.3 status	235

8.216nas_lteRsrpinformation Struct Reference	235
8.216.1 Detailed Description	236
8.216.2 Field Documentation	236
8.216.2.1 rsrplevel	236
8.217nas_LTERSRPThresh Struct Reference	236
8.217.1 Detailed Description	236
8.217.2 Field Documentation	236
8.217.2.1 LTERSRPThreshListLen	236
8.217.2.2 pLTERSRPThreshList	236
8.218nas_LTERSRQThresh Struct Reference	236
8.218.1 Detailed Description	237
8.218.2 Field Documentation	237
8.218.2.1 LTERSRQThreshListLen	237
8.218.2.2 pLTERSRQThreshList	237
8.219nas_LTERSSIThresh Struct Reference	237
8.219.1 Detailed Description	237
8.219.2 Field Documentation	237
8.219.2.1 LTERSSIThreshListLen	237
8.219.2.2 pLTERSSIThreshList	237
8.220nas_LTESigRptCfg Struct Reference	238
8.220.1 Detailed Description	238
8.220.2 Field Documentation	238
8.220.2.1 avgPeriod	238
8.220.2.2 rptRate	238
8.221nas_LTESigRptConfig Struct Reference	238
8.221.1 Detailed Description	239
8.221.2 Field Documentation	239
8.221.2.1 avgPeriod	239
8.221.2.2 rptRate	239
8.222nas_lteSnrinformation Struct Reference	239
8.222.1 Detailed Description	239
8.222.2 Field Documentation	240
8.222.2.1 snrlevel	240
8.223nas_LTESNRThresh Struct Reference	240
8.223.1 Detailed Description	240
8.223.2 Field Documentation	240
8.223.2.1 LTESNRThresListLen	240
8.223.2.2 pLTESNRThresList	240
8.224nas_LTESNRThreshold Struct Reference	240
8.224.1 Detailed Description	240

8.224.2 Field Documentation	241
8.224.2.1 LTESNRThreshListLen	241
8.224.2.2 pLTESNRThreshList	241
8.225nas_LTESysInfo Struct Reference	241
8.225.1 Detailed Description	241
8.225.2 Field Documentation	243
8.225.2.1 cellId	243
8.225.2.2 cellIdValid	243
8.225.2.3 lac	243
8.225.2.4 lacValid	243
8.225.2.5 MCC	243
8.225.2.6 MNC	243
8.225.2.7 networkIdValid	243
8.225.2.8 regRejectInfoValid	243
8.225.2.9 rejCause	243
8.225.2.10rejectSrvDomain	243
8.225.2.11sysInfoLTE	243
8.225.2.12tac	243
8.225.2.13tacValid	243
8.226nas_lteWcdmaCellInfo Struct Reference	243
8.226.1 Detailed Description	244
8.226.2 Field Documentation	244
8.226.2.1 cellReselPriority	244
8.226.2.2 cellsLen	244
8.226.2.3 threshXhigh	244
8.226.2.4 threshXlow	244
8.226.2.5 uarfcn	244
8.226.2.6 WCDMACellInfo	244
8.227nas_minBasedIMSI Struct Reference	244
8.227.1 Detailed Description	245
8.227.2 Field Documentation	245
8.227.2.1 imsiM1112	245
8.227.2.2 imsiMS1	245
8.227.2.3 imsiMS2	245
8.227.2.4 mccM	245
8.228nas_MNRInfo Struct Reference	245
8.228.1 Detailed Description	245
8.228.2 Field Documentation	246
8.228.2.1 mcc	246
8.228.2.2 mnc	246

8.228.2.3 rat	246
8.229nas_ModePrefTlv Struct Reference	246
8.229.1 Detailed Description	246
8.229.2 Field Documentation	246
8.229.2.1 ModePref	246
8.229.2.2 TlvPresent	246
8.230nas_namName Struct Reference	246
8.230.1 Detailed Description	247
8.230.2 Field Documentation	247
8.230.2.1 namName	247
8.230.2.2 namNameLen	247
8.231nas_netSelectionPref Struct Reference	247
8.231.1 Detailed Description	247
8.231.2 Field Documentation	248
8.231.2.1 mcc	248
8.231.2.2 mnc	248
8.231.2.3 netReg	248
8.232nas_NetSelPrefTlv Struct Reference	248
8.232.1 Detailed Description	248
8.232.2 Field Documentation	248
8.232.2.1 NetSelPref	248
8.232.2.2 TlvPresent	248
8.233nas_NetworkStat1x Struct Reference	248
8.233.1 Detailed Description	249
8.233.2 Field Documentation	250
8.233.2.1 ActSetCnt	250
8.233.2.2 NeighborSetCnt	250
8.233.2.3 pActPilotPNElements	250
8.233.2.4 pNeighborSetPilotPN	250
8.233.2.5 RX_EC_IO	250
8.233.2.6 RX_PWR	250
8.233.2.7 SO	250
8.233.2.8 State	250
8.233.2.9 TX_PWR	250
8.234nas_NetworkStatEVDO Struct Reference	250
8.234.1 Detailed Description	251
8.234.2 Field Documentation	252
8.234.2.1 MACIndex	252
8.234.2.2 PER	252
8.234.2.3 PilotEnergy	252

8.234.2.4 pSectorID	252
8.234.2.5 RX_PWR	252
8.234.2.6 SectorIDLen	252
8.234.2.7 SNR	252
8.234.2.8 State	252
8.235nas_nmrCellInfo Struct Reference	252
8.235.1 Detailed Description	252
8.235.2 Field Documentation	253
8.235.2.1 nmrArfcn	253
8.235.2.2 nmrBsic	253
8.235.2.3 nmrCellID	253
8.235.2.4 nmrLac	253
8.235.2.5 nmrPlmn	253
8.235.2.6 nmrRxLev	253
8.236nas_nr5gBandPref Struct Reference	253
8.236.1 Detailed Description	253
8.236.2 Field Documentation	254
8.236.2.1 bits_129_192	254
8.236.2.2 bits_193_256	254
8.236.2.3 bits_1_64	254
8.236.2.4 bits_65_128	254
8.237nas_NR5GCellStatusTlv Struct Reference	254
8.237.1 Detailed Description	254
8.237.2 Field Documentation	254
8.237.2.1 nr5gCellStatus	254
8.237.2.2 TlvPresent	254
8.238nas_NR5GSerStatTlv Struct Reference	255
8.238.1 Detailed Description	255
8.238.2 Field Documentation	255
8.238.2.1 isPrefDataPath	255
8.238.2.2 srvStatus	255
8.238.2.3 TlvPresent	255
8.238.2.4 trueSrvStatus	255
8.239nas_NR5GSystemInfoTlv Struct Reference	256
8.239.1 Detailed Description	256
8.239.2 Field Documentation	258
8.239.2.1 cellId	258
8.239.2.2 cellIdValid	258
8.239.2.3 lac	258
8.239.2.4 lacValid	258

8.239.2.5 MCC	258
8.239.2.6 MNC	258
8.239.2.7 nwIdValid	259
8.239.2.8 regRejectInfoValid	259
8.239.2.9 rejCause	259
8.239.2.10 rejectSrvDomain	259
8.239.2.11 roamStatus	259
8.239.2.12 roamStatusValid	259
8.239.2.13 srvcapability	259
8.239.2.14 srvCapValid	259
8.239.2.15 srvDomain	259
8.239.2.16 srvDomainValid	259
8.239.2.17 sysForbidden	259
8.239.2.18 sysForbiddenValid	259
8.239.2.19 tac	259
8.239.2.20 tacValid	259
8.239.2.21 TlvPresent	259
8.240 nas_operatorNameString Struct Reference	259
8.240.1 Detailed Description	259
8.240.2 Field Documentation	259
8.240.2.1 PLMNName	259
8.241 nas_OperatorPLMNData Struct Reference	259
8.241.1 Detailed Description	260
8.241.2 Field Documentation	260
8.241.2.1 lac1	260
8.241.2.2 lac2	260
8.241.2.3 mcc	260
8.241.2.4 mnc	260
8.241.2.5 PLMNRecID	260
8.242 nas_operatorPLMNList Struct Reference	260
8.242.1 Detailed Description	261
8.242.2 Field Documentation	261
8.242.2.1 numInstance	261
8.242.2.2 PLMNData	261
8.243 nas_PhyCaAggPcellInfo Struct Reference	261
8.243.1 Detailed Description	261
8.243.2 Field Documentation	262
8.243.2.1 dl_bw_value	262
8.243.2.2 freq	262
8.243.2.3 iLTEbandValue	262

8.243.2.4 pci	262
8.243.2.5 TlvPresent	262
8.244nas_PhyCaAggScellDIBw Struct Reference	262
8.244.1 Detailed Description	262
8.244.2 Field Documentation	262
8.244.2.1 dl_bw_value	262
8.244.2.2 TlvPresent	262
8.245nas_PhyCaAggScellIndex Struct Reference	262
8.245.1 Detailed Description	262
8.245.2 Field Documentation	263
8.245.2.1 scell_idx	263
8.245.2.2 TlvPresent	263
8.246nas_PhyCaAggScellIndType Struct Reference	263
8.246.1 Detailed Description	263
8.246.2 Field Documentation	263
8.246.2.1 freq	263
8.246.2.2 pci	263
8.246.2.3 scell_state	263
8.246.2.4 TlvPresent	263
8.247nas_PhyCaAggScellInfo Struct Reference	264
8.247.1 Detailed Description	264
8.247.2 Field Documentation	266
8.247.2.1 dl_bw_value	266
8.247.2.2 freq	266
8.247.2.3 iLTEbandValue	266
8.247.2.4 pci	266
8.247.2.5 scell_state	266
8.247.2.6 TlvPresent	266
8.248nas_PilotSetData Struct Reference	266
8.248.1 Detailed Description	266
8.248.2 Field Documentation	266
8.248.2.1 NumPilots	266
8.248.2.2 pPilotSetInfo	266
8.249nas_PilotSetParams Struct Reference	267
8.249.1 Detailed Description	267
8.249.2 Field Documentation	267
8.249.2.1 PilotPN	267
8.249.2.2 PilotStrength	267
8.249.2.3 PilotType	267
8.250nas_PlmnID Struct Reference	267

8.250.1 Detailed Description	267
8.250.2 Field Documentation	268
8.250.2.1 mcc	268
8.250.2.2 mnc	268
8.250.2.3 pcsDigit	268
8.250.2.4 TlvPresent	268
8.251 nas_PLMNNetworkName Struct Reference	268
8.251.1 Detailed Description	268
8.251.2 Field Documentation	268
8.251.2.1 numInstance	268
8.251.2.2 PLMNNetName	268
8.252 nas_PLMNNetworkNameData Struct Reference	268
8.252.1 Detailed Description	269
8.252.2 Field Documentation	270
8.252.2.1 codingScheme	270
8.252.2.2 countryInitials	270
8.252.2.3 longName	270
8.252.2.4 longNameLen	270
8.252.2.5 longNameSpareBits	270
8.252.2.6 shortName	270
8.252.2.7 shortNameLen	270
8.252.2.8 shortNameSpareBits	270
8.253 nas_PRLPrefTlv Struct Reference	270
8.253.1 Detailed Description	270
8.253.2 Field Documentation	271
8.253.2.1 PRLPref	271
8.253.2.2 TlvPresent	271
8.254 nas_protocolSubtypeElement Struct Reference	271
8.254.1 Detailed Description	271
8.254.2 Field Documentation	272
8.254.2.1 AccessMac	272
8.254.2.2 AuthProt	272
8.254.2.3 ControlMac	272
8.254.2.4 EncryptProt	272
8.254.2.5 ForwardMac	272
8.254.2.6 IdleState	272
8.254.2.7 KeyExchange	272
8.254.2.8 MultDisc	272
8.254.2.9 PhysicalLayer	272
8.254.2.10 ReverseMac	272

8.254.2.11SecProt	272
8.254.2.12VirtStream	272
8.255nas_qaQmi3Gpp2TimeZone Struct Reference	272
8.255.1 Detailed Description	273
8.255.2 Field Documentation	273
8.255.2.1 daylightSavings	273
8.255.2.2 leapSeconds	273
8.255.2.3 localTimeOffset	273
8.256nas_QmiNas3GppNetworkInfo Struct Reference	273
8.256.1 Detailed Description	273
8.256.2 Field Documentation	274
8.256.2.1 Description	274
8.256.2.2 Forbidden	274
8.256.2.3 InUse	274
8.256.2.4 MCC	274
8.256.2.5 MNC	274
8.256.2.6 Preferred	274
8.256.2.7 Roaming	274
8.257nas_QmiNas3GppNetworkRAT Struct Reference	274
8.257.1 Detailed Description	275
8.257.2 Field Documentation	275
8.257.2.1 MCC	275
8.257.2.2 MNC	275
8.257.2.3 RAT	275
8.258nas_QmisNasPcsDigit Struct Reference	275
8.258.1 Detailed Description	275
8.258.2 Field Documentation	276
8.258.2.1 includes_pcs_digit	276
8.258.2.2 MCC	276
8.258.2.3 MNC	276
8.259nas_QmisNasSlqsNasPCICellInfo Struct Reference	276
8.259.1 Detailed Description	276
8.259.2 Field Documentation	276
8.259.2.1 cellID	276
8.259.2.2 freq	276
8.259.2.3 GlobalCellID	276
8.259.2.4 nasQmisNasPcsDigit	276
8.259.2.5 PlmnLen	276
8.260nas_QmisNasSlqsNasPCIInfo Struct Reference	277
8.260.1 Detailed Description	277

8.260.2 Field Documentation	277
8.260.2.1 nasQmisNasSlqsNasPCICellInfo	277
8.260.2.2 PCICellInfoLen	277
8.260.2.3 rsrp	277
8.260.2.4 rsrpRx0	277
8.260.2.5 rsrpRx1	278
8.260.2.6 rsrq	278
8.260.2.7 rsrqRx0	278
8.260.2.8 rsrqRx1	278
8.261 nas_RankIndicatorTlv Struct Reference	278
8.261.1 Detailed Description	278
8.261.2 Field Documentation	278
8.261.2.1 count0	278
8.261.2.2 count1	278
8.261.2.3 TlvPresent	278
8.262 nas_RatDisabledMaskTlv Struct Reference	278
8.262.1 Detailed Description	278
8.262.2 Field Documentation	279
8.262.2.1 ratDisabledMask	279
8.262.2.2 TlvPresent	279
8.263 nas_RejectReasonTlv Struct Reference	279
8.263.1 Detailed Description	279
8.263.2 Field Documentation	279
8.263.2.1 rejectCause	279
8.263.2.2 serviceDomain	279
8.263.2.3 TlvPresent	279
8.264 nas_RFBandInfoElements Struct Reference	279
8.264.1 Detailed Description	280
8.264.2 Field Documentation	280
8.264.2.1 activeBandClass	280
8.264.2.2 activeChannel	280
8.264.2.3 radiolInterface	280
8.265 nas_RfBandInfoExtFormat Struct Reference	280
8.265.1 Detailed Description	280
8.265.2 Field Documentation	281
8.265.2.1 pInstancesSize	281
8.265.2.2 pRfBandInfoExtFormatParam	281
8.265.2.3 TlvPresent	281
8.266 nas_RfBandInfoExtFormatElements Struct Reference	281
8.266.1 Detailed Description	281

8.266.2 Field Documentation	281
8.266.2.1 activeBand	281
8.266.2.2 activeChannel	281
8.266.2.3 radiolInterface	282
8.267nas_RfBandInfoList Struct Reference	282
8.267.1 Detailed Description	282
8.267.2 Field Documentation	282
8.267.2.1 plnInstanceSize	282
8.267.2.2 pRfBandInfoParam	282
8.268nas_RfBandwidthInfo Struct Reference	282
8.268.1 Detailed Description	282
8.268.2 Field Documentation	283
8.268.2.1 plnInstancesSize	283
8.268.2.2 pRfBandwidthInfoParam	283
8.268.2.3 TlvPresent	283
8.269nas_RfBandwidthInfoElements Struct Reference	283
8.269.1 Detailed Description	283
8.269.2 Field Documentation	283
8.269.2.1 bandwidth	283
8.269.2.2 radiolInterface	284
8.270nas_RfDedicatedBandInfo Struct Reference	284
8.270.1 Detailed Description	284
8.270.2 Field Documentation	284
8.270.2.1 plnInstancesSize	284
8.270.2.2 pRfDedicatedBandInfoParam	284
8.270.2.3 TlvPresent	284
8.271nas_RfDedicatedBandInfoElements Struct Reference	284
8.271.1 Detailed Description	284
8.271.2 Field Documentation	285
8.271.2.1 dedicatedBand	285
8.271.2.2 radiolInterface	285
8.272nas_RFInfoTlv Struct Reference	285
8.272.1 Detailed Description	285
8.272.2 Field Documentation	286
8.272.2.1 activeBandClass	286
8.272.2.2 activeChannel	286
8.272.2.3 radiolInterface	286
8.272.2.4 radiolInterfaceSize	286
8.272.2.5 TlvPresent	286
8.273nas_roamIndList Struct Reference	286

8.273.1 Detailed Description	286
8.273.2 Field Documentation	286
8.273.2.1 numInstances	287
8.273.2.2 radiolInterface	287
8.273.2.3 roamIndicator	287
8.274nas_RoamPrefTlv Struct Reference	287
8.274.1 Detailed Description	287
8.274.2 Field Documentation	287
8.274.2.1 RoamPref	287
8.274.2.2 TlvPresent	287
8.275nas_RSRPThresh Struct Reference	287
8.275.1 Detailed Description	287
8.275.2 Field Documentation	288
8.275.2.1 pRSRPThresList	288
8.275.2.2 RSRPThresListLen	288
8.276nas_rsrqInformation Struct Reference	288
8.276.1 Detailed Description	288
8.276.2 Field Documentation	288
8.276.2.1 radiolf	288
8.276.2.2 rsrq	288
8.277nas_RSRQThresh Struct Reference	288
8.277.1 Detailed Description	289
8.277.2 Field Documentation	289
8.277.2.1 pRSRQThresList	289
8.277.2.2 RSRQThresListLen	289
8.278nas_RSSIThresh Struct Reference	289
8.278.1 Detailed Description	289
8.278.2 Field Documentation	290
8.278.2.1 pRSSIThresList	290
8.278.2.2 RSSIThresListLen	290
8.279nas_rxInfo Struct Reference	290
8.279.1 Detailed Description	290
8.279.2 Field Documentation	291
8.279.2.1 ecio	291
8.279.2.2 isRadioTuned	291
8.279.2.3 phase	291
8.279.2.4 rscp	291
8.279.2.5 rsrp	291
8.279.2.6 rxPower	291
8.280nas_RxSigInfo Struct Reference	291

8.280.1 Detailed Description	291
8.280.2 Field Documentation	292
8.280.2.1 isRadioTuned	292
8.280.2.2 rsrp	292
8.280.2.3 rxChainIndex	292
8.280.2.4 rxPower	292
8.281 nas_rxSignalStrengthListElement Struct Reference	292
8.281.1 Detailed Description	292
8.281.2 Field Documentation	293
8.281.2.1 radiolf	293
8.281.2.2 rxSignalStrength	293
8.282 nas_SccRxInfo Struct Reference	293
8.282.1 Detailed Description	293
8.282.2 Field Documentation	294
8.282.2.1 numInstances	294
8.282.2.2 rsrq	294
8.282.2.3 sigInfo	294
8.282.2.4 snr	294
8.282.2.5 TlvPresent	294
8.283 nas_serviceProviderName Struct Reference	294
8.283.1 Detailed Description	294
8.283.2 Field Documentation	294
8.283.2.1 displayCondition	294
8.283.2.2 spn	294
8.283.2.3 spnLength	294
8.284 nas_servSystem Struct Reference	295
8.284.1 Detailed Description	295
8.284.2 Field Documentation	296
8.284.2.1 csAttachState	296
8.284.2.2 numRadioInterfaces	296
8.284.2.3 psAttachState	296
8.284.2.4 radioInterface	296
8.284.2.5 regState	296
8.284.2.6 selNetwork	296
8.285 nas_sidNid Struct Reference	296
8.285.1 Detailed Description	296
8.285.2 Field Documentation	296
8.285.2.1 nid	296
8.285.2.2 sid	296
8.286 nas_SignalStrengthTlv Struct Reference	297

8.286.1 Detailed Description	297
8.286.2 Field Documentation	297
8.286.2.1 radiolInterface	297
8.286.2.2 signalStrength	297
8.286.2.3 TlvPresent	297
8.287nas_SLQSSignalStrengthsIndReq Struct Reference	297
8.287.1 Detailed Description	297
8.287.2 Field Documentation	298
8.287.2.1 ecioDelta	298
8.287.2.2 ecioThresholdList	298
8.287.2.3 ecioThresholdListLen	299
8.287.2.4 ioDelta	299
8.287.2.5 lteRsrpDelta	299
8.287.2.6 lteSnrDelta	299
8.287.2.7 rsrqDelta	299
8.287.2.8 rxSignalStrengthDelta	299
8.287.2.9 sinrDelta	299
8.287.2.10sinrThresholdList	299
8.287.2.11sinrThresholdListLen	299
8.288nas_SLQSSignalStrengthsInformation Struct Reference	299
8.288.1 Detailed Description	299
8.288.2 Field Documentation	300
8.288.2.1 ecioInfo	300
8.288.2.2 errorRateInfo	300
8.288.2.3 io	300
8.288.2.4 lteRsrpinfo	300
8.288.2.5 lteSnrinfo	300
8.288.2.6 rsrqInfo	300
8.288.2.7 rxSignalStrengthInfo	300
8.288.2.8 sinr	300
8.289nas_SLQSSignalStrengthsTlv Struct Reference	300
8.289.1 Detailed Description	300
8.289.2 Field Documentation	301
8.289.2.1 sSLQSSignalStrengthsInfo	301
8.289.2.2 TlvPresent	301
8.290nas_SrvDomainPrefTlv Struct Reference	301
8.290.1 Detailed Description	301
8.290.2 Field Documentation	301
8.290.2.1 SrvDomainPref	301
8.290.2.2 TlvPresent	301

8.291nas_SrvStatusInfo Struct Reference	301
8.291.1 Detailed Description	301
8.291.2 Field Documentation	302
8.291.2.1 isPrefDataPath	302
8.291.2.2 srvStatus	302
8.292nas_sysInfoCommon Struct Reference	302
8.292.1 Detailed Description	302
8.292.2 Field Documentation	304
8.292.2.1 isSysForbidden	304
8.292.2.2 isSysForbiddenValid	304
8.292.2.3 roamStatus	304
8.292.2.4 roamStatusValid	304
8.292.2.5 srvCapability	304
8.292.2.6 srvCapabilityValid	304
8.292.2.7 srvDomain	304
8.292.2.8 srvDomainValid	304
8.293nas_TDSCDMAECIOThresh Struct Reference	305
8.293.1 Detailed Description	305
8.293.2 Field Documentation	305
8.293.2.1 pTDSCDMAECIOThreshList	305
8.293.2.2 TDSCDMAECIOThreshListLen	305
8.294nas_TDSCDMARSCPThresh Struct Reference	305
8.294.1 Detailed Description	305
8.294.2 Field Documentation	305
8.294.2.1 pTDSCDMARSCPThreshList	305
8.294.2.2 TDSCDMARSCPThreshListLen	306
8.295nas_TDSCDMARSSIThresh Struct Reference	306
8.295.1 Detailed Description	306
8.295.2 Field Documentation	306
8.295.2.1 pTDSCDMARSSIThreshList	306
8.295.2.2 TDSCDMARSSIThreshListLen	306
8.296nas_TDSCDMASINRCONFThresh Struct Reference	306
8.296.1 Detailed Description	306
8.296.2 Field Documentation	306
8.296.2.1 pTDSCDMASINRCONFThreshList	307
8.296.2.2 TDSCDMASINRCONFThreshListLen	307
8.297nas_TDSCDMASINRThresh Struct Reference	307
8.297.1 Detailed Description	307
8.297.2 Field Documentation	307
8.297.2.1 pTDSCDMASINRThreshList	307

8.297.2.2 TDSCDMASINRThreshListLen	307
8.298nas_timeInfo Struct Reference	307
8.298.1 Detailed Description	307
8.298.2 Field Documentation	309
8.298.2.1 day	309
8.298.2.2 dayLtSavingAdj	309
8.298.2.3 dayOfWeek	309
8.298.2.4 hour	309
8.298.2.5 minute	309
8.298.2.6 month	309
8.298.2.7 radiolInterface	309
8.298.2.8 second	309
8.298.2.9 timeZone	309
8.298.2.10TlvPresent	309
8.298.2.11year	309
8.299nas_trueIMSI Struct Reference	309
8.299.1 Detailed Description	309
8.299.2 Field Documentation	310
8.299.2.1 imsiT1112	310
8.299.2.2 imsiTaddrNum	310
8.299.2.3 imsiTS1	310
8.299.2.4 imsiTS2	310
8.299.2.5 mccT	310
8.300nas_txInfo Struct Reference	310
8.300.1 Detailed Description	310
8.300.2 Field Documentation	310
8.300.2.1 isInTraffic	310
8.300.2.2 txPower	310
8.301nas_UMTSInfo Struct Reference	310
8.301.1 Detailed Description	311
8.301.2 Field Documentation	312
8.301.2.1 cellID	312
8.301.2.2 ecio	312
8.301.2.3 geranInst	312
8.301.2.4 GeranInstInfo	312
8.301.2.5 lac	312
8.301.2.6 plmn	312
8.301.2.7 psc	312
8.301.2.8 rscp	312
8.301.2.9 uarfcn	312

8.301.2.10 umtsInst	312
8.301.2.11 UMTSInstInfo	312
8.302nas_UMTSInstInfo Struct Reference	312
8.302.1 Detailed Description	312
8.302.2 Field Documentation	313
8.302.2.1 umtsEcio	313
8.302.2.2 umtsPsc	313
8.302.2.3 umtsRscp	313
8.302.2.4 umtsUarfcn	313
8.303nas_umtsLTENbrCell Struct Reference	313
8.303.1 Detailed Description	313
8.303.2 Field Documentation	314
8.303.2.1 cellsTDD	314
8.303.2.2 earfcn	314
8.303.2.3 pci	314
8.303.2.4 rsrp	314
8.303.2.5 rsrq	314
8.303.2.6 srxlev	314
8.304nas_UniversalTime Struct Reference	314
8.304.1 Detailed Description	314
8.304.2 Field Documentation	315
8.304.2.1 day	315
8.304.2.2 dayOfWeek	315
8.304.2.3 hour	315
8.304.2.4 minute	315
8.304.2.5 month	315
8.304.2.6 second	315
8.304.2.7 year	315
8.305nas_wcdmaCellInfo Struct Reference	315
8.305.1 Detailed Description	315
8.305.2 Field Documentation	316
8.305.2.1 cpich_ecno	316
8.305.2.2 cpich_rscp	316
8.305.2.3 psc	316
8.305.2.4 srxlev	316
8.306nas_WCDMAECIOThresh Struct Reference	316
8.306.1 Detailed Description	316
8.306.2 Field Documentation	316
8.306.2.1 pWCDMAECIOThreshList	316
8.306.2.2 WCDMAECIOThreshListLen	316

8.307nas_WCDMAInfoLTENeighborCell Struct Reference	316
8.307.1 Detailed Description	317
8.307.2 Field Documentation	317
8.307.2.1 UMTSLTENbrCell	317
8.307.2.2 umtsLTENbrCellLen	317
8.307.2.3 wcdmaRRCTest	317
8.308nas_WCDMARSSITresh Struct Reference	317
8.308.1 Detailed Description	317
8.308.2 Field Documentation	318
8.308.2.1 pWCDMARSSITreshList	318
8.308.2.2 WCDMARSSITreshListLen	318
8.309nas_WCDMASysInfo Struct Reference	318
8.309.1 Detailed Description	318
8.309.2 Field Documentation	320
8.309.2.1 cellId	321
8.309.2.2 cellIdValid	321
8.309.2.3 hsCallStatus	321
8.309.2.4 hsCallStatusValid	321
8.309.2.5 hsInd	321
8.309.2.6 hsIndValid	321
8.309.2.7 lac	321
8.309.2.8 lacValid	321
8.309.2.9 MCC	321
8.309.2.10MNC	321
8.309.2.11networkIdValid	321
8.309.2.12psc	321
8.309.2.13pscValid	321
8.309.2.14regRejectInfoValid	321
8.309.2.15rejCause	321
8.309.2.16rejectSrvDomain	321
8.309.2.17sysInfoWCDMA	321
8.310nas_wcdmaUARFCN Struct Reference	321
8.310.1 Detailed Description	321
8.310.2 Field Documentation	322
8.310.2.1 status	322
8.310.2.2 uarfcn	322
8.311NASAcqOrderPrefTlv Struct Reference	322
8.311.1 Detailed Description	322
8.311.2 Field Documentation	322
8.311.2.1 AcqOrderLen	322

8.311.2.2 AcqOrderPref	322
8.311.2.3 TlvPresent	322
8.312NASBandPreferenceTlv Struct Reference	322
8.312.1 Detailed Description	322
8.312.2 Field Documentation	322
8.312.2.1 band_pref	322
8.312.2.2 TlvPresent	322
8.313NASCiotAcqOrderPrefTlv Struct Reference	323
8.313.1 Detailed Description	323
8.313.2 Field Documentation	323
8.313.2.1 CiotAcqOrderLen	323
8.313.2.2 CiotAcqOrderPref	323
8.313.2.3 TlvPresent	323
8.314NASCiotLteOpModePrefTlv Struct Reference	323
8.314.1 Detailed Description	323
8.314.2 Field Documentation	323
8.314.2.1 CiotLteOpModePref	323
8.314.2.2 TlvPresent	323
8.315NASEmergencyModeTlv Struct Reference	323
8.315.1 Detailed Description	324
8.315.2 Field Documentation	324
8.315.2.1 EmerMode	324
8.315.2.2 TlvPresent	324
8.316NasGetLTECphyCalInfo Struct Reference	324
8.316.1 Detailed Description	324
8.316.2 Field Documentation	325
8.316.2.1 PhyCaAggPcellInfo	325
8.316.2.2 PhyCaAggScellArray	325
8.316.2.3 PhyCaAggScellDIBw	325
8.316.2.4 PhyCaAggScellIndex	325
8.316.2.5 PhyCaAggScellIndType	325
8.316.2.6 PhyCaAggScellInfo	325
8.317NASGWAcqOrderPrefTlv Struct Reference	325
8.317.1 Detailed Description	325
8.317.2 Field Documentation	325
8.317.2.1 GWAcqOrderPref	325
8.317.2.2 TlvPresent	325
8.318NASLTEBandPreferenceTlv Struct Reference	325
8.318.1 Detailed Description	326
8.318.2 Field Documentation	326

8.318.2.1 LteBandPref	326
8.318.2.2 TlvPresent	326
8.319NASLteM1BandPrefTlv Struct Reference	326
8.319.1 Detailed Description	326
8.319.2 Field Documentation	326
8.319.2.1 LteM1BandPref	326
8.319.2.2 TlvPresent	326
8.320NASLteNasReleaseInfoTlv Struct Reference	326
8.320.1 Detailed Description	326
8.320.2 Field Documentation	327
8.320.2.1 nas_major	327
8.320.2.2 nas_minor	327
8.320.2.3 nas_release	327
8.320.2.4 TlvPresent	327
8.321NASLteNB1BandPrefTlv Struct Reference	327
8.321.1 Detailed Description	327
8.321.2 Field Documentation	327
8.321.2.1 LteNB1BandPref	327
8.321.2.2 TlvPresent	327
8.322NASModePreferenceTlv Struct Reference	327
8.322.1 Detailed Description	328
8.322.2 Field Documentation	328
8.322.2.1 ModePref	328
8.322.2.2 TlvPresent	328
8.323NASNetSelPreferenceTlv Struct Reference	328
8.323.1 Detailed Description	328
8.323.2 Field Documentation	328
8.323.2.1 NetSelPref	328
8.323.2.2 TlvPresent	328
8.324NASNr5gBandPrefTlv Struct Reference	328
8.324.1 Detailed Description	328
8.324.2 Field Documentation	329
8.324.2.1 Nr5gBandPrefbits1_64	329
8.324.2.2 Nr5gBandPrefbits_129_192	329
8.324.2.3 Nr5gBandPrefbits_193_256	329
8.324.2.4 Nr5gBandPrefbits_65_128	329
8.324.2.5 TlvPresent	329
8.325NASOTAMessageTlv Struct Reference	329
8.325.1 Detailed Description	329
8.325.2 Field Documentation	329

8.325.2.1 data_buf	330
8.325.2.2 data_len	330
8.325.2.3 message_type	330
8.325.2.4 TlvPresent	330
8.326NASPhyCaAggPcellInfo Struct Reference	330
8.326.1 Detailed Description	330
8.326.2 Field Documentation	330
8.326.2.1 dl_bw_value	330
8.326.2.2 freq	330
8.326.2.3 iLTEbandValue	330
8.326.2.4 pci	330
8.326.2.5 TlvPresent	331
8.327NASPhyCaAggScellArray Struct Reference	331
8.327.1 Detailed Description	331
8.327.2 Field Documentation	332
8.327.2.1 band	332
8.327.2.2 cphy_ca_dl_bandwidth	332
8.327.2.3 cphy_scell_info_list_len	332
8.327.2.4 freq	332
8.327.2.5 pci	332
8.327.2.6 scell_idx	332
8.327.2.7 scell_state	332
8.327.2.8 TlvPresent	332
8.328NASPhyCaAggScellIDBw Struct Reference	332
8.328.1 Detailed Description	332
8.328.2 Field Documentation	332
8.328.2.1 dl_bw_value	332
8.328.2.2 TlvPresent	332
8.329NASPhyCaAggScellIndex Struct Reference	332
8.329.1 Detailed Description	333
8.329.2 Field Documentation	333
8.329.2.1 scell_idx	333
8.329.2.2 TlvPresent	333
8.330NASPhyCaAggScellIndType Struct Reference	333
8.330.1 Detailed Description	333
8.330.2 Field Documentation	333
8.330.2.1 freq	334
8.330.2.2 pci	334
8.330.2.3 scell_state	334
8.330.2.4 TlvPresent	334

8.331NASPhyCaAggScellInfo Struct Reference	334
8.331.1 Detailed Description	334
8.331.2 Field Documentation	334
8.331.2.1 dl_bw_value	334
8.331.2.2 freq	335
8.331.2.3 iLTEbandValue	335
8.331.2.4 pci	335
8.331.2.5 scell_state	335
8.331.2.6 TlvPresent	335
8.332NASPRLPreferenceTlv Struct Reference	335
8.332.1 Detailed Description	335
8.332.2 Field Documentation	335
8.332.2.1 PRLPref	335
8.332.2.2 TlvPresent	335
8.333NASQmiCbkNasSwtOTAMessageInd Struct Reference	335
8.333.1 Detailed Description	335
8.333.2 Field Documentation	336
8.333.2.1 nasRelInfoTlv	336
8.333.2.2 otaMsgTlv	336
8.333.2.3 timeTlv	336
8.334NASQmiCbkNasSystemSelPrefInd Struct Reference	336
8.334.1 Detailed Description	336
8.334.2 Field Documentation	338
8.334.2.1 AOPTlv	338
8.334.2.2 BPTlv	338
8.334.2.3 CiotAOPTlv	338
8.334.2.4 CiotOpMPTlv	338
8.334.2.5 EMTlv	338
8.334.2.6 GWAOPTlv	338
8.334.2.7 LBPTlv	338
8.334.2.8 M1BandPTlv	338
8.334.2.9 MPTlv	338
8.334.2.10NB1BandPTlv	338
8.334.2.11NR5gBandPTlv	338
8.334.2.12NSPTlv	338
8.334.2.13PRLPTlv	338
8.334.2.14RatDMTlv	338
8.334.2.15RPTlv	338
8.334.2.16SDPTlv	338
8.335NASRatDisabledMaskTlv Struct Reference	338

8.335.1 Detailed Description	338
8.335.2 Field Documentation	339
8.335.2.1 RatDisabledMask	339
8.335.2.2 TlvPresent	339
8.336NASRoamPreferenceTlv Struct Reference	339
8.336.1 Detailed Description	339
8.336.2 Field Documentation	339
8.336.2.1 RoamPref	339
8.336.2.2 TlvPresent	339
8.337NASServDomainPrefTlv Struct Reference	339
8.337.1 Detailed Description	339
8.337.2 Field Documentation	339
8.337.2.1 SrvDomainPref	339
8.337.2.2 TlvPresent	339
8.338NASServingSystemInfo Struct Reference	339
8.338.1 Detailed Description	340
8.338.2 Field Documentation	341
8.338.2.1 csAttachState	341
8.338.2.2 hdrPersonality	341
8.338.2.3 psAttachState	341
8.338.2.4 radiolInterfaceList	341
8.338.2.5 radiolInterfaceNo	341
8.338.2.6 registrationState	341
8.338.2.7 selectedNetwork	341
8.339NASTimeInfoTlv Struct Reference	341
8.339.1 Detailed Description	341
8.339.2 Field Documentation	341
8.339.2.1 time	341
8.339.2.2 TlvPresent	341
8.340newMTMessageTlv Struct Reference	342
8.340.1 Detailed Description	342
8.340.2 Field Documentation	342
8.340.2.1 MTMessageInfo	342
8.340.2.2 TlvPresent	342
8.341pack_audio_SLQSGetAudioPathConfig_t Struct Reference	342
8.341.1 Detailed Description	342
8.341.2 Field Documentation	343
8.341.2.1 Item	343
8.341.2.2 Profile	343
8.342pack_audio_SLQSGetAudioProfile_t Struct Reference	343

8.342.1 Detailed Description	343
8.342.2 Field Documentation	343
8.342.2.1 Generator	343
8.343pack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference	344
8.343.1 Detailed Description	344
8.343.2 Field Documentation	344
8.343.2.1 Generator	344
8.343.2.2 Item	344
8.343.2.3 Profile	344
8.343.2.4 Volume	344
8.344pack_audio_SLQSSetAudioPathConfig_t Struct Reference	344
8.344.1 Detailed Description	345
8.344.2 Field Documentation	346
8.344.2.1 pCodecSTGain	346
8.344.2.2 pDTMFTXGain	346
8.344.2.3 pECMode	346
8.344.2.4 pNSEnable	346
8.344.2.5 Profile	346
8.344.2.6 pRXAGCList	346
8.344.2.7 pRXAVCAGCSwitch	346
8.344.2.8 pRXAVCList	346
8.344.2.9 pRXPCMIIRFitr	346
8.344.2.10pTXAGCList	346
8.344.2.11pTXAVCSwitch	346
8.344.2.12pTXGain	346
8.344.2.13pTXPCMIIRFitr	346
8.345pack_audio_SLQSSetAudioProfile_t Struct Reference	346
8.345.1 Detailed Description	347
8.345.2 Field Documentation	347
8.345.2.1 EarMute	347
8.345.2.2 Generator	347
8.345.2.3 MicMute	347
8.345.2.4 Profile	347
8.345.2.5 Volume	347
8.346pack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference	348
8.346.1 Detailed Description	348
8.346.2 Field Documentation	348
8.346.2.1 Generator	348
8.346.2.2 Item	348
8.346.2.3 Profile	348

8.346.2.4 Volume	348
8.346.2.5 VolValue	348
8.347pack_cat_CATSendEnvelopeCommand_t Struct Reference	348
8.347.1 Detailed Description	349
8.347.2 Field Documentation	349
8.347.2.1 cmdID	349
8.347.2.2 dataLen	349
8.347.2.3 pData	349
8.348pack_cat_CATSendTerminalResponse_t Struct Reference	349
8.348.1 Detailed Description	349
8.348.2 Field Documentation	350
8.348.2.1 dataLen	350
8.348.2.2 pData	350
8.348.2.3 refID	350
8.349pack_cat_SetCATEventCallback_t Struct Reference	350
8.349.1 Detailed Description	350
8.349.2 Field Documentation	350
8.349.2.1 eventMask	350
8.350pack_dms_ActivateAutomatic_t Struct Reference	351
8.350.1 Detailed Description	351
8.350.2 Field Documentation	351
8.350.2.1 actCode	351
8.351pack_dms_GetCustFeaturesV2_t Struct Reference	351
8.351.1 Detailed Description	351
8.351.2 Field Documentation	351
8.351.2.1 cust_id	351
8.351.2.2 list_type	351
8.351.2.3 Tlvresult	352
8.352pack_dms_ResetToFactoryDefaults_t Struct Reference	352
8.352.1 Detailed Description	352
8.352.2 Field Documentation	352
8.352.2.1 spc	352
8.353pack_dms_SetActivationStatusCallback_t Struct Reference	352
8.353.1 Detailed Description	352
8.353.2 Field Documentation	352
8.353.2.1 activationState	352
8.354pack_dms_SetCrashAction_t Struct Reference	352
8.354.1 Detailed Description	353
8.354.2 Field Documentation	353
8.354.2.1 crashAction	353

8.355pack_dms_SetCustFeature_t Struct Reference	353
8.355.1 Detailed Description	353
8.355.2 Field Documentation	355
8.355.2.1 DHCPRelayEnabled	355
8.355.2.2 DisableIMSI	355
8.355.2.3 GpsEnable	355
8.355.2.4 GPSLPM	355
8.355.2.5 GPSSel	355
8.355.2.6 IPFamSupport	355
8.355.2.7 IsVoiceEnabled	355
8.355.2.8 RMAutoConnect	355
8.355.2.9 SMSSupport	355
8.356pack_dms_SetCustFeaturesV2_t Struct Reference	355
8.356.1 Detailed Description	355
8.356.2 Field Documentation	356
8.356.2.1 cust_id	356
8.356.2.2 cust_value	356
8.356.2.3 Tlvresult	356
8.356.2.4 value_length	356
8.357pack_dms_SetEventReport_t Struct Reference	356
8.357.1 Detailed Description	356
8.357.2 Field Documentation	356
8.357.2.1 mode	356
8.358pack_dms_SetIndicationRegister_t Struct Reference	356
8.358.1 Detailed Description	356
8.358.2 Field Documentation	357
8.358.2.1 PSMCfgChangeInfo	357
8.358.2.2 PSMStatus	357
8.358.2.3 RptIMSCapability	357
8.359pack_dms_SetPower_t Struct Reference	357
8.359.1 Detailed Description	357
8.359.2 Field Documentation	358
8.359.2.1 mode	358
8.359.2.2 Tlvresult	358
8.360pack_dms_SetUSBComp_t Struct Reference	358
8.360.1 Detailed Description	358
8.360.2 Field Documentation	358
8.360.2.1 Tlvresult	358
8.360.2.2 USBComp	359
8.361pack_dms_SLQSDmsSwlIndicationRegister_t Struct Reference	359

8.361.1 Detailed Description	359
8.361.2 Field Documentation	359
8.361.2.1 resetInfoInd	359
8.362pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference	359
8.362.1 Detailed Description	359
8.362.2 Field Documentation	360
8.362.2.1 pActiveTimer	360
8.362.2.2 pPeriodicUpdateTimer	360
8.362.2.3 pPsmEnableState	360
8.363pack_dms_SLQSSwiGetCrashInfo_t Struct Reference	360
8.363.1 Detailed Description	360
8.363.2 Field Documentation	360
8.363.2.1 clear	360
8.364pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	360
8.364.1 Detailed Description	360
8.364.2 Field Documentation	361
8.364.2.1 pDestSMSContent	361
8.364.2.2 pDestSMSNum	361
8.365pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference	361
8.365.1 Detailed Description	361
8.365.2 Field Documentation	362
8.365.2.1 hostID	362
8.365.2.2 manString	362
8.365.2.3 modelString	362
8.365.2.4 plasmaIDString	362
8.365.2.5 swVerString	362
8.366pack_dms_SLQSSwiSetOSInfo_t Struct Reference	362
8.366.1 Detailed Description	362
8.366.2 Field Documentation	362
8.366.2.1 nameString	362
8.366.2.2 versionString	362
8.367pack_dms_SwiSetEventReport_t Struct Reference	362
8.367.1 Detailed Description	362
8.367.2 Field Documentation	363
8.367.2.1 pTempReport	363
8.367.2.2 pUIMStatusReport	363
8.367.2.3 pVoltReport	363
8.368pack_dms_SwiUimSelect_t Struct Reference	363
8.368.1 Detailed Description	363
8.368.2 Field Documentation	363

8.368.2.1 uim_select	364
8.369pack_dms_UIMChangePIN_t Struct Reference	364
8.369.1 Detailed Description	364
8.369.2 Field Documentation	364
8.369.2.1 id	364
8.369.2.2 newValue	364
8.369.2.3 oldValue	364
8.370pack_dms_UIMGetControlKeyStatus_t Struct Reference	364
8.370.1 Detailed Description	364
8.370.2 Field Documentation	365
8.370.2.1 facility	365
8.371pack_dms_UIMGetICCID_t Struct Reference	365
8.371.1 Detailed Description	365
8.371.2 Field Documentation	365
8.371.2.1 ParamPresenceMask	365
8.371.2.2 Tlvresult	365
8.372pack_dms_UIMSetControlKeyProtection_t Struct Reference	365
8.372.1 Detailed Description	365
8.372.2 Field Documentation	366
8.372.2.1 facility	366
8.372.2.2 facilityCk	366
8.372.2.3 facilityState	366
8.373pack_dms_UIMSetPINProtection_t Struct Reference	366
8.373.1 Detailed Description	366
8.373.2 Field Documentation	367
8.373.2.1 bEnable	367
8.373.2.2 id	367
8.373.2.3 value	367
8.374pack_dms_UIMUnblockControlKey_t Struct Reference	367
8.374.1 Detailed Description	367
8.374.2 Field Documentation	367
8.374.2.1 facility	367
8.374.2.2 facilityCk	367
8.375pack_dms_UIMUnblockPIN_t Struct Reference	367
8.375.1 Detailed Description	367
8.375.2 Field Documentation	368
8.375.2.1 id	368
8.375.2.2 newPin	368
8.375.2.3 pukValue	368
8.376pack_dms_UIMVerifyPIN_t Struct Reference	368

8.376.1 Detailed Description	368
8.376.2 Field Documentation	368
8.376.2.1 id	368
8.376.2.2 value	368
8.377pack_fms_GetImagesPreference_t Struct Reference	368
8.377.1 Detailed Description	369
8.377.2 Field Documentation	369
8.377.2.1 Tlvresult	369
8.378pack_fms_GetStoredImages_t Struct Reference	369
8.378.1 Detailed Description	369
8.378.2 Field Documentation	369
8.378.2.1 Tlvresult	369
8.379pack_fms_SetImagesPreference_t Struct Reference	369
8.379.1 Detailed Description	369
8.379.2 Field Documentation	370
8.379.2.1 bForceDownload	370
8.379.2.2 imageListSize	370
8.379.2.3 modemindex	370
8.379.2.4 pImageList	370
8.379.2.5 Tlvresult	370
8.380pack_ims_SLQSImsConfigIndicationRegister_t Struct Reference	370
8.380.1 Detailed Description	370
8.380.2 Field Documentation	371
8.380.2.1 pRegMgrConfigEvents	371
8.380.2.2 pSIPConfigEvents	371
8.380.2.3 pSMSConfigEvents	371
8.380.2.4 pUserConfigEvents	371
8.380.2.5 pVoIPConfigEvents	371
8.381pack_ims_SLQSSetIMSSMSConfig_t Struct Reference	371
8.381.1 Detailed Description	372
8.381.2 Field Documentation	372
8.381.2.1 pPhoneCtxtURI	372
8.381.2.2 pPhoneCtxtURILen	372
8.381.2.3 pSMSFormat	372
8.381.2.4 pSMSOverIPNwInd	372
8.382pack_ims_SLQSSetIMSUserConfig_t Struct Reference	372
8.382.1 Detailed Description	372
8.382.2 Field Documentation	373
8.382.2.1 pIMSDomain	373
8.382.2.2 pIMSDomainLen	373

8.383	pack_ims_SLQSSetIMSVoIPConfig_t Struct Reference	373
8.383.1	Detailed Description	373
8.383.2	Field Documentation	375
8.383.2.1	pAmrMode	375
8.383.2.2	pAmrOctetAligned	375
8.383.2.3	pAmrWbEnable	375
8.383.2.4	pAmrWBMode	375
8.383.2.5	pAmrWBOctetAligned	375
8.383.2.6	pMinSessionExpiryTimer	375
8.383.2.7	pRingBackTimer	375
8.383.2.8	pRingingTimer	375
8.383.2.9	pRTPRTCPInactTimer	375
8.383.2.10	pScrAmrEnable	375
8.383.2.11	pScrAmrWbEnable	375
8.383.2.12	pSessionExpiryTimer	375
8.384	pack_ims_SLQSSetRegMgrConfig_t Struct Reference	375
8.384.1	Detailed Description	375
8.384.2	Field Documentation	376
8.384.2.1	pCSCFPortName	376
8.384.2.2	pCSCFPortNameLen	376
8.384.2.3	pIMSTestMode	376
8.384.2.4	pPriCSCFPort	376
8.385	pack_ims_SLQSSetSIPConfig_t Struct Reference	376
8.385.1	Detailed Description	376
8.385.2	Field Documentation	376
8.385.2.1	pSigCompEnabled	377
8.385.2.2	pSIPLocalPort	377
8.385.2.3	pSubscribeTimer	377
8.385.2.4	pTimerSIPReg	377
8.385.2.5	pTimerT1	377
8.385.2.6	pTimerT2	377
8.385.2.7	pTimerTf	377
8.386	pack_imsa_SLQSRegisterIMSAIndication_t Struct Reference	377
8.386.1	Detailed Description	377
8.386.2	Field Documentation	378
8.386.2.1	has_PdpStatusConfig	378
8.386.2.2	has_RatHandoverStatusConfig	378
8.386.2.3	has_RegStatusConfig	378
8.386.2.4	has_ServiceStatusConfig	378
8.386.2.5	PdpStatusConfig	378

8.386.2.6 RatHandoverStatusConfig	378
8.386.2.7 RegStatusConfig	378
8.386.2.8 ServiceStatusConfig	378
8.387pack_loc_Delete_Assist_Data_t Struct Reference	378
8.387.1 Detailed Description	378
8.387.2 Field Documentation	379
8.387.2.1 pBdsSVInfo	379
8.387.2.2 pCellDb	379
8.387.2.3 pClkInfo	379
8.387.2.4 pGnssData	379
8.387.2.5 pSVInfo	379
8.387.2.6 Tlvresult	379
8.388pack_loc_EventRegister_t Struct Reference	379
8.388.1 Detailed Description	379
8.388.2 Field Documentation	381
8.388.2.1 eventRegister	381
8.388.2.2 Tlvresult	381
8.389pack_loc_SetExtPowerState_t Struct Reference	381
8.389.1 Detailed Description	381
8.389.2 Field Documentation	381
8.389.2.1 extPowerState	381
8.389.2.2 Tlvresult	381
8.390pack_loc_SetOperationMode_t Struct Reference	381
8.390.1 Detailed Description	382
8.390.2 Field Documentation	382
8.390.2.1 mode	382
8.390.2.2 Tlvresult	382
8.391pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	382
8.391.1 Detailed Description	382
8.391.2 Field Documentation	382
8.391.2.1 Tlvresult	382
8.391.2.2 xid	383
8.392pack_loc_SLQSLOCGetServer_t Struct Reference	383
8.392.1 Detailed Description	383
8.392.2 Field Documentation	383
8.392.2.1 has_serverAddrTypeMask	383
8.392.2.2 serverAddrTypeMask	383
8.392.2.3 serverType	383
8.393pack_loc_SLQSLOCInjectPosition_t Struct Reference	383
8.393.1 Detailed Description	384

8.393.2 Field Documentation	387
8.393.2.1 altitudeSrcInfo	387
8.393.2.2 altitudeWrtEllipsoid	387
8.393.2.3 altitudeWrtMeanSeaLevel	387
8.393.2.4 has_altitudeSrcInfo	387
8.393.2.5 has_altitudeWrtEllipsoid	387
8.393.2.6 has_altitudeWrtMeanSeaLevel	387
8.393.2.7 has_horConfidence	387
8.393.2.8 has_horReliability	387
8.393.2.9 has_horUncCircular	387
8.393.2.10 has_latitude	387
8.393.2.11 has_longitude	387
8.393.2.12 has_positionSrc	387
8.393.2.13 has_rawHorConfidence	387
8.393.2.14 has_rawHorUncCircular	387
8.393.2.15 has_timestampAge	387
8.393.2.16 has_timestampUtc	387
8.393.2.17 has_vertConfidence	387
8.393.2.18 has_vertReliability	388
8.393.2.19 has_vertUnc	388
8.393.2.20 horConfidence	388
8.393.2.21 horReliability	388
8.393.2.22 horUncCircular	388
8.393.2.23 attitude	388
8.393.2.24 longitude	388
8.393.2.25 positionSrc	388
8.393.2.26 rawHorConfidence	388
8.393.2.27 rawHorUncCircular	388
8.393.2.28 timestampAge	388
8.393.2.29 timestampUtc	388
8.393.2.30 vertConfidence	388
8.393.2.31 vertReliability	388
8.393.2.32 vertUnc	388
8.394 pack_loc_SLQSLOCInjectSensorData_t Struct Reference	388
8.394.1 Detailed Description	388
8.394.2 Field Documentation	389
8.394.2.1 acceleroData	389
8.394.2.2 acceleroTimeSrc	389
8.394.2.3 accelTemp	389
8.394.2.4 gyroData	389

8.394.2.5 gyroTemp	390
8.394.2.6 gyroTimeSrc	390
8.394.2.7 has_acceleroTimeSrc	390
8.394.2.8 has_accelTemp	390
8.394.2.9 has_accleroData	390
8.394.2.10has_gyroData	390
8.394.2.11has_gyroTemp	390
8.394.2.12has_gyroTimeSrc	390
8.394.2.13has_opaqueId	390
8.394.2.14opaqueId	390
8.395pack_loc_SLQSLOCInjectUTCTime_t Struct Reference	390
8.395.1 Detailed Description	390
8.395.2 Field Documentation	390
8.395.2.1 timeMsec	390
8.395.2.2 timeUncMsec	390
8.396pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference	390
8.396.1 Detailed Description	391
8.396.2 Field Documentation	391
8.396.2.1 confidence	391
8.396.2.2 has_confidence	391
8.396.2.3 state	391
8.397pack_loc_SLQSLOCSetServer_t Struct Reference	391
8.397.1 Detailed Description	391
8.397.2 Field Documentation	392
8.397.2.1 pIPv4Config	392
8.397.2.2 pIPv6Config	392
8.397.2.3 pURLAddr	392
8.397.2.4 serverType	392
8.398pack_loc_Start_t Struct Reference	392
8.398.1 Detailed Description	392
8.398.2 Field Documentation	393
8.398.2.1 pApplicationInfo	393
8.398.2.2 pConfigAltitudeAssumed	393
8.398.2.3 pHorizontalAccuracyLvl	393
8.398.2.4 pIntermediateReportState	393
8.398.2.5 pMinIntervalTime	393
8.398.2.6 pRecurrenceType	394
8.398.2.7 SessionId	394
8.398.2.8 Tlvresult	394
8.399pack_loc_Stop_t Struct Reference	394

8.399.1 Detailed Description	394
8.399.2 Field Documentation	394
8.399.2.1 SessionId	394
8.399.2.2 Tlvresult	394
8.400pack_nas_InitiateDomainAttach_t Struct Reference	394
8.400.1 Detailed Description	394
8.400.2 Field Documentation	394
8.400.2.1 action	395
8.401pack_nas_PerformNetworkScanPCI_t Struct Reference	395
8.401.1 Field Documentation	395
8.401.1.1 pCiotOpModePref	395
8.401.1.2 pLteM1BandPref	395
8.401.1.3 pLteNB1BandPref	395
8.401.1.4 pNetworkType	395
8.401.1.5 pScanType	395
8.402pack_nas_SetACCOLC_t Struct Reference	395
8.402.1 Detailed Description	395
8.402.2 Field Documentation	395
8.402.2.1 accolc	395
8.402.2.2 spc	395
8.403pack_nas_SetCDMANetworkParameters_t Struct Reference	395
8.403.1 Detailed Description	396
8.403.2 Field Documentation	397
8.403.2.1 pApplication	397
8.403.2.2 pBroadcast	397
8.403.2.3 pCustomSCP	397
8.403.2.4 pForceRev0	397
8.403.2.5 pProtocol	397
8.403.2.6 pRoaming	397
8.403.2.7 pSPC	397
8.404pack_nas_SetNetworkPreference_t Struct Reference	397
8.404.1 Detailed Description	397
8.404.2 Field Documentation	398
8.404.2.1 Duration	398
8.404.2.2 TechnologyPref	398
8.404.2.3 Tlvresult	398
8.405pack_nas_SLQSCfgSigInfo_t Struct Reference	398
8.405.1 Detailed Description	398
8.405.2 Field Documentation	399
8.405.2.1 pECIOThresh	399

8.405.2.2 pHDRSINRThresh	399
8.405.2.3 pIOTresh	399
8.405.2.4 pLTESigRptCfg	399
8.405.2.5 pLTESNRThresh	399
8.405.2.6 pRSRPThresh	399
8.405.2.7 pRSRQThresh	399
8.405.2.8 pRSSIThresh	399
8.405.2.9 pTDSCDMASINRCONFThresh	399
8.406pack_nas_SLQSGetPLMNName_t Struct Reference	399
8.406.1 Detailed Description	400
8.406.2 Field Documentation	400
8.406.2.1 mcc	400
8.406.2.2 mnc	400
8.406.2.3 pMncPcsStatus	400
8.407pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference	400
8.407.1 Detailed Description	400
8.407.2 Field Documentation	401
8.407.2.1 pChangeDuration	401
8.407.2.2 pMncPcsDigitStatus	401
8.407.2.3 pMNRInfo	401
8.407.2.4 regAction	401
8.408pack_nas_SLQSNasConfigSigInfo2_t Struct Reference	401
8.408.1 Detailed Description	402
8.408.2 Field Documentation	404
8.408.2.1 pCDMAECIODelta	404
8.408.2.2 pCDMAECIOThresh	404
8.408.2.3 pCDMARSSIDelta	405
8.408.2.4 pCDMARSSIThresh	405
8.408.2.5 pGSMRSSIDelta	405
8.408.2.6 pGSMRSSIThresh	405
8.408.2.7 pHDRECIODelta	405
8.408.2.8 pHDRECIOThresh	405
8.408.2.9 pHDRIODelta	405
8.408.2.10pHDRIOTresh	405
8.408.2.11pHDRRSSIDelta	405
8.408.2.12pHDRRSSIThresh	405
8.408.2.13pHDRSINRDelta	405
8.408.2.14pHDRSINRThresh	405
8.408.2.15pLTERSRPDelta	405
8.408.2.16pLTERSRPThresh	405

8.408.2.17	pLTERSRQDelta	405
8.408.2.18	pLTERSRQThresh	405
8.408.2.19	pLTERSSIDelta	405
8.408.2.20	pLTERSSIThresh	405
8.408.2.21	pLTESigRptConfig	405
8.408.2.22	pLTESNRDelta	405
8.408.2.23	pLTESNRThresh	405
8.408.2.24	pTDSCDMAECIODelta	405
8.408.2.25	pTDSCDMAECIOThresh	405
8.408.2.26	pTDSCDMARSCPDelta	405
8.408.2.27	pTDSCDMARSCPThresh	405
8.408.2.28	pTDSCDMARSSIDelta	405
8.408.2.29	pTDSCDMARSSIThresh	405
8.408.2.30	pTDSCDMASINRDelta	405
8.408.2.31	pTDSCDMASINRThresh	406
8.408.2.32	pWCDMAECIODelta	406
8.408.2.33	pWCDMAECIOThresh	406
8.408.2.34	pWCDMARSSIDelta	406
8.408.2.35	pWCDMARSSIThresh	406
8.409	pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference	406
8.409.1	Detailed Description	406
8.409.2	Field Documentation	406
8.409.2.1	namID	406
8.410	pack_nas_SLQSNASGeteDRXParamsExt_t Struct Reference	406
8.410.1	Detailed Description	406
8.410.2	Field Documentation	407
8.410.2.1	pEdrxRAT	407
8.410.2.2	pLteOpMode	407
8.411	pack_nas_SLQSNasGetTxRxInfo_t Struct Reference	407
8.411.1	Detailed Description	407
8.411.2	Field Documentation	408
8.411.2.1	radio_if	408
8.412	pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference	408
8.412.1	Detailed Description	408
8.412.2	Field Documentation	410
8.412.2.1	pDDTMInd	410
8.412.2.2	pDualStandByPrefInd	410
8.412.2.3	pEdrxChangeInfoInd	410
8.412.2.4	pErrorRateInd	410
8.412.2.5	pHDRNewUATIAssInd	410

8.412.2.6 pHDRSessionCloseInd	410
8.412.2.7 pLTECphyCa	410
8.412.2.8 pManagedRoamingInd	411
8.412.2.9 pNetworkRejectInd	411
8.412.2.10 pNetworkTimeInd	411
8.412.2.11 pServingSystemInd	411
8.412.2.12 pSignalStrengthInd	411
8.412.2.13 pSubscriptionInfoInd	411
8.412.2.14 pSuppressSysInfoInd	411
8.412.2.15 pSysInfoInd	411
8.412.2.16 pSystemSelectionInd	411
8.413 pack_nas_SLQSNASSeteDRXParams_t Struct Reference	411
8.413.1 Detailed Description	411
8.413.2 Field Documentation	412
8.413.2.1 pCycleLen	412
8.413.2.2 pEdrxCiotLteMode	412
8.413.2.3 pEdrxEnable	412
8.413.2.4 pEdrxRatType	412
8.413.2.5 pPagingTimeWindow	412
8.414 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference	412
8.414.1 Detailed Description	413
8.414.2 Field Documentation	413
8.414.2.1 gsmUmtsDI	413
8.414.2.2 gsmUmtsUI	413
8.414.2.3 lteEmmDI	413
8.414.2.4 lteEmmUI	413
8.414.2.5 lteEsmDI	413
8.414.2.6 lteEsmUI	414
8.414.2.7 pRankIndicatorInd	414
8.414.2.8 pTimer	414
8.415 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference	414
8.415.1 Detailed Description	414
8.415.2 Field Documentation	414
8.415.2.1 pLteEARFCN	414
8.415.2.2 pLtePCI	414
8.415.2.3 pWcdmaUARFCN	414
8.416 pack_nas_SLQSSetBandPreference_t Struct Reference	414
8.416.1 Detailed Description	414
8.416.2 Field Documentation	415
8.416.2.1 bandPref	416

8.417	pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference	416
8.417.1	Detailed Description	416
8.417.2	Field Documentation	416
8.417.2.1	bEnable	416
8.417.2.2	pSigIndReq	416
8.418	pack_nas_SLQSSetSysSelectionPref_t Struct Reference	416
8.418.1	Detailed Description	416
8.418.2	Field Documentation	421
8.418.2.1	pAcqOrderPref	421
8.418.2.2	pBandPref	421
8.418.2.3	pChgDuration	421
8.418.2.4	pCSGID	421
8.418.2.5	pEmerMode	421
8.418.2.6	pGWAcqOrderPref	421
8.418.2.7	pLTEBandPref	421
8.418.2.8	pMNCIncPCSDigStat	421
8.418.2.9	pModePref	421
8.418.2.10	pNetSelPref	421
8.418.2.11	pPRLPref	421
8.418.2.12	pRAT	421
8.418.2.13	pRoamPref	421
8.418.2.14	pSrvDomainPref	421
8.418.2.15	pSrvRegRestriction	422
8.418.2.16	pTdsdmaBandPref	422
8.419	pack_nas_SLQSSetSysSelectionPrefExt_t Struct Reference	422
8.419.1	Detailed Description	422
8.419.2	Field Documentation	431
8.419.2.1	pAcqOrderPref	431
8.419.2.2	pBandPref	431
8.419.2.3	pChgDuration	431
8.419.2.4	pCiotAcqOrderPref	431
8.419.2.5	pCiotLteOpMode	431
8.419.2.6	pCiotLteOpModePref	431
8.419.2.7	pCSGID	431
8.419.2.8	pEmerMode	431
8.419.2.9	pGWAcqOrderPref	431
8.419.2.10	pLTEBandPref	431
8.419.2.11	pLTEBandPrefExt	431
8.419.2.12	pLteM1BandPref	431
8.419.2.13	pLteNb1BandPref	431

8.419.2.14	pMNCIncPCSDigStat	431
8.419.2.15	pModePref	431
8.419.2.16	pNetSelPref	431
8.419.2.17	pNr5gBandPref	431
8.419.2.18	pPRLPref	432
8.419.2.19	pRAT	432
8.419.2.20	pRoamPref	432
8.419.2.21	pSrvDomainPref	432
8.419.2.22	pSrvRegRestriction	432
8.419.2.23	pTdsdmaBandPref	432
8.420	pack_nas_SLQSSwiPSDetach_t Struct Reference	432
8.420.1	Detailed Description	432
8.420.2	Field Documentation	432
8.420.2.1	pDetachAction	432
8.421	pack_pds_PDSInjectTimeReference_t Struct Reference	432
8.421.1	Detailed Description	432
8.421.2	Field Documentation	433
8.421.2.1	systemDiscontinuities	433
8.421.2.2	systemTime	433
8.422	pack_pds_ResetPDSDData_t Struct Reference	433
8.422.1	Detailed Description	433
8.422.2	Field Documentation	433
8.422.2.1	pCellDataMask	434
8.422.2.2	pGPSDataMask	434
8.423	pack_pds_SetEventReportCallback_t Struct Reference	434
8.423.1	Detailed Description	434
8.423.2	Field Documentation	434
8.423.2.1	posDataNmea	434
8.423.2.2	rptPosData	434
8.424	pack_pds_SetPDSDDefaults_t Struct Reference	434
8.424.1	Detailed Description	434
8.424.2	Field Documentation	435
8.424.2.1	accuracy	435
8.424.2.2	interval	435
8.424.2.3	operation	435
8.424.2.4	timeout	435
8.425	pack_pds_SetPDSSState_t Struct Reference	435
8.425.1	Detailed Description	435
8.425.2	Field Documentation	435
8.425.2.1	enable	435

8.426	pack_pds_SetPortAutomaticTracking_t Struct Reference	435
8.426.1	Detailed Description	436
8.426.2	Field Documentation	436
8.426.2.1	bAuto	436
8.427	pack_pds_SetServiceAutomaticTracking_t Struct Reference	436
8.427.1	Detailed Description	436
8.427.2	Field Documentation	436
8.427.2.1	bAuto	436
8.428	pack_pds_SetXTRAAutomaticDownload_t Struct Reference	436
8.428.1	Detailed Description	437
8.428.2	Field Documentation	437
8.428.2.1	bEnabled	437
8.428.2.2	interval	437
8.429	pack_pds_SetXTRANetwork_t Struct Reference	437
8.429.1	Detailed Description	437
8.429.2	Field Documentation	437
8.429.2.1	preference	437
8.430	pack_pds_SLQSGetAGPSConfig_t Struct Reference	437
8.430.1	Detailed Description	438
8.430.2	Field Documentation	438
8.430.2.1	pNetworkMode	438
8.431	pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference	438
8.431.1	Detailed Description	438
8.431.2	Field Documentation	438
8.431.2.1	forceFlag	438
8.431.2.2	timeBase	438
8.431.2.3	timeMsec	439
8.431.2.4	timeUncMsec	439
8.432	pack_pds_SLQSPDSInjectPositionData_t Struct Reference	439
8.432.1	Detailed Description	439
8.432.2	Field Documentation	440
8.432.2.1	pAltitudeWrtEllipsoid	440
8.432.2.2	pAltitudeWrtSealevel	440
8.432.2.3	pHorizontalConfidence	440
8.432.2.4	pHorizontalUncCircular	440
8.432.2.5	pLatitude	440
8.432.2.6	pLongitude	440
8.432.2.7	pPositionSource	440
8.432.2.8	pTimeStamp	440
8.432.2.9	pTimeType	440

8.432.2.1qVerticalConfidence	440
8.432.2.1pVerticalUnc	441
8.433pack_pds_SLQSSetAGPSConfig_t Struct Reference	441
8.433.1 Detailed Description	441
8.433.2 Field Documentation	441
8.433.2.1 pNetworkMode	441
8.433.2.2 pServerAddress	441
8.433.2.3 pServerPort	441
8.433.2.4 pServerURL	441
8.433.2.5 pServerURLLength	441
8.434pack_pds_SLQSSetPositionMethodState_t Struct Reference	441
8.434.1 Detailed Description	442
8.434.2 Field Documentation	442
8.434.2.1 pWifiState	442
8.434.2.2 pXtraDataState	442
8.434.2.3 pXtraTimeState	442
8.435pack_pds_StartPDSTrackingSessionExt_t Struct Reference	442
8.435.1 Detailed Description	442
8.435.2 Field Documentation	443
8.435.2.1 fixAccuracy	443
8.435.2.2 fixCount	443
8.435.2.3 fixInterval	443
8.435.2.4 fixTimeout	443
8.435.2.5 sessionControl	443
8.435.2.6 sessionOperation	443
8.435.2.7 sessionServerOption	443
8.435.2.8 sessionType	443
8.436pack_qmi_t Struct Reference	443
8.436.1 Detailed Description	444
8.436.2 Field Documentation	444
8.436.2.1 msgid	444
8.436.2.2 svc	444
8.436.2.3 timeout	444
8.436.2.4 xid	444
8.437pack_qos_BindDataPort_t Struct Reference	444
8.437.1 Detailed Description	444
8.437.2 Field Documentation	445
8.437.2.1 pMuxID	445
8.437.2.2 pPeripheralEndPointID	445
8.437.2.3 pSIODataPort	445

8.438	pack_qos_SLQSQoSswiReadApnExtraParams_t Struct Reference	445
8.438.1	Detailed Description	445
8.438.2	Field Documentation	445
8.438.2.1	apnId	445
8.439	pack_qos_SLQSQoSswiReadDataStats_t Struct Reference	445
8.439.1	Detailed Description	445
8.439.2	Field Documentation	445
8.439.2.1	apnId	445
8.440	pack_qos_SLQSSetQoSEventCallback_t Struct Reference	445
8.440.1	Detailed Description	446
8.440.2	Field Documentation	446
8.440.2.1	enable	446
8.441	pack_rms_SetSMSWake_t Struct Reference	446
8.441.1	Detailed Description	446
8.441.2	Field Documentation	446
8.441.2.1	enabled	446
8.441.2.2	wake_mask	446
8.442	pack_sar_SLQSSetRfSarState_t Struct Reference	446
8.442.1	Detailed Description	447
8.442.2	Field Documentation	447
8.442.2.1	RfSarState	447
8.443	pack_sms_SaveSMS_t Struct Reference	447
8.443.1	Detailed Description	447
8.443.2	Field Documentation	448
8.443.2.1	messageFormat	448
8.443.2.2	messageSize	448
8.443.2.3	pMessage	448
8.443.2.4	storageType	448
8.444	pack_sms_SendSMS_t Struct Reference	448
8.444.1	Detailed Description	448
8.444.2	Field Documentation	448
8.444.2.1	messageFormat	448
8.444.2.2	messageSize	448
8.444.2.3	pLinktimer	449
8.444.2.4	pMessage	449
8.445	pack_sms_SetNewSMSCallback_t Struct Reference	449
8.445.1	Detailed Description	449
8.445.2	Field Documentation	449
8.445.2.1	status	449
8.446	pack_sms_SetSMSCAddress_t Struct Reference	449

8.446.1 Detailed Description	449
8.446.2 Field Documentation	449
8.446.2.1 pSMSCAddress	449
8.446.2.2 pSMSCType	449
8.447pack_sms_SLQSDeleteSMS_t Struct Reference	449
8.447.1 Detailed Description	450
8.447.2 Field Documentation	450
8.447.2.1 pMessageIndex	450
8.447.2.2 pMessageMode	450
8.447.2.3 pMessageTag	450
8.447.2.4 storageType	450
8.448pack_sms_SLQSGetSMS_t Struct Reference	450
8.448.1 Detailed Description	451
8.448.2 Field Documentation	451
8.448.2.1 messageIndex	451
8.448.2.2 pMessageMode	451
8.448.2.3 storageType	451
8.449pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference	451
8.449.1 Detailed Description	451
8.449.2 Field Documentation	451
8.449.2.1 mode	451
8.450pack_sms_SLQSGetSMSList_t Struct Reference	451
8.450.1 Detailed Description	452
8.450.2 Field Documentation	452
8.450.2.1 pMessageMode	452
8.450.2.2 pRequestedTag	452
8.450.2.3 storageType	452
8.451pack_sms_SLQSModifySMSStatus_t Struct Reference	452
8.451.1 Detailed Description	452
8.451.2 Field Documentation	453
8.451.2.1 messageIndex	453
8.451.2.2 messageTag	453
8.451.2.3 pMessageMode	453
8.451.2.4 storageType	453
8.452pack_sms_SLQSSendAsyncSMS_t Struct Reference	453
8.452.1 Detailed Description	453
8.452.2 Field Documentation	453
8.452.2.1 pSendSmsParams	453
8.453pack_sms_SLQSSetIndicationRegister_t Struct Reference	453
8.453.1 Detailed Description	454

8.453.2 Field Documentation	454
8.453.2.1 pSetIndicationRegReq	454
8.454pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference	454
8.454.1 Detailed Description	454
8.454.2 Field Documentation	454
8.454.2.1 broadcastActivate	454
8.454.2.2 mode	454
8.455pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference	454
8.455.1 Detailed Description	455
8.455.2 Field Documentation	455
8.455.2.1 mode	455
8.455.2.2 pBroadcastConfig	455
8.455.2.3 pCDMABroadcastConfig	455
8.456pack_sms_SLQSSetSmsStorage_t Struct Reference	455
8.456.1 Detailed Description	455
8.456.2 Field Documentation	455
8.456.2.1 smsStorage	456
8.457pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference	456
8.457.1 Detailed Description	456
8.457.2 Field Documentation	456
8.457.2.1 pMaxStorageSizeReq	456
8.458pack_sms_SLQSSmsSetRoutes_t Struct Reference	456
8.458.1 Detailed Description	456
8.458.2 Field Documentation	456
8.458.2.1 pSetRoutesReq	456
8.459pack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference	456
8.459.1 Detailed Description	457
8.459.2 Field Documentation	457
8.459.2.1 pGenerator	457
8.460pack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference	457
8.460.1 Detailed Description	457
8.460.2 Field Documentation	457
8.460.2.1 Generator	457
8.460.2.2 Profile	457
8.461pack_swiaudio_SLQSGetM2MAVMute_t Struct Reference	457
8.461.1 Detailed Description	457
8.461.2 Field Documentation	458
8.461.2.1 Profile	458
8.462pack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference	458
8.462.1 Detailed Description	458

8.462.2 Field Documentation	458
8.462.2.1 Profile	458
8.463pack_swiaudio_SLQSSetM2MAudioAVCFG_t Struct Reference	458
8.463.1 Detailed Description	458
8.463.2 Field Documentation	459
8.463.2.1 Device	459
8.463.2.2 PIFACEId	459
8.463.2.3 pPCMPParams	459
8.463.2.4 Profile	459
8.464pack_swiaudio_SLQSSetM2MAudioLPBK_t Struct Reference	459
8.464.1 Detailed Description	459
8.464.2 Field Documentation	459
8.464.2.1 Enable	459
8.465pack_swiaudio_SLQSSetM2MAudioProfile_t Struct Reference	459
8.465.1 Detailed Description	460
8.465.2 Field Documentation	460
8.465.2.1 pCwtMute	460
8.465.2.2 pEarMute	460
8.465.2.3 pGenerator	460
8.465.2.4 pMicMute	460
8.465.2.5 Profile	460
8.465.2.6 pVolume	460
8.466pack_swiaudio_SLQSSetM2MAudioVolume_t Struct Reference	461
8.466.1 Detailed Description	461
8.466.2 Field Documentation	461
8.466.2.1 Generator	461
8.466.2.2 Level	461
8.466.2.3 Profile	461
8.467pack_swiaudio_SLQSSetM2MAVMute_t Struct Reference	461
8.467.1 Detailed Description	461
8.467.2 Field Documentation	462
8.467.2.1 EarMute	462
8.467.2.2 MicMute	462
8.467.2.3 pCwtMute	462
8.467.2.4 Profile	462
8.468pack_swiaudio_SLQSSetM2MSpkrGain_t Struct Reference	462
8.468.1 Detailed Description	462
8.468.2 Field Documentation	462
8.468.2.1 Profile	462
8.468.2.2 Value	462

8.469	pack_swiaavms_SLQSAVMSSendSelection_t Struct Reference	463
8.469.1	Detailed Description	463
8.469.2	Field Documentation	463
8.469.2.1	pClientPerformOperationFlag	463
8.469.2.2	pDeferTime	463
8.469.2.3	pPackageID	463
8.469.2.4	pRejectReason	463
8.469.2.5	selection	463
8.470	pack_swiaavms_SLQSAVMSSetSettings_t Struct Reference	463
8.470.1	Detailed Description	464
8.470.2	Field Documentation	465
8.470.2.1	AutoConnect	465
8.470.2.2	AutoReboot	465
8.470.2.3	pAPNInfo	465
8.470.2.4	pConnectionRetryTimers	465
8.470.2.5	pFwAutoSDM	465
8.470.2.6	pNotifStore	465
8.470.2.7	pPeriodInfo	465
8.470.2.8	pPollingTimer	465
8.470.2.9	PromptFwDownload	465
8.470.2.10	PromptFwUpdate	465
8.471	pack_swiaavms_SLQSAVMSSetSettings_v2_t Struct Reference	465
8.471.1	Detailed Description	465
8.471.2	Field Documentation	466
8.471.2.1	AutoConnect	467
8.471.2.2	pAPNInfo	467
8.471.2.3	pAutoReboot	467
8.471.2.4	pConnectionRetryTimers	467
8.471.2.5	pFwAutoSDM	467
8.471.2.6	pNotifStore	467
8.471.2.7	pPeriodInfo	467
8.471.2.8	pPollingTimer	467
8.471.2.9	PromptFwDownload	467
8.471.2.10	PromptFwUpdate	467
8.472	pack_swiaavms_SLQSAVMSStartSession_t Struct Reference	467
8.472.1	Detailed Description	467
8.472.2	Field Documentation	467
8.472.2.1	sessionType	467
8.473	pack_swiaavms_SLQSAVMSStopSession_t Struct Reference	467
8.473.1	Detailed Description	467

8.473.2 Field Documentation	468
8.473.2.1 sessionType	468
8.474pack_swidms_SLQSSwiDmsSetHWWatchdog_t Struct Reference	468
8.474.1 Detailed Description	468
8.474.2 Field Documentation	468
8.474.2.1 enable	468
8.474.2.2 resetDelay	468
8.474.2.3 timeout	468
8.475pack_swidms_SLQSSwiDmsSetMTU_t Struct Reference	468
8.475.1 Detailed Description	469
8.475.2 Field Documentation	469
8.475.2.1 MTUSize	469
8.476pack_swidms_SLQSSwiDmsSetUsbComp_t Struct Reference	469
8.476.1 Detailed Description	469
8.476.2 Field Documentation	470
8.476.2.1 CfgValue	470
8.477pack_swidms_SLQSSwiDmsSetUsbNetNum_t Struct Reference	470
8.477.1 Detailed Description	470
8.477.2 Field Documentation	470
8.477.2.1 nUsbNetNum	471
8.478pack_swiloc_SwiLocSetAutoStart_t Struct Reference	471
8.478.1 Detailed Description	471
8.478.2 Field Documentation	472
8.478.2.1 fix_rate	472
8.478.2.2 fix_type	472
8.478.2.3 function	472
8.478.2.4 max_dist	472
8.478.2.5 max_time	472
8.478.2.6 set_fix_rate	472
8.478.2.7 set_fix_type	472
8.478.2.8 set_function	472
8.478.2.9 set_max_dist	472
8.478.2.10set_max_time	472
8.479pack_swioma_SLQSOMADMCancelSession_t Struct Reference	472
8.479.1 Detailed Description	472
8.479.2 Field Documentation	473
8.479.2.1 sessionType	473
8.480pack_swioma_SLQSOMADMCancelSessionExt_t Struct Reference	473
8.480.1 Detailed Description	473
8.480.2 Field Documentation	473

8.480.2.1 sessionType	473
8.481pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference	473
8.481.1 Detailed Description	473
8.481.2 Field Documentation	474
8.481.2.1 SessionType	474
8.482pack_swioama_SLQSOMADMSendSelection_t Struct Reference	474
8.482.1 Detailed Description	474
8.482.2 Field Documentation	474
8.482.2.1 pDeferTime	474
8.482.2.2 pRejectReason	474
8.482.2.3 selection	474
8.483pack_swioama_SLQSOMADMSendSelectionExt_t Struct Reference	474
8.483.1 Detailed Description	474
8.483.2 Field Documentation	475
8.483.2.1 selection	475
8.484pack_swioama_SLQSOMADMSetSettings_t Struct Reference	475
8.484.1 Detailed Description	475
8.484.2 Field Documentation	476
8.484.2.1 FOTAdownload	476
8.484.2.2 FOTAUpdate	476
8.484.2.3 pAutosdm	476
8.484.2.4 pFwAutoCheck	476
8.485pack_swioama_SLQSOMADMSetSettingsExt_t Struct Reference	476
8.485.1 Detailed Description	476
8.485.2 Field Documentation	477
8.485.2.1 FOTAdownload	477
8.485.2.2 FOTAUpdate	477
8.485.2.3 FUMOEnable	477
8.485.2.4 OMADMEEnable	477
8.485.2.5 OMADMLogEnable	477
8.485.2.6 PRLEnable	477
8.486pack_swioama_SLQSOMADMStartSession_t Struct Reference	477
8.486.1 Detailed Description	477
8.486.2 Field Documentation	477
8.486.2.1 sessionType	477
8.487pack_swioama_SLQSOMADMStartSessionExt_t Struct Reference	477
8.487.1 Detailed Description	478
8.487.2 Field Documentation	478
8.487.2.1 sessionType	478
8.488pack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference	478

8.488.1 Detailed Description	478
8.488.2 Field Documentation	478
8.488.2.1 mitigationDevID	478
8.488.2.2 mitigationDevIDLen	478
8.489pack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference	478
8.489.1 Detailed Description	479
8.489.2 Field Documentation	479
8.489.2.1 mitigationDevID	479
8.489.2.2 mitigationDevIDLen	479
8.490pack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference	479
8.490.1 Detailed Description	479
8.490.2 Field Documentation	479
8.490.2.1 mitigationDevID	479
8.490.2.2 mitigationDevIDLen	479
8.491pack_uim_ChangePin_t Struct Reference	479
8.491.1 Detailed Description	480
8.491.2 Field Documentation	480
8.491.2.1 changePIN	480
8.491.2.2 EncryptedPIN1	480
8.491.2.3 pIndicationToken	480
8.491.2.4 pKeyReferenceID	480
8.491.2.5 sessionInfo	480
8.491.2.6 Tlvresult	480
8.492pack_uim_ReadTransparent_t Struct Reference	480
8.492.1 Detailed Description	481
8.492.2 Field Documentation	481
8.492.2.1 fileIndex	481
8.492.2.2 pEncryptData	481
8.492.2.3 pIndicationToken	481
8.492.2.4 readTransparent	481
8.492.2.5 sessionInfo	481
8.492.2.6 Tlvresult	481
8.493pack_uim_SetPinProtection_t Struct Reference	482
8.493.1 Detailed Description	482
8.493.2 Field Documentation	482
8.493.2.1 EncryptedPIN1	482
8.493.2.2 pIndicationToken	482
8.493.2.3 pinProtection	482
8.493.2.4 pKeyReferenceID	482
8.493.2.5 sessionInfo	482

8.493.2.6 Tlvresult	483
8.494pack_uim_SLQSUIMAuthenticate_t Struct Reference	483
8.494.1 Detailed Description	483
8.494.2 Field Documentation	483
8.494.2.1 authData	483
8.494.2.2 pIndicationToken	483
8.494.2.3 sessionInfo	483
8.495pack_uim_SLQSUIDepersonalization_t Struct Reference	483
8.495.1 Detailed Description	483
8.495.2 Field Documentation	484
8.495.2.1 depersonilisationInfo	484
8.496pack_uim_SLQSUIEventRegister_t Struct Reference	484
8.496.1 Detailed Description	484
8.496.2 Field Documentation	484
8.496.2.1 eventMask	484
8.497pack_uim_SLQSUIGetConfiguration_t Struct Reference	484
8.497.1 Detailed Description	484
8.497.2 Field Documentation	484
8.497.2.1 pConfigurationMask	484
8.498pack_uim_SLQSUIGetFileAttributes_t Struct Reference	485
8.498.1 Detailed Description	485
8.498.2 Field Documentation	485
8.498.2.1 fileIndex	485
8.498.2.2 pIndicationToken	485
8.498.2.3 sessionInfo	485
8.499pack_uim_SLQSUIPowerDown_t Struct Reference	485
8.499.1 Detailed Description	485
8.499.2 Field Documentation	486
8.499.2.1 slot	486
8.500pack_uim_SLQSUIPowerUp_t Struct Reference	486
8.500.1 Detailed Description	486
8.500.2 Field Documentation	486
8.500.2.1 plgnoreHotSwapSwitch	486
8.500.2.2 slot	486
8.501pack_uim_SLQSUIRefreshComplete_t Struct Reference	486
8.501.1 Detailed Description	486
8.501.2 Field Documentation	487
8.501.2.1 refreshComplete	487
8.501.2.2 sessionInfo	487
8.502pack_uim_SLQSUIRefreshGetLastEvent_t Struct Reference	487

8.502.1 Detailed Description	487
8.502.2 Field Documentation	487
8.502.2.1 sessionInfo	487
8.503pack_uim_SLQSUIRefreshOK_t Struct Reference	487
8.503.1 Detailed Description	487
8.503.2 Field Documentation	488
8.503.2.1 OKtoRefresh	488
8.503.2.2 sessionInfo	488
8.504pack_uim_SLQSUIRefreshRegister_t Struct Reference	488
8.504.1 Detailed Description	488
8.504.2 Field Documentation	488
8.504.2.1 regRefresh	488
8.504.2.2 sessionInfo	488
8.505pack_uim_SLQSUISwitchSlot_t Struct Reference	488
8.505.1 Detailed Description	488
8.505.2 Field Documentation	489
8.505.2.1 bLogicalSlot	489
8.505.2.2 ulPhysicalSlot	489
8.506pack_uim_UnblockPin_t Struct Reference	489
8.506.1 Detailed Description	489
8.506.2 Field Documentation	490
8.506.2.1 EncryptedPIN1	490
8.506.2.2 pIndicationToken	490
8.506.2.3 pinProtection	490
8.506.2.4 pKeyReferenceID	490
8.506.2.5 sessionInfo	490
8.506.2.6 Tlvresult	490
8.507pack_uim_VerifyPin_t Struct Reference	490
8.507.1 Detailed Description	490
8.507.2 Field Documentation	491
8.507.2.1 pEncryptedPIN1	491
8.507.2.2 pIndicationToken	491
8.507.2.3 pKeyReferenceID	491
8.507.2.4 sessionInfo	491
8.507.2.5 Tlvresult	491
8.507.2.6 verifyPIN	491
8.508pack_voice_AnswerUSSD_t Struct Reference	491
8.508.1 Detailed Description	491
8.508.2 Field Documentation	492
8.508.2.1 pInfo	492

8.509pack_voice_OriginateUSSD_t Struct Reference	492
8.509.1 Detailed Description	492
8.509.2 Field Documentation	492
8.509.2.1 pInfo	492
8.510pack_voice_SLQSOriinateUSSD_t Struct Reference	492
8.510.1 Detailed Description	492
8.510.2 Field Documentation	493
8.510.2.1 ussData	493
8.510.2.2 ussDCS	493
8.510.2.3 ussLen	493
8.511pack_voice_SLQSVoiceALSSelectLine_t Struct Reference	493
8.511.1 Detailed Description	493
8.511.2 Field Documentation	493
8.511.2.1 lineValue	493
8.512pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference	493
8.512.1 Detailed Description	493
8.512.2 Field Documentation	493
8.512.2.1 switchOption	493
8.513pack_voice_SLQSVoiceAnswerCall_t Struct Reference	494
8.513.1 Detailed Description	494
8.513.2 Field Documentation	494
8.513.2.1 pCallId	494
8.514pack_voice_SLQSVoiceBindSubscription_t Struct Reference	494
8.514.1 Detailed Description	494
8.514.2 Field Documentation	494
8.514.2.1 subType	494
8.515pack_voice_SLQSVoiceBurstDTMF_t Struct Reference	494
8.515.1 Detailed Description	495
8.515.2 Field Documentation	495
8.515.2.1 BurstDTMFInfo	495
8.515.2.2 pBurstDTMFLengths	495
8.516pack_voice_SLQSVoiceDialCall_t Struct Reference	495
8.516.1 Detailed Description	495
8.516.2 Field Documentation	496
8.516.2.1 callNumber	496
8.516.2.2 pCallPartySubAdd	496
8.516.2.3 pCallType	496
8.516.2.4 pCLIRType	496
8.516.2.5 pCUGInfo	496
8.516.2.6 pEmergencyCategory	497

8.516.2.7 pSvcType	497
8.516.2.8 pUUSnFo	497
8.517pack_voice_SLQSVoiceEndCall_t Struct Reference	497
8.517.1 Detailed Description	497
8.517.2 Field Documentation	497
8.517.2.1 pCallId	497
8.518pack_voice_SLQSVoiceGetCallBarring_t Struct Reference	497
8.518.1 Detailed Description	497
8.518.2 Field Documentation	498
8.518.2.1 pSvcClass	498
8.518.2.2 reason	498
8.519pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference	498
8.519.1 Detailed Description	498
8.519.2 Field Documentation	499
8.519.2.1 pSvcClass	499
8.519.2.2 Reason	499
8.520pack_voice_SLQSVoiceGetCallInfo_t Struct Reference	499
8.520.1 Detailed Description	499
8.520.2 Field Documentation	499
8.520.2.1 callID	499
8.521pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference	499
8.521.1 Detailed Description	500
8.521.2 Field Documentation	500
8.521.2.1 pSvcClass	500
8.522pack_voice_SLQSVoiceGetConfig_t Struct Reference	500
8.522.1 Detailed Description	500
8.522.2 Field Documentation	501
8.522.2.1 pAirTimer	501
8.522.2.2 pAMRStatus	501
8.522.2.3 pAutoAnswer	501
8.522.2.4 pNamID	501
8.522.2.5 pPrefVoicePrivacy	501
8.522.2.6 pPrefVoiceSO	501
8.522.2.7 pRoamTimer	501
8.522.2.8 pTTYMode	501
8.522.2.9 pVoiceDomainPref	501
8.523pack_voice_SLQSVoiceIndicationRegister_t Struct Reference	502
8.523.1 Detailed Description	502
8.523.2 Field Documentation	502
8.523.2.1 pRegDTMFEvents	502

8.523.2.2 pRegVoicePrivacyEvents	502
8.523.2.3 pSuppsNotifEvents	502
8.524pack_voice_SLQSVoiceManageCalls_t Struct Reference	502
8.524.1 Detailed Description	503
8.524.2 Field Documentation	503
8.524.2.1 pCallID	503
8.524.2.2 SUPSType	503
8.525pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference	503
8.525.1 Detailed Description	503
8.525.2 Field Documentation	504
8.525.2.1 USSInformation	504
8.526pack_voice_SLQSVoiceSendFlash_t Struct Reference	504
8.526.1 Detailed Description	504
8.526.2 Field Documentation	504
8.526.2.1 pCallID	504
8.526.2.2 pFlashPayLd	504
8.526.2.3 pFlashType	504
8.527pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference	504
8.527.1 Detailed Description	505
8.527.2 Field Documentation	505
8.527.2.1 newPasswd	505
8.527.2.2 newPasswdAgain	505
8.527.2.3 oldPasswd	505
8.527.2.4 Reason	505
8.528pack_voice_SLQSVoiceSetConfig_t Struct Reference	505
8.528.1 Detailed Description	506
8.528.2 Field Documentation	507
8.528.2.1 pAirTimerConfig	507
8.528.2.2 pAutoAnswer	507
8.528.2.3 pPrefVoiceDomain	507
8.528.2.4 pPrefVoiceSO	507
8.528.2.5 pRoamTimerConfig	507
8.528.2.6 pTTYMode	507
8.529pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference	507
8.529.1 Detailed Description	507
8.529.2 Field Documentation	507
8.529.2.1 privacyPref	507
8.530pack_voice_SLQSVoiceSetSUPSService_t Struct Reference	507
8.530.1 Detailed Description	507
8.530.2 Field Documentation	509

8.530.2.1 pCallBarringPasswd	509
8.530.2.2 pCallForwardingNumber	509
8.530.2.3 pCallFwdTypeAndPlan	509
8.530.2.4 pServiceClass	509
8.530.2.5 pTimerVal	509
8.530.2.6 reason	509
8.530.2.7 voiceSvc	509
8.531pack_voice_SLQSVoiceStartContDTMF_t Struct Reference	509
8.531.1 Detailed Description	509
8.531.2 Field Documentation	510
8.531.2.1 DTMFdigit	510
8.531.2.2 pCallID	510
8.532pack_voice_SLQSVoiceStopContDTMF_t Struct Reference	510
8.532.1 Detailed Description	510
8.532.2 Field Documentation	510
8.532.2.1 callID	510
8.533pack_wds_DHCPv4ClientLeaseChange_t Struct Reference	510
8.533.1 Detailed Description	510
8.533.2 Field Documentation	511
8.533.2.1 pEnableNotification	511
8.534pack_wds_GetDefaultProfile_t Struct Reference	511
8.534.1 Detailed Description	511
8.534.2 Field Documentation	511
8.534.2.1 profiletype	511
8.535pack_wds_GetDefaultProfileNum_t Struct Reference	511
8.535.1 Detailed Description	511
8.535.2 Field Documentation	511
8.535.2.1 family	511
8.535.2.2 type	512
8.536pack_wds_GetDefaultProfileV2_t Struct Reference	512
8.536.1 Detailed Description	512
8.536.2 Field Documentation	512
8.536.2.1 profiletype	512
8.537pack_wds_GetDormancyState_t Struct Reference	512
8.537.1 Detailed Description	512
8.538pack_wds_GetLastMobileIPError_t Struct Reference	512
8.538.1 Detailed Description	512
8.539pack_wds_GetMobileIP_t Struct Reference	512
8.539.1 Detailed Description	513
8.540pack_wds_GetMobileIPProfile_t Struct Reference	513

8.540.1 Detailed Description	513
8.540.2 Field Documentation	513
8.540.2.1 index	513
8.541 pack_wds_GetPacketStatistics_t Struct Reference	513
8.541.1 Detailed Description	513
8.541.2 Field Documentation	514
8.541.2.1 pStatMask	514
8.542 pack_wds_GetPacketStatus_t Struct Reference	514
8.542.1 Detailed Description	514
8.542.2 Field Documentation	514
8.542.2.1 statmask	514
8.543 pack_wds_GetSessionDuration_t Struct Reference	514
8.543.1 Detailed Description	514
8.544 pack_wds_RMSetTransferStatistics_t Struct Reference	514
8.544.1 Detailed Description	515
8.544.2 Field Documentation	515
8.544.2.1 RmTrasnferStaticsReq	515
8.545 pack_wds_SetAutoconnect_t Struct Reference	515
8.545.1 Detailed Description	515
8.545.2 Field Documentation	515
8.545.2.1 acroamsetting	515
8.545.2.2 acsetting	515
8.546 pack_wds_SetDefaultProfile_t Struct Reference	515
8.546.1 Detailed Description	516
8.546.2 Field Documentation	517
8.546.2.1 authentication	517
8.546.2.2 ipAddress	517
8.546.2.3 pApnname	517
8.546.2.4 pdpType	517
8.546.2.5 pName	517
8.546.2.6 pPassword	517
8.546.2.7 primaryDNS	517
8.546.2.8 profileType	517
8.546.2.9 pUsername	517
8.546.2.10secondaryDNS	517
8.547 pack_wds_SetDefaultProfileNum_t Struct Reference	517
8.547.1 Detailed Description	517
8.547.2 Field Documentation	518
8.547.2.1 family	518
8.547.2.2 index	518

8.547.2.3 type	518
8.548pack_wds_SetMobileIP_t Struct Reference	518
8.548.1 Detailed Description	518
8.548.2 Field Documentation	518
8.548.2.1 mode	518
8.549pack_wds_SetMobileIPParameters_t Struct Reference	518
8.549.1 Detailed Description	518
8.549.2 Field Documentation	519
8.549.2.1 pHA2002bis	519
8.549.2.2 pHAAAuthenticator	519
8.549.2.3 pMode	519
8.549.2.4 pReRegPeriod	519
8.549.2.5 pReRegTraffic	519
8.549.2.6 pRetryInterval	519
8.549.2.7 pRetryLimit	519
8.549.2.8 pSPC	519
8.550pack_wds_SetMobileIPProfile_t Struct Reference	519
8.550.1 Detailed Description	520
8.550.2 Field Documentation	521
8.550.2.1 index	521
8.550.2.2 pAAASPI	521
8.550.2.3 pAddress	521
8.550.2.4 pEnabled	521
8.550.2.5 pHASPI	521
8.550.2.6 pMNAAA	521
8.550.2.7 pMNHA	521
8.550.2.8 pNAI	521
8.550.2.9 pPrimaryHA	521
8.550.2.10pRevTunneling	521
8.550.2.11pSecondaryHA	521
8.550.2.12pc	521
8.551pack_wds_SLQSCreateProfile_t Struct Reference	521
8.551.1 Detailed Description	521
8.551.2 Field Documentation	522
8.551.2.1 pCurProfile	522
8.551.2.2 pProfileId	522
8.551.2.3 pProfileType	522
8.552pack_wds_SLQSDeleteProfile_t Struct Reference	522
8.552.1 Detailed Description	522
8.552.2 Field Documentation	522

8.552.2.1 profileIndex	523
8.552.2.2 profileType	523
8.553pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	523
8.553.1 Detailed Description	523
8.554pack_wds_SLQSGetDataBearerTechnology_t Struct Reference	523
8.554.1 Detailed Description	523
8.555pack_wds_SLQSGetDUNCallInfo_t Struct Reference	523
8.555.1 Detailed Description	523
8.555.2 Field Documentation	524
8.555.2.1 Mask	524
8.555.2.2 pReportChannelRate	524
8.555.2.3 pReportConnStatus	524
8.555.2.4 pReportDataBearerTech	524
8.555.2.5 pReportDormStatus	524
8.555.2.6 pTransferStatInd	524
8.556pack_wds_SLQSGetProfileSettings_t Struct Reference	525
8.556.1 Detailed Description	525
8.556.2 Field Documentation	525
8.556.2.1 ProfileId	525
8.556.2.2 ProfileType	525
8.557pack_wds_SLQSGetRuntimeSettings_t Struct Reference	525
8.557.1 Detailed Description	525
8.557.2 Field Documentation	526
8.557.2.1 pReqSettings	526
8.558pack_wds_SLQSModifyProfile_t Struct Reference	526
8.558.1 Detailed Description	526
8.558.2 Field Documentation	527
8.558.2.1 curProfile	527
8.558.2.2 pProfileId	527
8.558.2.3 pProfileType	527
8.559pack_wds_SLQSSet3GPPConfigItem_t Struct Reference	527
8.559.1 Detailed Description	527
8.559.2 Field Documentation	528
8.559.2.1 LTEAttachProfileListLen	528
8.559.2.2 p3gppRelease	528
8.559.2.3 pDefaultPDNEnabled	528
8.559.2.4 pLTEAttachProfile	528
8.559.2.5 pLTEAttachProfileList	528
8.559.2.6 pProfileList	529
8.560pack_wds_SLQSSetIPFamilyPreference_t Struct Reference	529

8.560.1 Detailed Description	529
8.560.2 Field Documentation	529
8.560.2.1 IPFamilyPreference	529
8.561 pack_wds_SLQSSetWdsEventCallback_t Struct Reference	529
8.561.1 Detailed Description	529
8.561.2 Field Documentation	530
8.561.2.1 currentDataBearer	530
8.561.2.2 dataBearer	530
8.561.2.3 dataBearerTechExt	530
8.561.2.4 dataSystemStatus	530
8.561.2.5 dormancyStatus	530
8.561.2.6 interval	530
8.561.2.7 mobileIP	530
8.561.2.8 transferStats	530
8.562 pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference	530
8.562.1 Detailed Description	530
8.562.2 Field Documentation	530
8.562.2.1 pProfileId	530
8.563 pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference	530
8.563.1 Detailed Description	530
8.563.2 Field Documentation	531
8.563.2.1 pHwConfig	531
8.563.2.2 pProfileId	531
8.563.2.3 pRequestOptionList	531
8.564 pack_wds_SLQSSetLoopback_t Struct Reference	531
8.564.1 Detailed Description	531
8.564.2 Field Documentation	531
8.564.2.1 loopbackMode	531
8.564.2.2 loopbackMultiplier	531
8.565 pack_wds_SLQSSetDataSession_t Struct Reference	531
8.565.1 Detailed Description	532
8.565.2 Field Documentation	532
8.565.2.1 pAuth	532
8.565.2.2 pPass	532
8.565.2.3 pprofileid3gpp	532
8.565.2.4 pprofileid3gpp2	532
8.565.2.5 pTech	532
8.565.2.6 pUser	533
8.566 pack_wds_SLQSSetDataSession_t Struct Reference	533
8.566.1 Detailed Description	533

8.566.2 Field Documentation	533
8.566.2.1 psid	533
8.567 pack_wds_SLQSWdsSetEventReport_t Struct Reference	533
8.567.1 Detailed Description	533
8.567.2 Field Documentation	534
8.567.2.1 pCurrChannelRateInd	534
8.567.2.2 pCurrDataBearerTechInd	534
8.567.2.3 pCurrPrefDataSysInd	534
8.567.2.4 pDataBearerTechInd	534
8.567.2.5 pDataCallStatusChangeInd	534
8.567.2.6 pDataSystemStatusChangeInd	535
8.567.2.7 pDormancyStatusInd	535
8.567.2.8 pEVDOPageMonPerChangeInd	535
8.567.2.9 pMIPStatusInd	535
8.567.2.10 pTransferStatInd	535
8.568 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	535
8.568.1 Detailed Description	535
8.568.2 Field Documentation	535
8.568.2.1 contextId	535
8.568.2.2 contextType	535
8.569 PackCreateProfileOut Struct Reference	535
8.569.1 Detailed Description	535
8.569.2 Field Documentation	536
8.569.2.1 ExtErrorCode	536
8.569.2.2 ProfileIndex	536
8.569.2.3 ProfileType	536
8.570 packgetDyingGaspCfg Struct Reference	536
8.570.1 Detailed Description	536
8.570.2 Field Documentation	536
8.570.2.1 pDestSMSContent	536
8.570.2.2 pDestSMSNum	536
8.571 packgetDyingGaspStatistics Struct Reference	536
8.571.1 Detailed Description	537
8.571.2 Field Documentation	537
8.571.2.1 pSMSAttemptedFlag	537
8.571.2.2 pTimeStamp	537
8.572 PackSwiAvmsSetSettingsAPNInfo Struct Reference	537
8.572.1 Detailed Description	537
8.572.2 Field Documentation	538
8.572.2.1 bAPNLength	538

8.572.2.2 bPWDLenght	538
8.572.2.3 bUnameLength	538
8.572.2.4 szAPN	538
8.572.2.5 szPWD	538
8.572.2.6 szUname	538
8.573PackSwiAvmsSetSettingsConnectionRetryTimers Struct Reference	538
8.573.1 Detailed Description	538
8.573.2 Field Documentation	538
8.573.2.1 Timers	539
8.574PackSwiAvmsSetSettingsPeriodInfo Struct Reference	539
8.574.1 Detailed Description	539
8.574.2 Field Documentation	539
8.574.2.1 ulMax	539
8.574.2.2 ulMin	539
8.575PackSwiAVMSSettingsAPNInfo Struct Reference	539
8.575.1 Detailed Description	539
8.575.2 Field Documentation	539
8.575.2.1 bAPNLength	539
8.575.2.2 bPWDLenght	540
8.575.2.3 bUnameLength	540
8.575.2.4 pAPN	540
8.575.2.5 pPWD	540
8.575.2.6 pUname	540
8.576PackSwiAVMSSettingsConnectionRetryTimers Struct Reference	540
8.576.1 Detailed Description	540
8.576.2 Field Documentation	540
8.576.2.1 Timers	540
8.577PackSwiAVMSSettingsPeriodsInfo Struct Reference	541
8.577.1 Detailed Description	541
8.577.2 Field Documentation	541
8.577.2.1 max	541
8.577.2.2 min	541
8.578qmiSmsMessageList Struct Reference	541
8.578.1 Detailed Description	541
8.578.2 Field Documentation	541
8.578.2.1 messageIndex	541
8.578.2.2 messageTag	541
8.579qmiWSDDataBearerTechnology Struct Reference	541
8.579.1 Detailed Description	542
8.579.2 Field Documentation	542

8.579.2.1 currentNetwork	542
8.579.2.2 ratMask	542
8.579.2.3 soMask	542
8.580qmTlvResult Struct Reference	542
8.580.1 Field Documentation	542
8.580.1.1 DeviceError	542
8.580.1.2 DeviceResult	542
8.580.1.3 TlvPresenceMask	542
8.580.1.4 TlvResultCode	542
8.581qos_BindDataPortMuxID_t Struct Reference	542
8.581.1 Detailed Description	543
8.581.2 Field Documentation	543
8.581.2.1 MuxID	543
8.582qos_BindDataPortPeripheralEndPointID_t Struct Reference	543
8.582.1 Detailed Description	543
8.582.2 Field Documentation	543
8.582.2.1 EndPointType	543
8.582.2.2 IfaceID	543
8.583qos_BindDataPortSIODDataPort_t Struct Reference	543
8.583.1 Detailed Description	544
8.583.2 Field Documentation	544
8.583.2.1 SIODDataPort	544
8.584RFBandInfoElements Struct Reference	544
8.584.1 Detailed Description	544
8.584.2 Field Documentation	544
8.584.2.1 activeBandClass	544
8.584.2.2 activeChannel	544
8.584.2.3 radiolInterface	544
8.585rmTrasferStaticsReq Struct Reference	544
8.585.1 Detailed Description	545
8.585.2 Field Documentation	545
8.585.2.1 bResetStatistics	545
8.585.2.2 ulMask	545
8.586sensorData_t Struct Reference	545
8.586.1 Detailed Description	545
8.586.2 Field Documentation	546
8.586.2.1 flags	546
8.586.2.2 sensorDataLen	546
8.586.2.3 timeOfFirstSample	546
8.586.2.4 timeOffset	546

8.586.2.5 xAxis	546
8.586.2.6 yAxis	546
8.586.2.7 zAxis	546
8.587slot_t Struct Reference	547
8.587.1 Detailed Description	547
8.587.2 Field Documentation	547
8.587.2.1 bICCID	547
8.587.2.2 bICCIDLength	547
8.587.2.3 bLogicalSlot	547
8.587.2.4 uPhyCardStatus	547
8.587.2.5 uPhySlotStatus	548
8.588slotInf Struct Reference	548
8.588.1 Detailed Description	548
8.588.2 Field Documentation	549
8.588.2.1 AppStatus	549
8.588.2.2 cardState	549
8.588.2.3 errorState	549
8.588.2.4 numApp	549
8.588.2.5 upinRetries	549
8.588.2.6 upinState	549
8.588.2.7 upukRetries	549
8.589slots_t Struct Reference	549
8.589.1 Detailed Description	549
8.589.2 Field Documentation	550
8.589.2.1 uimSlotStatus	550
8.590sms_BroadcastConfig Struct Reference	550
8.590.1 Detailed Description	550
8.590.2 Field Documentation	550
8.590.2.1 fromServiceId	550
8.590.2.2 selected	550
8.590.2.3 toServiceId	550
8.591sms_CDMABroadcastConfig Struct Reference	550
8.591.1 Detailed Description	551
8.591.2 Field Documentation	551
8.591.2.1 language	551
8.591.2.2 selected	551
8.591.2.3 serviceCategory	551
8.592sms_getIndicationReg Struct Reference	551
8.592.1 Detailed Description	551
8.592.2 Field Documentation	552

8.592.2.1 pRegCallStatInfoEvt	552
8.592.2.2 pRegTransLayerInfoEvt	552
8.592.2.3 pRegTransNWRegInfoEvt	552
8.593sms_getMsgWaitingInfo Struct Reference	552
8.593.1 Detailed Description	552
8.593.2 Field Documentation	552
8.593.2.1 msgWaitInfo	552
8.593.2.2 numInstances	552
8.594sms_getTransLayerInfo Struct Reference	552
8.594.1 Detailed Description	553
8.594.2 Field Documentation	553
8.594.2.1 pRegInd	553
8.594.2.2 pTransLayerInfo	553
8.595sms_getTransNWRegInfo Struct Reference	553
8.595.1 Detailed Description	553
8.595.2 Field Documentation	553
8.595.2.1 pRegStatus	554
8.596sms_maxStorageSizeReq Struct Reference	554
8.596.1 Detailed Description	554
8.596.2 Field Documentation	554
8.596.2.1 pMessageMode	554
8.596.2.2 storageType	554
8.597sms_maxStorageSizeResp Struct Reference	554
8.597.1 Detailed Description	554
8.597.2 Field Documentation	555
8.597.2.1 freeSlots	555
8.597.2.2 maxStorageSize	555
8.598sms_messageWaitingInfoContent Struct Reference	555
8.598.1 Detailed Description	555
8.598.2 Field Documentation	555
8.598.2.1 activeInd	555
8.598.2.2 msgCount	555
8.598.2.3 msgType	555
8.599sms_msgProtocolResp Struct Reference	556
8.599.1 Detailed Description	556
8.599.2 Field Documentation	556
8.599.2.1 msgProtocol	556
8.600sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference	556
8.600.1 Detailed Description	556
8.600.2 Field Documentation	557

8.600.2.1 activated_ind	557
8.600.2.2 CDMABroadcastConfig	557
8.600.2.3 num_instances	557
8.601 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference	557
8.601.1 Detailed Description	557
8.601.2 Field Documentation	557
8.601.2.1 activated_ind	557
8.601.2.2 broadcastConfig	557
8.601.2.3 num_instances	557
8.602 sms_routeEntry Struct Reference	557
8.602.1 Detailed Description	558
8.602.2 Field Documentation	558
8.602.2.1 messageClass	558
8.602.2.2 messageType	558
8.602.2.3 receiptAction	558
8.602.2.4 routeStorage	558
8.603 sms_sendAsyncsmsParams Struct Reference	558
8.603.1 Detailed Description	559
8.603.2 Field Documentation	560
8.603.2.1 messageFormat	560
8.603.2.2 messageSize	560
8.603.2.3 pFollowOnDC	560
8.603.2.4 pForceOnDC	560
8.603.2.5 pLinktimer	560
8.603.2.6 pMessage	560
8.603.2.7 pRetryMessage	560
8.603.2.8 pRetryMessageId	560
8.603.2.9 pServiceOption	560
8.603.2.10 pSmsOnIms	560
8.603.2.11 pUserData	560
8.604 sms_setIndicationReg Struct Reference	561
8.604.1 Detailed Description	561
8.604.2 Field Documentation	561
8.604.2.1 pRegCallStatInfoEvt	561
8.604.2.2 pRegTransLayerInfoEvt	561
8.604.2.3 pRegTransNWRegInfoEvt	561
8.605 sms_setRoutesReq Struct Reference	561
8.605.1 Detailed Description	562
8.605.2 Field Documentation	562
8.605.2.1 numOfRoutes	562

8.605.2.2 pTransferStatusReport	562
8.605.2.3 routeList	562
8.606sms_transLayerInfo Struct Reference	562
8.606.1 Detailed Description	562
8.606.2 Field Documentation	562
8.606.2.1 TransCap	563
8.606.2.2 TransType	563
8.607sMSCAddressInfo Struct Reference	563
8.607.1 Detailed Description	563
8.607.2 Field Documentation	563
8.607.2.1 data	563
8.607.2.2 length	563
8.608sMSCAddressTlv Struct Reference	563
8.608.1 Detailed Description	563
8.608.2 Field Documentation	563
8.608.2.1 SMSCInfo	564
8.608.2.2 TlvPresent	564
8.609sMSEtwsMessageInfo Struct Reference	564
8.609.1 Detailed Description	564
8.609.2 Field Documentation	564
8.609.2.1 data	564
8.609.2.2 length	564
8.609.2.3 notificationType	564
8.610sMSEtwsMessageTlv Struct Reference	564
8.610.1 Detailed Description	564
8.610.2 Field Documentation	565
8.610.2.1 EtwsMessageInfo	565
8.610.2.2 TlvPresent	565
8.611sMSEtwsPlmnInfo Struct Reference	565
8.611.1 Detailed Description	565
8.611.2 Field Documentation	565
8.611.2.1 mobileCountryCode	565
8.611.2.2 mobileNetworkCode	565
8.612sMSMessageModeInfo Struct Reference	565
8.612.1 Detailed Description	565
8.612.2 Field Documentation	566
8.612.2.1 messageMode	566
8.613sMSMTMessageInfo Struct Reference	566
8.613.1 Detailed Description	566
8.613.2 Field Documentation	566

8.613.2.1 messageIndex	566
8.613.2.2 storageType	566
8.614sMSOnIMSInfo Struct Reference	566
8.614.1 Detailed Description	566
8.614.2 Field Documentation	566
8.614.2.1 smsOnIMS	566
8.615sMSOnIMSTlv Struct Reference	567
8.615.1 Detailed Description	567
8.615.2 Field Documentation	567
8.615.2.1 IMSInfo	567
8.615.2.2 TlvPresent	567
8.616sMSTransferRouteMTMessageInfo Struct Reference	567
8.616.1 Detailed Description	567
8.616.2 Field Documentation	568
8.616.2.1 ackIndicator	568
8.616.2.2 data	568
8.616.2.3 format	568
8.616.2.4 length	568
8.616.2.5 transactionID	568
8.617swi_uint256_t Struct Reference	568
8.617.1 Detailed Description	568
8.617.2 Field Documentation	568
8.617.2.1 word	568
8.618swiaudio_PCMparams Struct Reference	568
8.618.1 Detailed Description	568
8.618.2 Field Documentation	569
8.618.2.1 iFaceTab	569
8.618.2.2 iFaceTabLen	569
8.619swidms_ehrpdMTUSizeTlv Struct Reference	569
8.619.1 Detailed Description	569
8.619.2 Field Documentation	569
8.619.2.1 ehrpdMTUSize	569
8.619.2.2 TlvPresent	569
8.620swidms_hrpdmTUSizeTlv Struct Reference	569
8.620.1 Detailed Description	569
8.620.2 Field Documentation	570
8.620.2.1 hrpdMTUSize	570
8.620.2.2 TlvPresent	570
8.621swidms_intfaceCfgTlv Struct Reference	570
8.621.1 Detailed Description	570

8.621.2 Field Documentation	571
8.621.2.1 CfgValue	571
8.621.2.2 CurrentCfgType	571
8.621.2.3 TlvPresent	571
8.622swidms_mtuSize3gppTlv Struct Reference	571
8.622.1 Detailed Description	571
8.622.2 Field Documentation	572
8.622.2.1 MTUSize3gpp	572
8.622.2.2 TlvPresent	572
8.623swidms_supportedIntBitmaskTlv Struct Reference	572
8.623.1 Detailed Description	572
8.623.2 Field Documentation	572
8.623.2.1 TlvPresent	572
8.623.2.2 ValidBitmasks	572
8.624swidms_SwiDmsGetHWWatchdog Struct Reference	572
8.624.1 Detailed Description	572
8.624.2 Field Documentation	573
8.624.2.1 count	573
8.624.2.2 enable	573
8.624.2.3 resetDelay	573
8.624.2.4 timeout	573
8.625swidms_usbMTUSizeTlv Struct Reference	573
8.625.1 Detailed Description	573
8.625.2 Field Documentation	573
8.625.2.1 TlvPresent	573
8.625.2.2 UsbMTUSize	573
8.626tdscdmaSigInfoExt Struct Reference	574
8.626.1 Detailed Description	574
8.626.2 Field Documentation	574
8.626.2.1 ecio	574
8.626.2.2 rscp	574
8.626.2.3 rssi	574
8.626.2.4 sinr	574
8.627tempData_t Struct Reference	574
8.627.1 Detailed Description	574
8.627.2 Field Documentation	575
8.627.2.1 temperature	575
8.627.2.2 temperatureDataLen	575
8.627.2.3 timeOfFirstSample	575
8.627.2.4 timeOffset	575

8.627.2.5 timeSource	575
8.628tmd_mitigationDevList Struct Reference	575
8.628.1 Detailed Description	575
8.628.2 Field Documentation	576
8.628.2.1 maxMitigationLevel	576
8.628.2.2 mitigationDevId	576
8.628.2.3 mitigationDevIdLen	576
8.629transferRouteMessageTlv Struct Reference	576
8.629.1 Detailed Description	576
8.629.2 Field Documentation	576
8.629.2.1 TlvPresent	576
8.629.2.2 TransferRouteMTMessageInfo	576
8.630uim_appStatus Struct Reference	576
8.630.1 Detailed Description	577
8.630.2 Field Documentation	579
8.630.2.1 aidLength	579
8.630.2.2 aidVal	579
8.630.2.3 appState	579
8.630.2.4 appType	579
8.630.2.5 persoFeature	579
8.630.2.6 persoRetries	579
8.630.2.7 persoState	579
8.630.2.8 persoUnblockRetries	579
8.630.2.9 pin1Retries	579
8.630.2.10pin1State	579
8.630.2.11pin2Retries	579
8.630.2.12pin2State	579
8.630.2.13puk1Retries	579
8.630.2.14puk2Retries	579
8.630.2.15univPin	579
8.631uim_authenticateResult Struct Reference	579
8.631.1 Detailed Description	580
8.631.2 Field Documentation	580
8.631.2.1 content	580
8.631.2.2 contentLen	580
8.632uim_authenticationData Struct Reference	580
8.632.1 Detailed Description	580
8.632.2 Field Documentation	581
8.632.2.1 context	581
8.632.2.2 data	581

8.632.2.3 dataLen	581
8.633uim_cardResult Struct Reference	581
8.633.1 Detailed Description	581
8.633.2 Field Documentation	582
8.633.2.1 sw1	582
8.633.2.2 sw2	582
8.634uim_cardStatus Struct Reference	582
8.634.1 Detailed Description	582
8.634.2 Field Documentation	583
8.634.2.1 index1xPri	583
8.634.2.2 index1xSec	583
8.634.2.3 indexGwPri	583
8.634.2.4 indexGwSec	583
8.634.2.5 numSlot	583
8.634.2.6 SlotInfo	583
8.635uim_changeUIMPIN Struct Reference	583
8.635.1 Detailed Description	583
8.635.2 Field Documentation	584
8.635.2.1 oldPINLen	584
8.635.2.2 oldPINVal	584
8.635.2.3 pinID	584
8.635.2.4 pinLen	584
8.635.2.5 pinVal	584
8.636uim_depersonalizationInformation Struct Reference	584
8.636.1 Detailed Description	584
8.636.2 Field Documentation	585
8.636.2.1 ckLen	585
8.636.2.2 ckVal	585
8.636.2.3 feature	585
8.636.2.4 operation	585
8.637uim_encryptedPIN1 Struct Reference	585
8.637.1 Detailed Description	585
8.637.2 Field Documentation	586
8.637.2.1 pin1Len	586
8.637.2.2 pin1Val	586
8.638uim_fileAttributes Struct Reference	586
8.638.1 Detailed Description	586
8.638.2 Field Documentation	589
8.638.2.1 fileID	589
8.638.2.2 fileSize	589

8.638.2.3 fileType	589
8.638.2.4 rawLen	589
8.638.2.5 rawValue	589
8.638.2.6 recordCount	589
8.638.2.7 recordSize	589
8.638.2.8 secActivate	589
8.638.2.9 secActivateMask	589
8.638.2.10secDeactivate	589
8.638.2.11secDeactivateMask	589
8.638.2.12secIncrease	589
8.638.2.13secIncreaseMask	589
8.638.2.14secRead	589
8.638.2.15secReadMask	589
8.638.2.16secWrite	589
8.638.2.17secWriteMask	589
8.639uim_fileInfo Struct Reference	589
8.639.1 Detailed Description	589
8.639.2 Field Documentation	590
8.639.2.1 fileID	590
8.639.2.2 path	590
8.639.2.3 pathLen	590
8.640uim_GetSlotsInfoTlv Struct Reference	590
8.640.1 Detailed Description	590
8.640.2 Field Documentation	590
8.640.2.1 NumberOfPhySlotInfo	590
8.640.2.2 TlvPresent	590
8.640.2.3 uimSlotInfo	590
8.641uim_GetSlotsStatusTlv Struct Reference	591
8.641.1 Detailed Description	591
8.641.2 Field Documentation	591
8.641.2.1 NumberOfPhySlot	591
8.641.2.2 TlvPresent	591
8.641.2.3 uimSlotStatus	591
8.642uim_hotSwapStatus Struct Reference	591
8.642.1 Detailed Description	591
8.642.2 Field Documentation	592
8.642.2.1 hotSwap	592
8.642.2.2 hotSwapLength	592
8.643uim_personalizationStatus Struct Reference	592
8.643.1 Detailed Description	592

8.643.2 Field Documentation	593
8.643.2.1 feature	593
8.643.2.2 numFeatures	593
8.643.2.3 unblockLeft	593
8.643.2.4 verifyLeft	593
8.644uim_physlotInfo Struct Reference	593
8.644.1 Detailed Description	593
8.644.2 Field Documentation	594
8.644.2.1 atrValue	594
8.644.2.2 atrValueLen	594
8.644.2.3 cardProtocol	594
8.644.2.4 iseUICC	594
8.644.2.5 numApp	594
8.645uim_physlotsInfo Struct Reference	594
8.645.1 Detailed Description	594
8.645.2 Field Documentation	594
8.645.2.1 uimSlotInfo	594
8.646uim_readResult Struct Reference	594
8.646.1 Detailed Description	594
8.646.2 Field Documentation	595
8.646.2.1 content	595
8.646.2.2 contentLen	595
8.647uim_readTransparentInfo Struct Reference	595
8.647.1 Detailed Description	595
8.647.2 Field Documentation	595
8.647.2.1 length	595
8.647.2.2 offset	595
8.648uim_refreshevent Struct Reference	595
8.648.1 Detailed Description	595
8.648.2 Field Documentation	596
8.648.2.1 aid	597
8.648.2.2 aidLength	597
8.648.2.3 arrfileInfo	597
8.648.2.4 mode	597
8.648.2.5 numOfFiles	597
8.648.2.6 sessionType	597
8.648.2.7 stage	597
8.649uim_registerRefresh Struct Reference	597
8.649.1 Detailed Description	597
8.649.2 Field Documentation	597

8.649.2.1 arrfileInfo	598
8.649.2.2 numFiles	598
8.649.2.3 registerFlag	598
8.649.2.4 voteForInit	598
8.650uim_remainingRetries Struct Reference	598
8.650.1 Detailed Description	598
8.650.2 Field Documentation	598
8.650.2.1 unblockLeft	598
8.650.2.2 verifyLeft	598
8.651uim_sessionInformation Struct Reference	598
8.651.1 Detailed Description	598
8.651.2 Field Documentation	599
8.651.2.1 aid	599
8.651.2.2 aidLength	599
8.651.2.3 sessionType	599
8.652uim_setPINProtection Struct Reference	599
8.652.1 Detailed Description	599
8.652.2 Field Documentation	600
8.652.2.1 pinID	600
8.652.2.2 pinLength	600
8.652.2.3 pinOperation	600
8.652.2.4 pinValue	600
8.653uim_simBusyStatus Struct Reference	600
8.653.1 Detailed Description	600
8.653.2 Field Documentation	600
8.653.2.1 simBusy	600
8.653.2.2 simBusyLength	600
8.654uim_slotInfo Struct Reference	601
8.654.1 Detailed Description	601
8.654.2 Field Documentation	602
8.654.2.1 AppStatus	602
8.654.2.2 cardState	602
8.654.2.3 errorState	602
8.654.2.4 numApp	602
8.654.2.5 upinRetries	602
8.654.2.6 upinState	602
8.654.2.7 upukRetries	602
8.655uim_UIMSessionInformation Struct Reference	602
8.655.1 Detailed Description	602
8.655.2 Field Documentation	603

8.655.2.1 aid	603
8.655.2.2 aidLength	603
8.655.2.3 sessionType	603
8.656uim_unblockUIMPIN Struct Reference	603
8.656.1 Detailed Description	603
8.656.2 Field Documentation	604
8.656.2.1 newPINLen	604
8.656.2.2 newPINVal	604
8.656.2.3 pinID	604
8.656.2.4 pukLen	604
8.656.2.5 pukVal	604
8.657uim_validCardStatus Struct Reference	604
8.657.1 Detailed Description	604
8.657.2 Field Documentation	604
8.657.2.1 validCard	605
8.657.2.2 validCardLength	605
8.658uim_verifyUIMPIN Struct Reference	605
8.658.1 Detailed Description	605
8.658.2 Field Documentation	605
8.658.2.1 pinID	605
8.658.2.2 pinLen	605
8.658.2.3 pinVal	605
8.659unpack_audio_SLQSGetAudioPathConfig_t Struct Reference	605
8.659.1 Detailed Description	606
8.659.2 Field Documentation	607
8.659.2.1 ParamPresenceMask	607
8.659.2.2 pCodecSTGain	607
8.659.2.3 pDTMFTXGain	607
8.659.2.4 pECMode	607
8.659.2.5 pMICGainSelect	607
8.659.2.6 pNSEnable	607
8.659.2.7 pRXAGCList	607
8.659.2.8 pRXAVCAGCSwitch	607
8.659.2.9 pRXAVCList	607
8.659.2.10pRXPCMIIRFitr	607
8.659.2.11pTXAGCList	607
8.659.2.12pTXAVCSwitch	607
8.659.2.13pTXGain	607
8.659.2.14pTXPCMIIRFitr	608
8.660unpack_audio_SLQSGetAudioProfile_t Struct Reference	608

8.660.1 Detailed Description	608
8.660.2 Field Documentation	608
8.660.2.1 EarMute	609
8.660.2.2 MicMute	609
8.660.2.3 ParamPresenceMask	609
8.660.2.4 Profile	609
8.660.2.5 Volume	609
8.661unpack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference	609
8.661.1 Detailed Description	609
8.661.2 Field Documentation	609
8.661.2.1 ParamPresenceMask	609
8.661.2.2 ResCode	609
8.662unpack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference	609
8.662.1 Detailed Description	609
8.662.2 Field Documentation	610
8.662.2.1 ParamPresenceMask	610
8.662.2.2 ResCode	610
8.663unpack_cat_SetCatEventCallback_ind_t Struct Reference	610
8.663.1 Detailed Description	610
8.663.2 Field Documentation	610
8.663.2.1 CCETlv	610
8.663.2.2 event_Index	610
8.663.2.3 ParamPresenceMask	610
8.664unpack_cat_SetCATEventCallback_t Struct Reference	610
8.664.1 Detailed Description	611
8.664.2 Field Documentation	611
8.664.2.1 errorMask	611
8.664.2.2 ParamPresenceMask	611
8.664.2.3 Tlvresult	611
8.665unpack_dms_GetActivationState_t Struct Reference	611
8.665.1 Detailed Description	611
8.665.2 Field Documentation	612
8.665.2.1 ParamPresenceMask	612
8.665.2.2 state	612
8.666unpack_dms_GetBandCapability_t Struct Reference	612
8.666.1 Detailed Description	612
8.666.2 Field Documentation	614
8.666.2.1 BandCapability	614
8.666.2.2 ParamPresenceMask	614
8.666.2.3 Tlvresult	614

8.667unpack_dms_GetCrashAction_t Struct Reference	614
8.667.1 Detailed Description	614
8.667.2 Field Documentation	614
8.667.2.1 DevCrashState	614
8.667.2.2 ParamPresenceMask	615
8.667.2.3 Tlvresult	615
8.668unpack_dms_GetCustFeature_t Struct Reference	615
8.668.1 Detailed Description	615
8.668.2 Field Documentation	617
8.668.2.1 DHCPRelayEnabled	617
8.668.2.2 DisableIMSI	617
8.668.2.3 GpsEnable	617
8.668.2.4 GPSLPM	617
8.668.2.5 GPSSel	617
8.668.2.6 IPFamSupport	617
8.668.2.7 IsVoiceEnabled	617
8.668.2.8 ParamPresenceMask	617
8.668.2.9 RMAutoConnect	617
8.668.2.10SMSSupport	617
8.668.2.11Tlvresult	617
8.669unpack_dms_GetCustFeaturesV2_t Struct Reference	617
8.669.1 Detailed Description	617
8.669.2 Field Documentation	618
8.669.2.1 GetCustomFeatureV2	618
8.669.2.2 ParamPresenceMask	618
8.669.2.3 Tlvresult	618
8.670unpack_dms_GetDeviceCap_t Struct Reference	618
8.670.1 Detailed Description	618
8.670.2 Field Documentation	619
8.670.2.1 DataServiceCapability	619
8.670.2.2 MaxRXChannelRate	619
8.670.2.3 MaxTXChannelRate	619
8.670.2.4 ParamPresenceMask	619
8.670.2.5 Radiofaces	619
8.670.2.6 RadiofacesSize	619
8.670.2.7 SimCapability	619
8.670.2.8 Tlvresult	619
8.671unpack_dms_GetDeviceCapabilities_t Struct Reference	619
8.671.1 Detailed Description	620
8.671.2 Field Documentation	620

8.671.2.1 dataServiceCaCapability	621
8.671.2.2 maxRxChannelRate	621
8.671.2.3 maxTxChannelRate	621
8.671.2.4 ParamPresenceMask	621
8.671.2.5 Radiolfaces	621
8.671.2.6 radiolfacesSize	621
8.671.2.7 simCapability	621
8.672unpack_dms_GetDeviceCapabilitiesV2_t Struct Reference	621
8.672.1 Detailed Description	621
8.672.2 Field Documentation	623
8.672.2.1 DevCaps	623
8.672.2.2 ParamPresenceMask	623
8.672.2.3 pDevCurSubsCaps	623
8.672.2.4 pDevExplicitCfgIndex	623
8.672.2.5 pDevMaxActDataSubsCaps	623
8.672.2.6 pDevMaxCfgListCaps	623
8.672.2.7 pDevMaxSubsCaps	623
8.672.2.8 pDevMultiSimCaps	623
8.672.2.9 pDevMultiSimVoiceDataCaps	623
8.672.2.10pDevSrvCaps	623
8.672.2.11pDevSubsFeatureModeCaps	623
8.672.2.12pDevSubsVoiceDataCaps	623
8.672.2.13pDevVoiceCaps	623
8.672.2.14pDevVoiceDataCaps	623
8.672.2.15Tlvresult	623
8.673unpack_dms_GetDeviceHardwareRev_t Struct Reference	623
8.673.1 Detailed Description	624
8.673.2 Field Documentation	624
8.673.2.1 ParamPresenceMask	624
8.673.2.2 String	624
8.673.2.3 stringSize	624
8.673.2.4 Tlvresult	624
8.674unpack_dms_GetDeviceMfr_t Struct Reference	624
8.674.1 Detailed Description	624
8.674.2 Field Documentation	625
8.674.2.1 ParamPresenceMask	625
8.674.2.2 String	625
8.674.2.3 stringSize	625
8.674.2.4 Tlvresult	625
8.675unpack_dms_GetDeviceSerialNumbers_t Struct Reference	625

8.675.1 Detailed Description	625
8.675.2 Field Documentation	626
8.675.2.1 esnSize	627
8.675.2.2 ESNString	627
8.675.2.3 imeiSize	627
8.675.2.4 IMEIString	627
8.675.2.5 imeiSvnSize	627
8.675.2.6 ImeiSvnString	627
8.675.2.7 meidSize	627
8.675.2.8 MEIDString	627
8.675.2.9 ParamPresenceMask	627
8.675.2.10Tlvresult	627
8.676unpack_dms_GetFirmwareInfo_t Struct Reference	627
8.676.1 Detailed Description	627
8.676.2 Field Documentation	628
8.676.2.1 appversion_str	628
8.676.2.2 bootversion_str	628
8.676.2.3 carrier_str	628
8.676.2.4 cur_carr_name	628
8.676.2.5 cur_carr_rev	628
8.676.2.6 modelid_str	628
8.676.2.7 packageid_str	628
8.676.2.8 ParamPresenceMask	628
8.676.2.9 priversion_str	628
8.676.2.10sku_str	628
8.676.2.11Tlvresult	629
8.677unpack_dms_GetFirmwareRevision_t Struct Reference	629
8.677.1 Detailed Description	629
8.677.2 Field Documentation	629
8.677.2.1 amssSize	629
8.677.2.2 AMSSString	629
8.677.2.3 ParamPresenceMask	629
8.677.2.4 PRIString	629
8.677.2.5 Tlvresult	629
8.678unpack_dms_GetFirmwareRevisions_t Struct Reference	630
8.678.1 Detailed Description	630
8.678.2 Field Documentation	631
8.678.2.1 amssSize	631
8.678.2.2 AMSSString	631
8.678.2.3 bootSize	631

8.678.2.4 BootString	631
8.678.2.5 ParamPresenceMask	631
8.678.2.6 priSize	631
8.678.2.7 PRIString	631
8.678.2.8 Tlvresult	631
8.679unpack_dms_GetFSN_t Struct Reference	631
8.679.1 Detailed Description	631
8.679.2 Field Documentation	631
8.679.2.1 ParamPresenceMask	631
8.679.2.2 String	631
8.679.2.3 Tlvresult	631
8.680unpack_dms_GetHardwareRevision_t Struct Reference	632
8.680.1 Detailed Description	632
8.680.2 Field Documentation	632
8.680.2.1 hwVer	632
8.680.2.2 ParamPresenceMask	632
8.681unpack_dms_GetIMSI_t Struct Reference	632
8.681.1 Detailed Description	632
8.681.2 Field Documentation	632
8.681.2.1 imsi	633
8.681.2.2 ParamPresenceMask	633
8.681.2.3 Tlvresult	633
8.682unpack_dms_GetManufacturer_t Struct Reference	633
8.682.1 Detailed Description	633
8.682.2 Field Documentation	633
8.682.2.1 manufacturer	633
8.682.2.2 ParamPresenceMask	633
8.682.2.3 Tlvresult	633
8.683unpack_dms_GetModelID_t Struct Reference	633
8.683.1 Detailed Description	633
8.683.2 Field Documentation	634
8.683.2.1 modelid	634
8.683.2.2 ParamPresenceMask	634
8.683.2.3 Tlvresult	634
8.684unpack_dms_GetNetworkTime_t Struct Reference	634
8.684.1 Detailed Description	634
8.684.2 Field Documentation	635
8.684.2.1 ParamPresenceMask	635
8.684.2.2 source	635
8.684.2.3 timestamp	635

8.684.2.4 Tlvresult	635
8.685unpack_dms_GetNetworkTimeV2_t Struct Reference	635
8.685.1 Detailed Description	635
8.685.2 Field Documentation	636
8.685.2.1 ParamPresenceMask	636
8.685.2.2 pSysTime	636
8.685.2.3 pUsrTime	636
8.685.2.4 source	636
8.685.2.5 timestamp	636
8.685.2.6 Tlvresult	636
8.686unpack_dms_GetOfflineReason_t Struct Reference	636
8.686.1 Detailed Description	636
8.686.2 Field Documentation	637
8.686.2.1 ParamPresenceMask	637
8.686.2.2 pbPlatform	637
8.686.2.3 pReasonMask	637
8.686.2.4 Tlvresult	637
8.687unpack_dms_GetPower_t Struct Reference	637
8.687.1 Detailed Description	637
8.687.2 Field Documentation	638
8.687.2.1 HardwareControlledMode	638
8.687.2.2 OfflineReason	638
8.687.2.3 OperationMode	638
8.687.2.4 ParamPresenceMask	638
8.687.2.5 Tlvresult	638
8.688unpack_dms_GetPRLVersion_t Struct Reference	638
8.688.1 Detailed Description	638
8.688.2 Field Documentation	639
8.688.2.1 ParamPresenceMask	639
8.688.2.2 Tlvresult	639
8.688.2.3 u16PRLVersion	639
8.688.2.4 u8PRLPreference	639
8.689unpack_dms_GetSerialNumbers_t Struct Reference	639
8.689.1 Detailed Description	639
8.689.2 Field Documentation	640
8.689.2.1 esn	640
8.689.2.2 imei_no	640
8.689.2.3 imeisv_svn	640
8.689.2.4 meid	640
8.689.2.5 ParamPresenceMask	640

8.690unpack_dms_GetUSBComp_t Struct Reference	640
8.690.1 Detailed Description	640
8.690.2 Field Documentation	642
8.690.2.1 NumSupUSBComps	642
8.690.2.2 ParamPresenceMask	642
8.690.2.3 SupUSBComps	642
8.690.2.4 Tlvresult	642
8.690.2.5 USBComp	642
8.691unpack_dms_GetVoiceNumber_t Struct Reference	642
8.691.1 Detailed Description	643
8.691.2 Field Documentation	643
8.691.2.1 MIN	643
8.691.2.2 minSize	643
8.691.2.3 ParamPresenceMask	643
8.691.2.4 Tlvresult	643
8.691.2.5 VoiceNumber	643
8.691.2.6 voiceNumberSize	643
8.692unpack_dms_PSMCfgChange_ind_t Struct Reference	644
8.692.1 Detailed Description	644
8.692.2 Field Documentation	644
8.692.2.1 ActiveTimer	644
8.692.2.2 EnableState	644
8.692.2.3 ParamPresenceMask	644
8.692.2.4 PeriodicUpdateTimer	644
8.692.2.5 Tlvresult	644
8.693unpack_dms_ResetToFactoryDefaults_t Struct Reference	644
8.693.1 Detailed Description	645
8.693.2 Field Documentation	645
8.693.2.1 ParamPresenceMask	645
8.693.2.2 Tlvresult	645
8.694unpack_dms_SetActivationStatusCallback_t Struct Reference	645
8.694.1 Detailed Description	645
8.694.2 Field Documentation	645
8.694.2.1 ParamPresenceMask	645
8.694.2.2 Tlvresult	645
8.695unpack_dms_SetCrashAction_t Struct Reference	645
8.695.1 Detailed Description	646
8.695.2 Field Documentation	646
8.695.2.1 notused	646
8.695.2.2 ParamPresenceMask	646

8.696unpack_dms_SetCustFeature_t Struct Reference	646
8.696.1 Detailed Description	646
8.696.2 Field Documentation	646
8.696.2.1 ParamPresenceMask	646
8.696.2.2 Tlvresult	646
8.697unpack_dms_SetCustFeaturesV2_t Struct Reference	646
8.697.1 Detailed Description	647
8.697.2 Field Documentation	647
8.697.2.1 ParamPresenceMask	647
8.697.2.2 Tlvresult	647
8.698unpack_dms_SetEventReport_ind_t Struct Reference	647
8.698.1 Detailed Description	647
8.698.2 Field Documentation	647
8.698.2.1 ActivationStatusTlv	647
8.698.2.2 OperatingModeTlv	648
8.698.2.3 ParamPresenceMask	648
8.698.2.4 Tlvresult	648
8.699unpack_dms_SetEventReport_t Struct Reference	648
8.699.1 Detailed Description	648
8.699.2 Field Documentation	648
8.699.2.1 ParamPresenceMask	648
8.699.2.2 Tlvresult	648
8.700unpack_dms_SetFirmwarePreference_t Struct Reference	648
8.700.1 Detailed Description	648
8.700.2 Field Documentation	649
8.700.2.1 ParamPresenceMask	649
8.700.2.2 Tlvresult	649
8.701unpack_dms_SetIndicationRegister_t Struct Reference	649
8.701.1 Detailed Description	649
8.701.2 Field Documentation	649
8.701.2.1 ParamPresenceMask	649
8.701.2.2 Tlvresult	649
8.702unpack_dms_SetPower_t Struct Reference	649
8.702.1 Detailed Description	649
8.702.2 Field Documentation	650
8.702.2.1 ParamPresenceMask	650
8.702.2.2 Tlvresult	650
8.703unpack_dms_SetUSBComp_t Struct Reference	650
8.703.1 Detailed Description	650
8.703.2 Field Documentation	650

8.703.2.1 ParamPresenceMask	650
8.703.2.2 Tlvresult	650
8.704unpack_dms_SLQSDmsSwiGetPCInfo_t Struct Reference	650
8.704.1 Detailed Description	650
8.704.2 Field Documentation	652
8.704.2.1 has_LpmFlag	652
8.704.2.2 has_PersistentLpm	652
8.704.2.3 has_PowerOffMode	652
8.704.2.4 has_Wdisable	653
8.704.2.5 LpmFlag	653
8.704.2.6 opMode	653
8.704.2.7 ParamPresenceMask	653
8.704.2.8 PersistentLpm	653
8.704.2.9 PowerOffMode	653
8.704.2.10Wdisable	653
8.705unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference	653
8.705.1 Detailed Description	653
8.705.2 Field Documentation	654
8.705.2.1 ParamPresenceMask	654
8.705.2.2 source	654
8.705.2.3 Tlvresult	654
8.705.2.4 type	654
8.706unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference	654
8.706.1 Detailed Description	654
8.706.2 Field Documentation	655
8.706.2.1 ParamPresenceMask	655
8.706.2.2 source	655
8.706.2.3 Tlvresult	655
8.706.2.4 type	655
8.707unpack_dms_SLQSDmsSwiGetUimSelection_t Struct Reference	655
8.707.1 Detailed Description	655
8.707.2 Field Documentation	656
8.707.2.1 ParamPresenceMask	656
8.707.2.2 pUimAutoSwitchActSlot	656
8.707.2.3 uimSelect	656
8.708unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	656
8.708.1 Detailed Description	656
8.708.2 Field Documentation	657
8.708.2.1 ParamPresenceMask	657
8.708.2.2 Tlvresult	657

8.709unpack_dms_SLQSGetBandCapability_t Struct Reference	657
8.709.1 Detailed Description	657
8.709.2 Field Documentation	661
8.709.2.1 bandCapability	661
8.709.2.2 is_LteBandCapability_Available	661
8.709.2.3 is_TdsBandCapability_Available	661
8.709.2.4 LteBandCapability	661
8.709.2.5 ParamPresenceMask	661
8.709.2.6 TdsBandCapability	661
8.710unpack_dms_SLQSGetBandCapabilityExt_t Struct Reference	661
8.710.1 Detailed Description	661
8.710.2 Field Documentation	665
8.710.2.1 bandCapability	665
8.710.2.2 is_LteBandCapability_Available	665
8.710.2.3 is_TdsBandCapability_Available	665
8.710.2.4 LteBandCapability	665
8.710.2.5 LteBandsSupport	665
8.710.2.6 ParamPresenceMask	665
8.710.2.7 TdsBandCapability	665
8.711unpack_dms_SLQSGetERIFile_t Struct Reference	665
8.711.1 Detailed Description	665
8.711.2 Field Documentation	666
8.711.2.1 eriFile	666
8.711.2.2 ParamPresenceMask	666
8.711.2.3 Tlvresult	666
8.712unpack_dms_SLQSGetPowerSaveModeConfig_t Struct Reference	666
8.712.1 Detailed Description	666
8.712.2 Field Documentation	667
8.712.2.1 pActiveTimer	667
8.712.2.2 ParamPresenceMask	667
8.712.2.3 pDurationDueToOOS	667
8.712.2.4 pDurationThreshold	667
8.712.2.5 pEarlyWakeupTime	667
8.712.2.6 pPeriodicUpdateTimer	667
8.712.2.7 pPsmEnableState	667
8.712.2.8 pRandomizationWindow	667
8.713unpack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference	667
8.713.1 Detailed Description	667
8.713.2 Field Documentation	668
8.713.2.1 ParamPresenceMask	668

8.713.2.2 Tlvresult	668
8.714unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference	668
8.714.1 Detailed Description	668
8.714.2 Field Documentation	668
8.714.2.1 ParamPresenceMask	668
8.714.2.2 Tlvresult	668
8.715unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference	668
8.715.1 Detailed Description	668
8.715.2 Field Documentation	669
8.715.2.1 crashInfoParam	669
8.715.2.2 ParamPresenceMask	669
8.715.2.3 Tlvresult	669
8.716unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference	669
8.716.1 Detailed Description	669
8.716.2 Field Documentation	670
8.716.2.1 ParamPresenceMask	670
8.716.2.2 pGetDyingGaspCfg	670
8.716.2.3 Tlvresult	670
8.717unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference	670
8.717.1 Detailed Description	670
8.717.2 Field Documentation	670
8.717.2.1 ParamPresenceMask	670
8.717.2.2 pGetDyingGaspStatistics	670
8.717.2.3 Tlvresult	670
8.718unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference	670
8.718.1 Detailed Description	671
8.718.2 Field Documentation	671
8.718.2.1 carrier	671
8.718.2.2 fwvers	671
8.718.2.3 numEntries	671
8.718.2.4 ParamPresenceMask	671
8.718.2.5 pCurrImgInfo	671
8.718.2.6 pkgver	671
8.718.2.7 priver	671
8.719unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference	671
8.719.1 Detailed Description	672
8.719.2 Field Documentation	673
8.719.2.1 imgType	673
8.719.2.2 logString	673
8.719.2.3 ParamPresenceMask	673

8.719.2.4 refData	673
8.719.2.5 refString	673
8.719.2.6 ResCode	673
8.719.2.7 Tlvresult	673
8.720unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference	673
8.720.1 Detailed Description	673
8.720.2 Field Documentation	674
8.720.2.1 hostID	674
8.720.2.2 manString	674
8.720.2.3 modelString	674
8.720.2.4 ParamPresenceMask	674
8.720.2.5 plasmaIDString	674
8.720.2.6 swVerString	674
8.720.2.7 Tlvresult	674
8.721unpack_dms_SLQSSwiGetOSInfo_t Struct Reference	674
8.721.1 Detailed Description	675
8.721.2 Field Documentation	675
8.721.2.1 nameString	675
8.721.2.2 ParamPresenceMask	675
8.721.2.3 Tlvresult	675
8.721.2.4 versionString	675
8.722unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference	675
8.722.1 Detailed Description	675
8.722.2 Field Documentation	676
8.722.2.1 meidString	676
8.722.2.2 ParamPresenceMask	676
8.722.2.3 Tlvresult	676
8.723unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	676
8.723.1 Detailed Description	676
8.723.2 Field Documentation	676
8.723.2.1 ParamPresenceMask	676
8.723.2.2 Tlvresult	676
8.724unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference	676
8.724.1 Detailed Description	676
8.724.2 Field Documentation	677
8.724.2.1 ParamPresenceMask	677
8.724.2.2 Tlvresult	677
8.725unpack_dms_SLQSSwiSetOSInfo_t Struct Reference	677
8.725.1 Detailed Description	677
8.725.2 Field Documentation	677

8.725.2.1 ParamPresenceMask	677
8.725.2.2 Tlvresult	677
8.726unpack_dms_SLQSUIMGetState_t Struct Reference	677
8.726.1 Detailed Description	677
8.726.2 Field Documentation	678
8.726.2.1 ParamPresenceMask	678
8.726.2.2 state	678
8.726.2.3 Tlvresult	678
8.727unpack_dms_SwiEventReportCallBack_ind_t Struct Reference	678
8.727.1 Detailed Description	678
8.727.2 Field Documentation	679
8.727.2.1 ParamPresenceMask	679
8.727.2.2 TempTlv	679
8.727.2.3 UimStatusTlv	679
8.727.2.4 VoltTlv	679
8.728unpack_dms_SwiSetEventReport_t Struct Reference	679
8.728.1 Detailed Description	679
8.728.2 Field Documentation	679
8.728.2.1 ParamPresenceMask	679
8.728.2.2 Tlvresult	679
8.729unpack_dms_SwiUimSelect_t Struct Reference	679
8.729.1 Detailed Description	679
8.729.2 Field Documentation	680
8.729.2.1 ParamPresenceMask	680
8.729.2.2 Tlvresult	680
8.730unpack_dms_UIMGetControlKeyStatus_t Struct Reference	680
8.730.1 Detailed Description	680
8.730.2 Field Documentation	681
8.730.2.1 facilityState	681
8.730.2.2 ParamPresenceMask	681
8.730.2.3 Tlvresult	681
8.730.2.4 unblockRetriesLeft	681
8.730.2.5 verifyRetriesLeft	681
8.731unpack_dms_UIMGetICCID_t Struct Reference	681
8.731.1 Detailed Description	681
8.731.2 Field Documentation	681
8.731.2.1 ParamPresenceMask	681
8.731.2.2 String	681
8.731.2.3 stringSize	681
8.731.2.4 Tlvresult	681

8.732unpack_dms_UIMGetPINStatus_t Struct Reference	682
8.732.1 Detailed Description	682
8.732.2 Field Documentation	683
8.732.2.1 p1Status	683
8.732.2.2 p1UnblockRetriesLeft	683
8.732.2.3 p1VerifyRetriesLeft	683
8.732.2.4 p2Status	683
8.732.2.5 p2UnblockRetriesLeft	683
8.732.2.6 p2VerifyRetriesLeft	683
8.732.2.7 ParamPresenceMask	683
8.732.2.8 Tlvresult	683
8.733unpack_dms_UIMSetControlKeyProtection_t Struct Reference	683
8.733.1 Detailed Description	683
8.733.2 Field Documentation	684
8.733.2.1 ParamPresenceMask	684
8.733.2.2 Tlvresult	684
8.733.2.3 verifyRetriesLeft	684
8.734unpack_dms_UIMSetPINProtection_t Struct Reference	684
8.734.1 Detailed Description	684
8.734.2 Field Documentation	685
8.734.2.1 ParamPresenceMask	685
8.734.2.2 Tlvresult	685
8.734.2.3 unblockRetriesLeft	685
8.734.2.4 verifyRetriesLeft	685
8.735unpack_dms_UIMUnblockControlKey_t Struct Reference	685
8.735.1 Detailed Description	685
8.735.2 Field Documentation	685
8.735.2.1 ParamPresenceMask	685
8.735.2.2 Tlvresult	685
8.735.2.3 unblockRetriesLeft	685
8.736unpack_fms_GetImagesPreference_t Struct Reference	685
8.736.1 Detailed Description	686
8.736.2 Field Documentation	686
8.736.2.1 ImageListSize	686
8.736.2.2 ParamPresenceMask	686
8.736.2.3 pImageList	686
8.736.2.4 Tlvresult	686
8.737unpack_fms_GetStoredImages_t Struct Reference	686
8.737.1 Detailed Description	686
8.737.2 Field Documentation	687

8.737.2.1	imageList	687
8.737.2.2	imagelistSize	687
8.737.2.3	ParamPresenceMask	687
8.737.2.4	Tlvresult	687
8.738	unpack_fms_SetImagesPreference_t Struct Reference	687
8.738.1	Detailed Description	687
8.738.2	Field Documentation	687
8.738.2.1	ImageTypes	687
8.738.2.2	ImageTypesSize	687
8.738.2.3	ParamPresenceMask	687
8.738.2.4	Tlvresult	687
8.739	unpack_ims_SLQSGetIMSSMSConfig_t Struct Reference	687
8.739.1	Detailed Description	688
8.739.2	Field Documentation	688
8.739.2.1	ParamPresenceMask	688
8.739.2.2	pPhoneCtxtURI	688
8.739.2.3	pPhoneCtxtURILen	688
8.739.2.4	pSettingResp	688
8.739.2.5	pSMSFormat	688
8.739.2.6	pSMSOverIPNwInd	688
8.740	unpack_ims_SLQSGetIMSUserConfig_t Struct Reference	688
8.740.1	Detailed Description	689
8.740.2	Field Documentation	689
8.740.2.1	ParamPresenceMask	689
8.740.2.2	pIMSDomain	689
8.740.2.3	pIMSDomainLen	689
8.740.2.4	pSettingResp	689
8.741	unpack_ims_SLQSGetIMSVoIPConfig_t Struct Reference	689
8.741.1	Detailed Description	690
8.741.2	Field Documentation	691
8.741.2.1	pAmrMode	691
8.741.2.2	pAmrOctetAligned	691
8.741.2.3	pAmrWbEnable	691
8.741.2.4	pAmrWBMode	692
8.741.2.5	pAmrWBOctetAligned	692
8.741.2.6	ParamPresenceMask	692
8.741.2.7	pMinSessionExpiryTimer	692
8.741.2.8	pRingBackTimer	692
8.741.2.9	pRingingTimer	692
8.741.2.10	pRTPRTCPInactTimer	692

8.741.2.1	pScrAmrEnable	692
8.741.2.12	pScrAmrWbEnable	692
8.741.2.13	pSessionExpiryTimer	692
8.741.2.14	pSettingResp	692
8.742	unpack_ims_SLQSGetRegMgrConfig_t Struct Reference	692
8.742.1	Detailed Description	692
8.742.2	Field Documentation	693
8.742.2.1	ParamPresenceMask	693
8.742.2.2	pIMSTestMode	693
8.742.2.3	pPCSCFPort	693
8.742.2.4	pPriCSCFPortName	693
8.742.2.5	pPriCSCFPortNameLen	693
8.742.2.6	pSettingResp	693
8.743	unpack_ims_SLQSGetSIPConfig_t Struct Reference	693
8.743.1	Detailed Description	693
8.743.2	Field Documentation	694
8.743.2.1	ParamPresenceMask	694
8.743.2.2	pSettingResp	694
8.743.2.3	pSigCompEnabled	694
8.743.2.4	pSIPLocalPort	694
8.743.2.5	pSubscribeTimer	694
8.743.2.6	pTimerSIPReg	694
8.743.2.7	pTimerT1	694
8.743.2.8	pTimerT2	694
8.743.2.9	pTimerTf	694
8.744	unpack_ims_SLQSRegMgrCfgCallBack_ind_t Struct Reference	694
8.744.1	Detailed Description	695
8.744.2	Field Documentation	695
8.744.2.1	ParamPresenceMask	695
8.744.2.2	PCTlv	695
8.744.2.3	PNTlv	695
8.744.2.4	TMTlv	695
8.745	unpack_ims_SLQSSetIMSSMSConfig_t Struct Reference	695
8.745.1	Detailed Description	695
8.745.2	Field Documentation	696
8.745.2.1	ParamPresenceMask	696
8.745.2.2	pSettingResp	696
8.746	unpack_ims_SLQSSetIMSUserConfig_t Struct Reference	696
8.746.1	Detailed Description	696
8.746.2	Field Documentation	696

8.746.2.1 ParamPresenceMask	696
8.746.2.2 pSettingResp	696
8.747unpack_ims_SLQSSetIMSVoIPConfig_t Struct Reference	696
8.747.1 Detailed Description	696
8.747.2 Field Documentation	697
8.747.2.1 ParamPresenceMask	697
8.747.2.2 pSettingResp	697
8.748unpack_ims_SLQSSetRegMgrConfig_t Struct Reference	697
8.748.1 Detailed Description	697
8.748.2 Field Documentation	697
8.748.2.1 ParamPresenceMask	697
8.748.2.2 pSettingResp	697
8.749unpack_ims_SLQSSetSIPConfig_t Struct Reference	697
8.749.1 Detailed Description	697
8.749.2 Field Documentation	698
8.749.2.1 ParamPresenceMask	698
8.749.2.2 pSettingResp	698
8.750unpack_ims_SLQSSIPCfgCallBack_ind_t Struct Reference	698
8.750.1 Detailed Description	698
8.750.2 Field Documentation	699
8.750.2.1 ParamPresenceMask	699
8.750.2.2 SCTlv	699
8.750.2.3 SPTlv	699
8.750.2.4 SRTlv	699
8.750.2.5 STTlv	699
8.750.2.6 TT1Tlv	699
8.750.2.7 TT2Tlv	699
8.750.2.8 TTfTlv	699
8.751unpack_ims_SLQSSMSCfgCallBack_ind_t Struct Reference	699
8.751.1 Detailed Description	699
8.751.2 Field Documentation	700
8.751.2.1 ParamPresenceMask	700
8.751.2.2 PCURTlv	700
8.751.2.3 SFTlv	700
8.751.2.4 SINTlv	700
8.752unpack_ims_SLQSUserCfgCallBack_ind_t Struct Reference	700
8.752.1 Detailed Description	700
8.752.2 Field Documentation	701
8.752.2.1 IDTlv	701
8.752.2.2 ParamPresenceMask	701

8.753unpack_ims_SLQSVolPCfgCallBack_ind_t Struct Reference	701
8.753.1 Detailed Description	701
8.753.2 Field Documentation	702
8.753.2.1 AMTlv	702
8.753.2.2 AOATlv	702
8.753.2.3 AWMtlv	702
8.753.2.4 AWOATlv	703
8.753.2.5 EAWTlv	703
8.753.2.6 ESATlv	703
8.753.2.7 ESAWTlv	703
8.753.2.8 MSETlv	703
8.753.2.9 ParamPresenceMask	703
8.753.2.10RBTTlv	703
8.753.2.11RTIDTlv	703
8.753.2.12RTTlv	703
8.753.2.13SDTlv	703
8.754unpack_imsa_SLQSGetIMSARegStatus_t Struct Reference	703
8.754.1 Detailed Description	703
8.754.2 Field Documentation	704
8.754.2.1 ImsRegErrCode	704
8.754.2.2 ImsRegStatus	704
8.754.2.3 NewImsRegStatus	704
8.754.2.4 ParamPresenceMask	704
8.755unpack_imsa_SLQSGetIMSAServiceStatus_t Struct Reference	704
8.755.1 Detailed Description	704
8.755.2 Field Documentation	706
8.755.2.1 ParamPresenceMask	706
8.755.2.2 SmsServiceRat	706
8.755.2.3 SmsServiceStatus	706
8.755.2.4 UtServiceRat	706
8.755.2.5 UtServiceStatus	706
8.755.2.6 VoipServiceRat	706
8.755.2.7 VoipServiceStatus	706
8.755.2.8 VsServiceRat	706
8.755.2.9 VsServiceStatus	706
8.755.2.10VtServiceRat	706
8.755.2.11VtServiceStatus	706
8.756unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t Struct Reference	706
8.756.1 Detailed Description	707
8.756.2 Field Documentation	707

8.756.2.1 FailErrCode	707
8.756.2.2 ParamPresenceMask	707
8.756.2.3 PdpConnState	707
8.757unpack_imsa_SLQSImsaRatStatusCallBack_ind_t Struct Reference	707
8.757.1 Detailed Description	707
8.757.2 Field Documentation	708
8.757.2.1 ParamPresenceMask	708
8.757.2.2 RatHandover	708
8.758unpack_imsa_SLQSImsaRegStatusCallBack_ind_t Struct Reference	708
8.758.1 Detailed Description	708
8.758.2 Field Documentation	708
8.758.2.1 IMSRegistration	708
8.758.2.2 IMSRegistrationError	708
8.758.2.3 NewIMSRegistration	708
8.758.2.4 ParamPresenceMask	709
8.759unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t Struct Reference	709
8.759.1 Detailed Description	709
8.759.2 Field Documentation	710
8.759.2.1 ParamPresenceMask	710
8.759.2.2 SmsRat	710
8.759.2.3 SmsService	710
8.759.2.4 UtRat	710
8.759.2.5 UtService	710
8.759.2.6 VoipRat	710
8.759.2.7 VoipService	710
8.759.2.8 VtRat	710
8.759.2.9 VtService	710
8.760unpack_loc_BestAvailPos_Ind_t Struct Reference	710
8.760.1 Detailed Description	711
8.760.2 Field Documentation	716
8.760.2.1 pAltitudeWrtEllipsoid	716
8.760.2.2 pAltitudeWrtMeanSeaLevel	716
8.760.2.3 ParamPresenceMask	716
8.760.2.4 pGpsTime	716
8.760.2.5 pHeading	716
8.760.2.6 pHeadingUnc	716
8.760.2.7 pHorCirConf	716
8.760.2.8 pHorEllpConf	716
8.760.2.9 pHorReliability	716
8.760.2.10pHorUncCircular	716

8.760.2.11pHorUncEllipseOrientAzimuth	716
8.760.2.12pHorUncEllipseSemiMajor	716
8.760.2.13pHorUncEllipseSemiMinor	716
8.760.2.14pLatitude	716
8.760.2.15pLongitude	716
8.760.2.16pMagneticDeviation	716
8.760.2.17pPrecisionDilution	716
8.760.2.18pSensorDataUsage	716
8.760.2.19pSpeedHorizontal	716
8.760.2.20pSpeedUnc	716
8.760.2.21pSpeedVertical	716
8.760.2.22pSpeedVerticalUnc	716
8.760.2.23pSvUsedforFix	716
8.760.2.24pTechnologyMask	716
8.760.2.25pTimeSrc	716
8.760.2.26pTimestampUtc	716
8.760.2.27pTimeUnc	716
8.760.2.28pVertConfidence	717
8.760.2.29pVertReliability	717
8.760.2.30pVertUnc	717
8.760.2.31pXid	717
8.760.2.32status	717
8.760.2.33Tlvresult	717
8.761unpack_loc_CradleMountCallback_Ind_t Struct Reference	717
8.761.1 Detailed Description	717
8.761.2 Field Documentation	717
8.761.2.1 cradleMountConfigStatus	717
8.761.2.2 ParamPresenceMask	717
8.762unpack_loc_Delete_Assist_Data_t Struct Reference	717
8.762.1 Detailed Description	718
8.762.2 Field Documentation	718
8.762.2.1 ParamPresenceMask	718
8.762.2.2 Tlvresult	718
8.763unpack_loc_DeleteAssistData_Ind_t Struct Reference	718
8.763.1 Detailed Description	718
8.763.2 Field Documentation	719
8.763.2.1 ParamPresenceMask	719
8.763.2.2 status	719
8.763.2.3 Tlvresult	719
8.764unpack_loc_EngineState_Ind_t Struct Reference	719

8.764.1 Detailed Description	719
8.764.2 Field Documentation	720
8.764.2.1 engineState	720
8.764.2.2 ParamPresenceMask	720
8.764.2.3 Tlvresult	720
8.765unpack_loc_EventNMEA_Ind_t Struct Reference	720
8.765.1 Detailed Description	720
8.765.2 Field Documentation	720
8.765.2.1 NMEADData	720
8.765.2.2 ParamPresenceMask	720
8.765.2.3 Tlvresult	720
8.766unpack_loc_EventRegister_t Struct Reference	721
8.766.1 Detailed Description	721
8.766.2 Field Documentation	721
8.766.2.1 ParamPresenceMask	721
8.766.2.2 Tlvresult	721
8.767unpack_loc_EventTimeSyncCallback_Ind_t Struct Reference	721
8.767.1 Detailed Description	721
8.767.2 Field Documentation	721
8.767.2.1 ParamPresenceMask	721
8.767.2.2 timeSyncRefCounter	721
8.768unpack_loc_GetOpMode_Ind_t Struct Reference	721
8.768.1 Detailed Description	722
8.768.2 Field Documentation	723
8.768.2.1 ParamPresenceMask	723
8.768.2.2 pMode	723
8.768.2.3 Status	723
8.769unpack_loc_GetServer_Ind_t Struct Reference	723
8.769.1 Detailed Description	723
8.769.2 Field Documentation	724
8.769.2.1 ParamPresenceMask	724
8.769.2.2 pIPv4AddrInfo	724
8.769.2.3 pIPv6AddrInfo	724
8.769.2.4 pURL	724
8.769.2.5 serverStatus	724
8.769.2.6 serverType	724
8.769.2.7 Tlvresult	724
8.770unpack_loc_GnssSvInfo_Ind_t Struct Reference	724
8.770.1 Detailed Description	724
8.770.2 Field Documentation	725

8.770.2.1 altitudeAssumed	725
8.770.2.2 ParamPresenceMask	725
8.770.2.3 pSatelliteInfo	725
8.770.2.4 Tlvresult	725
8.771unpack_loc_InjectPositionCallback_Ind_t Struct Reference	725
8.771.1 Detailed Description	725
8.771.2 Field Documentation	726
8.771.2.1 ParamPresenceMask	726
8.771.2.2 status	726
8.772unpack_loc_InjectSensorDataCallback_Ind_t Struct Reference	726
8.772.1 Detailed Description	726
8.772.2 Field Documentation	727
8.772.2.1 injectSensorDataStatus	727
8.772.2.2 pAccelSamplesAccepted	727
8.772.2.3 pAccelTempSamplesAccepted	728
8.772.2.4 ParamPresenceMask	728
8.772.2.5 pGyroSamplesAccepted	728
8.772.2.6 pGyroTempSamplesAccepted	728
8.772.2.7 pOpaqueIdentifier	728
8.773unpack_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference	728
8.773.1 Detailed Description	728
8.773.2 Field Documentation	728
8.773.2.1 injectTimeSyncStatus	728
8.773.2.2 ParamPresenceMask	728
8.774unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference	728
8.774.1 Detailed Description	729
8.774.2 Field Documentation	729
8.774.2.1 ParamPresenceMask	729
8.774.2.2 status	729
8.775unpack_loc_PositionRpt_Ind_t Struct Reference	729
8.775.1 Detailed Description	730
8.775.2 Field Documentation	735
8.775.2.1 pAltitudeAssumed	735
8.775.2.2 pAltitudeWrtEllipsoid	735
8.775.2.3 pAltitudeWrtMeanSeaLevel	735
8.775.2.4 ParamPresenceMask	735
8.775.2.5 pFixId	735
8.775.2.6 pGpsTime	735
8.775.2.7 pHeading	735
8.775.2.8 pHeadingUnc	735

8.775.2.9 pHorConfidence	735
8.775.2.10 pHorReliability	735
8.775.2.11 pHorUncCircular	735
8.775.2.12 pHorUncEllipseOrientAzimuth	735
8.775.2.13 pHorUncEllipseSemiMajor	735
8.775.2.14 pHorUncEllipseSemiMinor	735
8.775.2.15 pLatitude	735
8.775.2.16 pLeapSeconds	735
8.775.2.17 pLongitude	735
8.775.2.18 pMagneticDeviation	735
8.775.2.19 pPrecisionDilution	735
8.775.2.20 pSensorDataUsage	735
8.775.2.21 pSpeedHorizontal	735
8.775.2.22 pSpeedUnc	735
8.775.2.23 pSpeedVertical	735
8.775.2.24 pSvUsedforFix	735
8.775.2.25 pTechnologyMask	735
8.775.2.26 pTimeSrc	735
8.775.2.27 pTimestampUtc	735
8.775.2.28 pTimeUnc	736
8.775.2.29 pVertConfidence	736
8.775.2.30 pVertReliability	736
8.775.2.31 pVertUnc	736
8.775.2.32 sessionId	736
8.775.2.33 sessionStatus	736
8.775.2.34 Tlvresult	736
8.776unpack_loc_SensorStreamingCallback_Ind_t Struct Reference	736
8.776.1 Detailed Description	736
8.776.2 Field Documentation	736
8.776.2.1 pAccelAcceptReady	736
8.776.2.2 pAccelTempAcceptReady	736
8.776.2.3 ParamPresenceMask	737
8.776.2.4 pGyroAcceptReady	737
8.776.2.5 pGyroTempAcceptReady	737
8.777unpack_loc_SetExtPowerConfig_Ind_t Struct Reference	737
8.777.1 Detailed Description	737
8.777.2 Field Documentation	737
8.777.2.1 ParamPresenceMask	737
8.777.2.2 status	737
8.777.2.3 Tlvresult	737

8.778unpack_loc_SetExtPowerState_t Struct Reference	737
8.778.1 Detailed Description	738
8.778.2 Field Documentation	738
8.778.2.1 ParamPresenceMask	738
8.778.2.2 Tlvresult	738
8.779unpack_loc_SetOperationMode_Ind_t Struct Reference	738
8.779.1 Detailed Description	738
8.779.2 Field Documentation	739
8.779.2.1 ParamPresenceMask	739
8.779.2.2 status	739
8.779.2.3 Tlvresult	739
8.780unpack_loc_SetOperationMode_t Struct Reference	739
8.780.1 Detailed Description	739
8.780.2 Field Documentation	739
8.780.2.1 ParamPresenceMask	739
8.780.2.2 Tlvresult	739
8.781unpack_loc_SetServer_Ind_t Struct Reference	739
8.781.1 Detailed Description	739
8.781.2 Field Documentation	740
8.781.2.1 ParamPresenceMask	740
8.781.2.2 serverStatus	740
8.781.2.3 Tlvresult	740
8.782unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	740
8.782.1 Detailed Description	740
8.782.2 Field Documentation	741
8.782.2.1 ParamPresenceMask	741
8.782.2.2 Tlvresult	741
8.783unpack_loc_SLQSLOCGetOpMode_t Struct Reference	741
8.783.1 Detailed Description	741
8.783.2 Field Documentation	741
8.783.2.1 ParamPresenceMask	741
8.783.2.2 Tlvresult	741
8.784unpack_loc_Start_t Struct Reference	741
8.784.1 Detailed Description	741
8.784.2 Field Documentation	742
8.784.2.1 ParamPresenceMask	742
8.784.2.2 Tlvresult	742
8.785unpack_loc_Stop_t Struct Reference	742
8.785.1 Detailed Description	742
8.785.2 Field Documentation	742

8.785.2.1 ParamPresenceMask	742
8.785.2.2 Tlvresult	742
8.786unpack_nas_GetACCOLC_t Struct Reference	742
8.786.1 Detailed Description	742
8.786.2 Field Documentation	743
8.786.2.1 pAccolc	743
8.786.2.2 ParamPresenceMask	743
8.787unpack_nas_GetANAAAAAuthenticationStatus_t Struct Reference	743
8.787.1 Detailed Description	743
8.787.2 Field Documentation	743
8.787.2.1 ParamPresenceMask	743
8.787.2.2 pAuthStatus	743
8.788unpack_nas_GetCDMANetworkParameters_t Struct Reference	743
8.788.1 Detailed Description	744
8.788.2 Field Documentation	745
8.788.2.1 Application	745
8.788.2.2 Broadcast	745
8.788.2.3 CustomSCP	745
8.788.2.4 ForceRev0	745
8.788.2.5 ParamPresenceMask	745
8.788.2.6 Protocol	745
8.788.2.7 RegForeignNID	745
8.788.2.8 RegForeignSID	746
8.788.2.9 RegHomeSID	746
8.788.2.10Roaming	746
8.788.2.11SCI	746
8.788.2.12SCM	746
8.789unpack_nas_GetHomeNetwork3GPP2_t Struct Reference	746
8.789.1 Detailed Description	746
8.789.2 Field Documentation	748
8.789.2.1 nameSize	748
8.789.2.2 ParamPresenceMask	748
8.789.2.3 pMCC	748
8.789.2.4 pMNC	748
8.789.2.5 pName	748
8.789.2.6 pNID	748
8.789.2.7 pNw2DescDisp	748
8.789.2.8 pNw2DescEnc	748
8.789.2.9 pNw2DescLen	748
8.789.2.10pNw2MCC	748

8.789.2.1 1pNw2MNC	748
8.789.2.1 2pNw2Name	748
8.789.2.1 3pSID	748
8.790 unpack_nas_GetHomeNetwork_t Struct Reference	748
8.790.1 Detailed Description	748
8.790.2 Field Documentation	749
8.790.2.1 mcc	749
8.790.2.2 mnc	749
8.790.2.3 name	749
8.790.2.4 nid	749
8.790.2.5 ParamPresenceMask	749
8.790.2.6 sid	749
8.791 unpack_nas_GetNetworkPreference_t Struct Reference	749
8.791.1 Detailed Description	749
8.791.2 Field Documentation	750
8.791.2.1 ActiveTechPref	750
8.791.2.2 Duration	750
8.791.2.3 ParamPresenceMask	750
8.791.2.4 PersistentTechPref	750
8.791.2.5 Tlvresult	750
8.792 unpack_nas_GetRFInfo_t Struct Reference	750
8.792.1 Detailed Description	751
8.792.2 Field Documentation	751
8.792.2.1 instancesSize	751
8.792.2.2 ParamPresenceMask	751
8.792.2.3 RFBandInfoElements	751
8.793 unpack_nas_GetServingNetwork_t Struct Reference	751
8.793.1 Detailed Description	752
8.793.2 Field Documentation	753
8.793.2.1 CSDomain	753
8.793.2.2 DataCaps	753
8.793.2.3 DataCapsLen	753
8.793.2.4 MCC	753
8.793.2.5 MNC	753
8.793.2.6 Name	754
8.793.2.7 nameSize	754
8.793.2.8 ParamPresenceMask	754
8.793.2.9 PSDomain	754
8.793.2.10 Radiolfaces	754
8.793.2.11 RadiolfacesSize	754

8.793.2.12 RAN	754
8.793.2.13 RegistrationState	754
8.793.2.14 Roaming	754
8.794unpack_nas_GetServingNetworkCapabilities_t Struct Reference	754
8.794.1 Detailed Description	754
8.794.2 Field Documentation	755
8.794.2.1 DataCaps	755
8.794.2.2 DataCapsLen	755
8.794.2.3 ParamPresenceMask	755
8.795unpack_nas_GetSignalStrengths_t Struct Reference	755
8.795.1 Detailed Description	755
8.795.2 Field Documentation	755
8.795.2.1 len	755
8.795.2.2 ParamPresenceMask	755
8.795.2.3 radio	755
8.795.2.4 rssi	755
8.796unpack_nas_PerformNetworkScan_t Struct Reference	756
8.796.1 Detailed Description	756
8.796.2 Field Documentation	757
8.796.2.1 p3GppNetworkInfoInstances	757
8.796.2.2 p3GppNetworkInstanceSize	757
8.796.2.3 ParamPresenceMask	757
8.796.2.4 pLteOpModeTlv	757
8.796.2.5 pPCIInfo	757
8.796.2.6 pPCSInstance	757
8.796.2.7 pPCSInstanceSize	757
8.796.2.8 pRATInstance	757
8.796.2.9 pRATInstanceSize	757
8.796.2.10 pScanResult	757
8.797unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	757
8.797.1 Detailed Description	758
8.797.2 Field Documentation	758
8.797.2.1 dataCaps	758
8.797.2.2 dataCapsSize	758
8.797.2.3 ParamPresenceMask	758
8.798unpack_nas_SetEventReportInd_t Struct Reference	758
8.798.1 Detailed Description	758
8.798.2 Field Documentation	759
8.798.2.1 ParamPresenceMask	759
8.798.2.2 RFTlv	759

8.798.2.3 RRTlv	759
8.798.2.4 SLQSSSTlv	759
8.798.2.5 SSTlv	759
8.799unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference	759
8.799.1 Detailed Description	759
8.799.2 Field Documentation	760
8.799.2.1 ParamPresenceMask	760
8.799.2.2 sPhyCaAggPcellInfo	760
8.799.2.3 sPhyCaAggScellDIBw	760
8.799.2.4 sPhyCaAggScellIndex	760
8.799.2.5 sPhyCaAggScellIndType	760
8.799.2.6 sPhyCaAggScellInfo	760
8.800unpack_nas_SetNetworkPreference_t Struct Reference	760
8.800.1 Detailed Description	760
8.800.2 Field Documentation	760
8.800.2.1 ParamPresenceMask	761
8.800.2.2 Tlvresult	761
8.801unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference	761
8.801.1 Detailed Description	761
8.801.2 Field Documentation	761
8.801.2.1 ParamPresenceMask	761
8.801.2.2 roaming	761
8.802unpack_nas_SetServingSystemCallback_ind_t Struct Reference	761
8.802.1 Detailed Description	761
8.802.2 Field Documentation	762
8.802.2.1 ParamPresenceMask	762
8.802.2.2 SSInfo	762
8.802.2.3 Tlvresult	762
8.803unpack_nas_SLQSGetErrorRate_t Struct Reference	762
8.803.1 Detailed Description	762
8.803.2 Field Documentation	763
8.803.2.1 ParamPresenceMask	763
8.803.2.2 pCDMAFrameErrRate	763
8.803.2.3 pGSMBER	763
8.803.2.4 pHDRPackErrRate	763
8.803.2.5 pWCDMABER	763
8.804unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference	763
8.804.1 Detailed Description	763
8.804.2 Field Documentation	764
8.804.2.1 LTECphyCAInfo	764

8.804.2.2 ParamPresenceMask	764
8.804.2.3 Tlvresult	764
8.805unpack_nas_SLQSGetNetworkTime_t Struct Reference	764
8.805.1 Detailed Description	764
8.805.2 Field Documentation	764
8.805.2.1 p3GPP2TimeInfo	764
8.805.2.2 p3GPPTimeInfo	764
8.805.2.3 ParamPresenceMask	764
8.806unpack_nas_SLQSGetOperatorNameData_t Struct Reference	764
8.806.1 Detailed Description	765
8.806.2 Field Documentation	765
8.806.2.1 ParamPresenceMask	765
8.806.2.2 pNITZInformation	765
8.806.2.3 pOperatorNameString	765
8.806.2.4 pOperatorPLMNList	765
8.806.2.5 pPLMNNetworkName	765
8.806.2.6 pSvcProviderName	765
8.807unpack_nas_SLQSGetPLMNName_t Struct Reference	766
8.807.1 Detailed Description	766
8.807.2 Field Documentation	768
8.807.2.1 longName	768
8.807.2.2 longNameCI	768
8.807.2.3 longNameEn	768
8.807.2.4 longNameLen	768
8.807.2.5 longNameSB	768
8.807.2.6 ParamPresenceMask	768
8.807.2.7 shortName	768
8.807.2.8 shortNameCI	768
8.807.2.9 shortNameEn	768
8.807.2.10shortNameLen	768
8.807.2.11shortNameSB	768
8.807.2.12spn	768
8.807.2.13spnEncoding	769
8.807.2.14spnLength	769
8.808unpack_nas_SLQSGetServingSystem_t Struct Reference	769
8.808.1 Detailed Description	769
8.808.2 Field Documentation	772
8.808.2.1 BasestationID	772
8.808.2.2 BasestationLatitude	772
8.808.2.3 BasestationLongitude	772

8.808.2.4 CallBarStatus	772
8.808.2.5 CDMA_P_Rev	772
8.808.2.6 CDMASystemInfoExt	772
8.808.2.7 CellID	773
8.808.2.8 ConcSvcInfo	773
8.808.2.9 CurrentPLMN	773
8.808.2.10DataSrvCapabilities	773
8.808.2.11DefaultRoamInd	773
8.808.2.12DetailedSvcInfo	773
8.808.2.13DTMInd	773
8.808.2.14Gpp2TimeZone	773
8.808.2.15GppNetworkDSTAdjustment	773
8.808.2.16GppTimeZone	773
8.808.2.17HdrPersonality	773
8.808.2.18Lac	773
8.808.2.19NetworkID	773
8.808.2.20ParamPresenceMask	773
8.808.2.21PRLInd	773
8.808.2.22RoamIndicatorVal	773
8.808.2.23RoamingIndicatorList	773
8.808.2.24ServingSystem	773
8.808.2.25SystemID	773
8.808.2.26TrackAreaCode	773
8.809unpack_nas_SLQSGetSignalStrength_t Struct Reference	773
8.809.1 Detailed Description	774
8.809.2 Field Documentation	775
8.809.2.1 ecioList	775
8.809.2.2 ecioListLen	775
8.809.2.3 errorRateList	775
8.809.2.4 errorRateListLen	775
8.809.2.5 lo	776
8.809.2.6 ltersrp	776
8.809.2.7 ltesnr	776
8.809.2.8 ParamPresenceMask	776
8.809.2.9 rsrqInfo	776
8.809.2.10rxSignalStrengthList	776
8.809.2.11rxSignalStrengthListLen	776
8.809.2.12signalStrengthReqMask	776
8.809.2.13sinr	776
8.810unpack_nas_SLQSGetSysInfo_t Struct Reference	776

8.810.1 Detailed Description	776
8.810.2 Field Documentation	779
8.810.2.1 pAddCDMASysInfo	779
8.810.2.2 pAddGSMSysInfo	779
8.810.2.3 pAddHDRSysInfo	779
8.810.2.4 pAddLTESysInfo	779
8.810.2.5 pAddWCDMASysInfo	779
8.810.2.6 ParamPresenceMask	779
8.810.2.7 pCampedCiotLteOpMode	779
8.810.2.8 pCDMASrvStatusInfo	779
8.810.2.9 pCDMASysInfo	779
8.810.2.10pGSMCallBarringSysInfo	779
8.810.2.11pGSMCipherDomainSysInfo	779
8.810.2.12pGSMSrvStatusInfo	779
8.810.2.13pGSMSysInfo	779
8.810.2.14pHDRSrvStatusInfo	779
8.810.2.15pHDRSysInfo	779
8.810.2.16pLTESrvStatusInfo	779
8.810.2.17pLTESysInfo	779
8.810.2.18pLTEVoiceSupportSysInfo	779
8.810.2.19pWCDMACallBarringSysInfo	779
8.810.2.20pWCDMACipherDomainSysInfo	780
8.810.2.21pWCDMASrvStatusInfo	780
8.810.2.22pWCDMASysInfo	780
8.811unpack_nas_SLQSGetSysSelectionPref_t Struct Reference	780
8.811.1 Detailed Description	780
8.811.2 Field Documentation	784
8.811.2.1 ParamPresenceMask	784
8.811.2.2 pBandPref	784
8.811.2.3 pEmerMode	784
8.811.2.4 pGWAcqOrderPref	784
8.811.2.5 pLTEBandPref	784
8.811.2.6 pModePref	784
8.811.2.7 pNetSelPref	784
8.811.2.8 pPRLPref	784
8.811.2.9 pRoamPref	784
8.811.2.10pSrvDomainPref	784
8.812unpack_nas_SLQSGetSysSelectionPrefExt_t Struct Reference	784
8.812.1 Detailed Description	785
8.812.2 Field Documentation	786

8.812.2.1 pAcqOrderPref	786
8.812.2.2 pBandPref	786
8.812.2.3 pCiotAcqOrderPref	786
8.812.2.4 pCiotLteOpModePref	787
8.812.2.5 pEmerMode	787
8.812.2.6 pGWAcqOrderPref	787
8.812.2.7 pLTEBandPref	787
8.812.2.8 pLTEBandPrefExt	787
8.812.2.9 pLteM1BandPref	787
8.812.2.10pLteNb1BandPref	787
8.812.2.11pModePref	787
8.812.2.12pNetSelPref	787
8.812.2.13pNr5gBandPref	787
8.812.2.14pPRLPref	787
8.812.2.15pRatDisabledMask	787
8.812.2.16pRoamPref	787
8.812.2.17pSrvDomainPref	787
8.813unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t Struct Reference	787
8.813.1 Detailed Description	787
8.813.2 Field Documentation	788
8.813.2.1 edrxCiotLteModeTlv	788
8.813.2.2 edrxCycleLengthTlv	788
8.813.2.3 edrxEnableTypeTlv	788
8.813.2.4 edrxPagingTimeWindowTlv	788
8.813.2.5 edrxRatTypeTlv	788
8.813.2.6 ParamPresenceMask	788
8.814unpack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference	788
8.814.1 Detailed Description	788
8.814.2 Field Documentation	789
8.814.2.1 ParamPresenceMask	789
8.814.2.2 pCDMAChannel	789
8.814.2.3 pDirNum	789
8.814.2.4 pHomeSIDNID	789
8.814.2.5 pMinBasedIMSI	789
8.814.2.6 pNAMNameInfo	789
8.814.2.7 pTrueIMSI	789
8.815unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference	789
8.815.1 Detailed Description	790
8.815.2 Field Documentation	790
8.815.2.1 ParamPresenceMask	791

8.815.2.2 pCDMAInfo	791
8.815.2.3 pGERANInfo	791
8.815.2.4 pLTEInfoInterfreq	791
8.815.2.5 pLTEInfoIntrafreq	791
8.815.2.6 pLTEInfoNeighboringGSM	791
8.815.2.7 pLTEInfoNeighboringWCDMA	791
8.815.2.8 pUMTSCellID	791
8.815.2.9 pUMTSInfo	791
8.815.2.10 pWCDMAInfoLTENeighborCell	791
8.816unpack_nas_SLQSNASGeteDRXParams_t Struct Reference	791
8.816.1 Detailed Description	791
8.816.2 Field Documentation	792
8.816.2.1 ParamPresenceMask	792
8.816.2.2 pCycleLen	792
8.816.2.3 pEdrxEnable	792
8.816.2.4 pPagingTimeWindow	792
8.817unpack_nas_SLQSNASGeteDRXParamsExt_t Struct Reference	792
8.817.1 Detailed Description	792
8.817.2 Field Documentation	793
8.817.2.1 ParamPresenceMask	793
8.817.2.2 pCycleLen	793
8.817.2.3 pEdrxEnable	793
8.817.2.4 pEdrxRAT	793
8.817.2.5 pLteOpMode	793
8.817.2.6 pPagingTimeWindow	793
8.818unpack_nas_SLQSNASGetForbiddenNetworks_t Struct Reference	793
8.818.1 Detailed Description	794
8.818.2 Field Documentation	794
8.818.2.1 ParamPresenceMask	794
8.818.2.2 pForbiddenNetworks3GPP	794
8.819unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference	794
8.819.1 Detailed Description	794
8.819.2 Field Documentation	794
8.819.2.1 ParamPresenceMask	794
8.819.2.2 pColorCode	794
8.820unpack_nas_SLQSNasGetRFInfo_t Struct Reference	794
8.820.1 Detailed Description	795
8.820.2 Field Documentation	795
8.820.2.1 pLTEOperationMode	795
8.820.2.2 pRfBandInfoExtFormat	795

8.820.2.3 pRfBandwidthInfo	795
8.820.2.4 pRfDedicatedBandInfo	795
8.820.2.5 rfbandInfoList	795
8.820.2.6 Tlvresult	795
8.821unpack_nas_SLQSNasGetSigInfo_t Struct Reference	795
8.821.1 Detailed Description	796
8.821.2 Field Documentation	796
8.821.2.1 CDMASSInfo	796
8.821.2.2 GSMSSInfo	796
8.821.2.3 HDRSSInfo	796
8.821.2.4 LTESInfo	796
8.821.2.5 ParamPresenceMask	796
8.821.2.6 WCDMAInfo	796
8.822unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference	796
8.822.1 Detailed Description	797
8.822.2 Field Documentation	797
8.822.2.1 ParamPresenceMask	797
8.822.2.2 pRXChain0Info	797
8.822.2.3 pRXChain1Info	797
8.822.2.4 pTXInfo	797
8.823unpack_nas_SLQSNasNetworkRejectCallback_ind_t Struct Reference	797
8.823.1 Detailed Description	797
8.823.2 Field Documentation	800
8.823.2.1 ParamPresenceMask	800
8.823.2.2 pCsgId	800
8.823.2.3 pLteOpMode	800
8.823.2.4 pPlmnId	800
8.823.2.5 radiolId	800
8.823.2.6 rejectCause	800
8.823.2.7 serviceDomain	800
8.824unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference	800
8.824.1 Detailed Description	800
8.824.2 Field Documentation	801
8.824.2.1 ParamPresenceMask	801
8.824.2.2 pDayltSavAdj	801
8.824.2.3 pRadioInterface	801
8.824.2.4 pTimeZone	801
8.824.2.5 universalTime	801
8.825unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference	801
8.825.1 Detailed Description	802

8.825.2 Field Documentation	802
8.825.2.1 ParamPresenceMask	802
8.825.2.2 pCDMASigInfo	802
8.825.2.3 pGSMSigInfo	802
8.825.2.4 pHDRSigInfo	802
8.825.2.5 pLTESigInfo	802
8.825.2.6 pRscp	802
8.825.2.7 pTDSCDMASigInfoExt	803
8.825.2.8 pWCDMASigInfo	803
8.826unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference	803
8.826.1 Detailed Description	803
8.826.2 Field Documentation	803
8.826.2.1 ParamPresenceMask	803
8.826.2.2 pLteEARFCN	803
8.826.2.3 pLtePCI	803
8.826.2.4 pWcdmaUARFCN	803
8.827unpack_nas_SLQSNasSwiModemStatus_t Struct Reference	803
8.827.1 Detailed Description	804
8.827.2 Field Documentation	804
8.827.2.1 commonInfo	804
8.827.2.2 ParamPresenceMask	804
8.827.2.3 pLTEInfo	804
8.828unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference	804
8.828.1 Detailed Description	804
8.828.2 Field Documentation	805
8.828.2.1 Info	805
8.828.2.2 ParamPresenceMask	805
8.828.2.3 Tlvresult	805
8.829unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference	805
8.829.1 Detailed Description	805
8.829.2 Field Documentation	805
8.829.2.1 ParamPresenceMask	805
8.829.2.2 t3396_apn	805
8.829.2.3 t3396_plmn_id	805
8.829.2.4 t3396_val	805
8.830unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference	805
8.830.1 Detailed Description	806
8.830.2 Field Documentation	806
8.830.2.1 Info	806
8.830.2.2 ParamPresenceMask	806

8.830.2.3 Tlvresult	806
8.831unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference	806
8.831.1 Detailed Description	806
8.831.2 Field Documentation	807
8.831.2.1 ParamPresenceMask	807
8.831.2.2 pCurrentPersonality	807
8.831.2.3 pPersonalityListLength	807
8.831.2.4 pProtocolSubtypeElement	807
8.832unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference	807
8.832.1 Detailed Description	807
8.832.2 Field Documentation	808
8.832.2.1 pAppSubType	808
8.832.2.2 ParamPresenceMask	808
8.832.2.3 pCurrentPrsnlty	808
8.832.2.4 pPersonalityListLength	808
8.832.2.5 pProtoSubTypeElmnt	808
8.833unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference	808
8.833.1 Detailed Description	808
8.833.2 Field Documentation	809
8.833.2.1 ParamPresenceMask	809
8.833.2.2 pDRCPParams	809
8.833.2.3 pPilotSetData	809
8.833.2.4 pUATI	809
8.834unpack_nas_SLQSSwiGetLteCQI_t Struct Reference	809
8.834.1 Detailed Description	809
8.834.2 Field Documentation	810
8.834.2.1 CQIValueCW0	810
8.834.2.2 CQIValueCW1	810
8.834.2.3 ParamPresenceMask	810
8.834.2.4 ValidityCW0	810
8.834.2.5 ValidityCW1	810
8.835unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference	810
8.835.1 Detailed Description	810
8.835.2 Field Documentation	810
8.835.2.1 ParamPresenceMask	810
8.835.2.2 pSccRxInfo	810
8.836unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference	810
8.836.1 Detailed Description	810
8.836.2 Field Documentation	811
8.836.2.1 ParamPresenceMask	811

8.836.2.2 pHDRPersInd	811
8.837unpack_nas_SLQSSwiNetworkDebug_t Struct Reference	811
8.837.1 Detailed Description	811
8.837.2 Field Documentation	812
8.837.2.1 ParamPresenceMask	812
8.837.2.2 pDataStatusDetail	812
8.837.2.3 pDeviceConfigDetail	812
8.837.2.4 pNetworkStat1x	812
8.837.2.5 pNetworkStatEVDO	812
8.837.2.6 pObjectVer	812
8.838unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t Struct Reference	812
8.838.1 Detailed Description	812
8.838.2 Field Documentation	812
8.838.2.1 ParamPresenceMask	812
8.838.2.2 rankIndicatorTlv	812
8.839unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference	812
8.839.1 Detailed Description	813
8.839.2 Field Documentation	815
8.839.2.1 pAddCDMASysInfo	816
8.839.2.2 pAddGSMSysInfo	816
8.839.2.3 pAddHDRSysInfo	816
8.839.2.4 pAddLTESysInfo	816
8.839.2.5 pAddWCDMASysInfo	816
8.839.2.6 ParamPresenceMask	816
8.839.2.7 pCDMASrvStatusInfo	816
8.839.2.8 pCDMASysInfo	816
8.839.2.9 pGSMCallBarringSysInfo	816
8.839.2.10pGSMCipherDomainSysInfo	816
8.839.2.11pGSMSrvStatusInfo	816
8.839.2.12pGSMSysInfo	816
8.839.2.13pHDRSrvStatusInfo	816
8.839.2.14pHDRSysInfo	816
8.839.2.15pLteCiotOpModeTlv	816
8.839.2.16pLTESrvStatusInfo	816
8.839.2.17pLTESysInfo	816
8.839.2.18pLTEVoiceSupportSysInfo	816
8.839.2.19pNR5GCellStatus	816
8.839.2.20pNR5GSerStatTlv	816
8.839.2.21pNR5GSystemInfoTlv	816
8.839.2.22pSysInfoNoChange	816

8.839.2.23	pWCDMACallBarringSysInfo	816
8.839.2.24	pWCDMACipherDomainSysInfo	816
8.839.2.25	pWCDMASrvStatusInfo	816
8.839.2.26	pWCDMASysInfo	816
8.840	unpack_omaDmConfigTlv_t Struct Reference	816
8.840.1	Detailed Description	817
8.840.2	Field Documentation	817
8.840.2.1	alertmsg	817
8.840.2.2	alertmsglength	817
8.840.2.3	state	817
8.840.2.4	userInputReq	817
8.840.2.5	userInputTimeout	817
8.841	unpack_omaDmFotaTlv_t Struct Reference	817
8.841.1	Detailed Description	818
8.841.2	Field Documentation	819
8.841.2.1	description	819
8.841.2.2	descriptionlength	819
8.841.2.3	fwloadsize	819
8.841.2.4	fwloadComplete	819
8.841.2.5	namelength	819
8.841.2.6	package_name	819
8.841.2.7	sessionType	819
8.841.2.8	severity	819
8.841.2.9	state	819
8.841.2.10	updateCompleteStatus	819
8.841.2.11	userInputReq	819
8.841.2.12	userInputTimeout	819
8.841.2.13	version	819
8.841.2.14	versionlength	819
8.842	unpack_omaDmNotificationsTlv_t Struct Reference	819
8.842.1	Detailed Description	820
8.842.2	Field Documentation	820
8.842.2.1	notification	820
8.842.2.2	sessionStatus	820
8.843	unpack_pds_GetPDSDDefaults_t Struct Reference	820
8.843.1	Detailed Description	820
8.843.2	Field Documentation	821
8.843.2.1	pAccuracy	821
8.843.2.2	ParamPresenceMask	821
8.843.2.3	pInterval	821

8.843.2.4 pOperation	821
8.843.2.5 pTimeout	821
8.844unpack_pds_GetPDSSState_t Struct Reference	821
8.844.1 Detailed Description	821
8.844.2 Field Documentation	822
8.844.2.1 ParamPresenceMask	822
8.844.2.2 pEnabledStatus	822
8.844.2.3 pTrackingStatus	822
8.845unpack_pds_GetPortAutomaticTracking_t Struct Reference	822
8.845.1 Detailed Description	822
8.845.2 Field Documentation	822
8.845.2.1 ParamPresenceMask	823
8.845.2.2 pbAuto	823
8.846unpack_pds_GetServiceAutomaticTracking_t Struct Reference	823
8.846.1 Detailed Description	823
8.846.2 Field Documentation	823
8.846.2.1 ParamPresenceMask	823
8.846.2.2 pbAuto	823
8.847unpack_pds_GetXTRAAutomaticDownload_t Struct Reference	823
8.847.1 Detailed Description	823
8.847.2 Field Documentation	824
8.847.2.1 ParamPresenceMask	824
8.847.2.2 pbEnabled	824
8.847.2.3 pInterval	824
8.848unpack_pds_GetXTRANetwork_t Struct Reference	824
8.848.1 Detailed Description	824
8.848.2 Field Documentation	824
8.848.2.1 ParamPresenceMask	824
8.848.2.2 pPreference	824
8.849unpack_pds_GetXTRAVality_t Struct Reference	824
8.849.1 Detailed Description	825
8.849.2 Field Documentation	825
8.849.2.1 ParamPresenceMask	825
8.849.2.2 pDuration	825
8.849.2.3 pGPSWeek	825
8.849.2.4 pGPSWeekOffset	825
8.850unpack_pds_SetEventReport_Ind_t Struct Reference	825
8.850.1 Detailed Description	826
8.850.2 Field Documentation	827
8.850.2.1 dLatitude	827

8.850.2.2 dLongitude	827
8.850.2.3 has_dLatitude	827
8.850.2.4 has_dLongitude	827
8.850.2.5 has_PositionDataNMEA	827
8.850.2.6 has_posSrc	827
8.850.2.7 has_SessionStatus	827
8.850.2.8 ParamPresenceMask	827
8.850.2.9 PositionDataNMEA	827
8.850.2.10posSrc	827
8.850.2.11SessionStatus	827
8.851unpack_pds_SetPdsState_Ind_t Struct Reference	827
8.851.1 Detailed Description	827
8.851.2 Field Documentation	828
8.851.2.1 EnabledStatus	828
8.851.2.2 ParamPresenceMask	828
8.851.2.3 TlvPresent	828
8.851.2.4 TrackingStatus	828
8.852unpack_pds_SLQSGetAGPSConfig_t Struct Reference	828
8.852.1 Detailed Description	828
8.852.2 Field Documentation	829
8.852.2.1 ParamPresenceMask	829
8.852.2.2 pServerAddress	829
8.852.2.3 pServerPort	829
8.852.2.4 pServerURL	829
8.852.2.5 pServerURLLength	829
8.853unpack_pds_SLQSGetGPSSStateInfo_t Struct Reference	829
8.853.1 Detailed Description	830
8.853.2 Field Documentation	833
8.853.2.1 Altitude	833
8.853.2.2 EngineState	833
8.853.2.3 glo_almanac_sv_msk	833
8.853.2.4 glo_ephemeris_sv_msk	833
8.853.2.5 glo_health_sv_msk	833
8.853.2.6 glo_visible_sv_msk	833
8.853.2.7 gps_almanac_sv_msk	833
8.853.2.8 gps_ephemeris_sv_msk	833
8.853.2.9 gps_health_sv_msk	833
8.853.2.10gps_visible_sv_msk	833
8.853.2.11HorizontalUncertainty	833
8.853.2.12ono_valid	833

8.853.2.13	Latitude	833
8.853.2.14	Longitude	833
8.853.2.15	ParamPresenceMask	833
8.853.2.16	bas_almanac_sv_msk	833
8.853.2.17	bas_ephemeris_sv_msk	833
8.853.2.18	bas_health_sv_msk	833
8.853.2.19	bas_visible_sv_msk	833
8.853.2.20	Time_uncert_ms	833
8.853.2.21	TimeStmp_gps_week	833
8.853.2.22	TimeStmp_tow_ms	833
8.853.2.23	ValidMask	833
8.853.2.24	VerticalUncertainty	833
8.853.2.25	xtra_start_gps_minutes	834
8.853.2.26	xtra_start_gps_week	834
8.853.2.27	xtra_valid_duration_hours	834
8.854	unpack_qmi_t Struct Reference	834
8.854.1	Detailed Description	834
8.854.2	Field Documentation	834
8.854.2.1	msgid	834
8.854.2.2	type	834
8.854.2.3	xid	834
8.855	unpack_qos_BindDataPort_t Struct Reference	834
8.855.1	Detailed Description	834
8.855.2	Field Documentation	834
8.855.2.1	ParamPresenceMask	834
8.856	unpack_qos_dataRate_t Struct Reference	835
8.856.1	Detailed Description	835
8.856.2	Field Documentation	835
8.856.2.1	dataRateMax	835
8.856.2.2	guaranteedRate	835
8.857	unpack_qos_IPv4Addr_t Struct Reference	835
8.857.1	Detailed Description	835
8.857.2	Field Documentation	835
8.857.2.1	addr	835
8.857.2.2	subnetMask	835
8.858	unpack_qos_IPv6Addr_t Struct Reference	835
8.858.1	Detailed Description	836
8.858.2	Field Documentation	836
8.858.2.1	addr	836
8.858.2.2	prefixLen	836

8.859unpack_qos_IPv6TrafCls_t Struct Reference	836
8.859.1 Detailed Description	836
8.859.2 Field Documentation	836
8.859.2.1 mask	836
8.859.2.2 val	836
8.860unpack_qos_pktErrRate_t Struct Reference	836
8.860.1 Detailed Description	837
8.860.2 Field Documentation	837
8.860.2.1 exponent	837
8.860.2.2 multiplier	837
8.861unpack_qos_Port_t Struct Reference	837
8.861.1 Detailed Description	837
8.861.2 Field Documentation	837
8.861.2.1 port	837
8.861.2.2 range	837
8.862unpack_qos_QosFlowInfo_t Struct Reference	837
8.862.1 Detailed Description	838
8.862.2 Field Documentation	838
8.862.2.1 BearerID	838
8.862.2.2 is_RxQFlowGranted_Available	838
8.862.2.3 is_TxQFlowGranted_Available	838
8.862.2.4 NumRxFilters	839
8.862.2.5 NumTxFilters	839
8.862.2.6 QFlowState	839
8.862.2.7 RxQFilter	839
8.862.2.8 RxQFlowGranted	839
8.862.2.9 TxQFilter	839
8.862.2.10TxQFlowGranted	839
8.863unpack_qos_QosFlowInfoState_t Struct Reference	839
8.863.1 Detailed Description	839
8.863.2 Field Documentation	839
8.863.2.1 id	839
8.863.2.2 isNewFlow	839
8.863.2.3 state	839
8.864unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference	839
8.864.1 Detailed Description	840
8.864.2 Field Documentation	840
8.864.2.1 NWQoSStatus	840
8.864.2.2 ParamPresenceMask	840
8.865unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference	840

8.865.1 Detailed Description	840
8.865.2 Field Documentation	841
8.865.2.1 ambr_dl	841
8.865.2.2 ambr_dl_ext	841
8.865.2.3 ambr_dl_ext2	841
8.865.2.4 ambr_ul	841
8.865.2.5 ambr_ul_ext	841
8.865.2.6 ambr_ul_ext2	841
8.865.2.7 apnId	841
8.865.2.8 ParamPresenceMask	841
8.866unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference	842
8.866.1 Detailed Description	842
8.866.2 Field Documentation	843
8.866.2.1 apnId	843
8.866.2.2 numQosFlow	843
8.866.2.3 ParamPresenceMask	843
8.866.2.4 qosFlow	843
8.866.2.5 total_rx_bytes	843
8.866.2.6 total_rx_pkt	843
8.866.2.7 total_tx_bytes	843
8.866.2.8 total_tx_bytes_drp	843
8.866.2.9 total_tx_pkt	843
8.866.2.10total_tx_pkt_drp	843
8.867unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference	843
8.867.1 Detailed Description	843
8.867.2 Field Documentation	844
8.867.2.1 NumFlows	844
8.867.2.2 ParamPresenceMask	844
8.867.2.3 QosFlowInfo	844
8.868unpack_qos_SLQSSetQosEventCallback_t Struct Reference	844
8.868.1 Detailed Description	844
8.868.2 Field Documentation	844
8.868.2.1 ParamPresenceMask	844
8.869unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference	844
8.869.1 Detailed Description	844
8.869.2 Field Documentation	845
8.869.2.1 ParamPresenceMask	845
8.869.2.2 status	845
8.870unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference	845
8.870.1 Detailed Description	845

8.870.2 Field Documentation	845
8.870.2.1 event	845
8.870.2.2 ParamPresenceMask	845
8.871unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference	845
8.871.1 Detailed Description	846
8.871.2 Field Documentation	847
8.871.2.1 event	847
8.871.2.2 id	847
8.871.2.3 ParamPresenceMask	847
8.871.2.4 reason	847
8.871.2.5 status	847
8.872unpack_qos_swiQosFilter_t Struct Reference	847
8.872.1 Detailed Description	848
8.872.2 Field Documentation	849
8.872.2.1 EspSpi	849
8.872.2.2 Id	849
8.872.2.3 index	849
8.872.2.4 IPv4DstAddr	849
8.872.2.5 IPv4SrcAddr	849
8.872.2.6 IPv4Tos	849
8.872.2.7 IPv6DstAddr	849
8.872.2.8 IPv6Label	849
8.872.2.9 IPv6SrcAddr	849
8.872.2.10IPv6TrafCls	849
8.872.2.11is_EspSpi_Available	849
8.872.2.12s_Id_Available	849
8.872.2.13s_IPv4DstAddr_Available	849
8.872.2.14s_IPv4SrcAddr_Available	849
8.872.2.15s_IPv4Tos_Available	849
8.872.2.16s_IPv6DstAddr_Available	849
8.872.2.17s_IPv6Label_Available	849
8.872.2.18s_IPv6SrcAddr_Available	849
8.872.2.19s_IPv6TrafCls_Available	850
8.872.2.20s_NxtHdrProto_Available	850
8.872.2.21is_Precedence_Available	850
8.872.2.22s_TCPDstPort_Available	850
8.872.2.23s_TCPSrcPort_Available	850
8.872.2.24s_TranDstPort_Available	850
8.872.2.25s_TranSrcPort_Available	850
8.872.2.26s_UDPDstPort_Available	850

8.872.2.27s_UDPSrcPort_Available	850
8.872.2.28NxtHdrProto	850
8.872.2.29Precedence	850
8.872.2.30TCPDstPort	850
8.872.2.31TCPSrcPort	850
8.872.2.32TranDstPort	850
8.872.2.33TranSrcPort	850
8.872.2.34UDPDstPort	850
8.872.2.35UDPSrcPort	850
8.872.2.36version	850
8.873unpack_qos_swiQosFlow_t Struct Reference	850
8.873.1 Detailed Description	851
8.873.2 Field Documentation	853
8.873.2.1 DataRate	853
8.873.2.2 index	853
8.873.2.3 is_DataRate_Available	853
8.873.2.4 is_Jitter_Available	853
8.873.2.5 is_Latency_Available	853
8.873.2.6 is_LteQci_Available	853
8.873.2.7 is_MaxAllowedPktSz_Available	853
8.873.2.8 is_MinPolicedPktSz_Available	853
8.873.2.9 is_PktErrRate_Available	853
8.873.2.10s_ProfileId3GPP2_Available	853
8.873.2.11is_TokenBucket_Available	853
8.873.2.12s_TrafficClass_Available	853
8.873.2.13s_val_3GPP2Pri_Available	853
8.873.2.14s_val_3GPPImCn_Available	853
8.873.2.15s_val_3GPPResResidualBER_Available	854
8.873.2.16s_val_3GPPSigInd_Available	854
8.873.2.17s_val_3GPPTraHdlPri_Available	854
8.873.2.18Jitter	854
8.873.2.19Latency	854
8.873.2.20LteQci	854
8.873.2.21MaxAllowedPktSz	854
8.873.2.22MinPolicedPktSz	854
8.873.2.23PktErrRate	854
8.873.2.24ProfileId3GPP2	854
8.873.2.25TokenBucket	854
8.873.2.26TrafficClass	854
8.873.2.27val_3GPP2Pri	854

8.873.2.28val_3GPPImCn	854
8.873.2.29val_3GPPResResidualBER	854
8.873.2.30val_3GPPSigInd	854
8.873.2.31val_3GPPTraHdlPri	854
8.874unpack_qos_tokenBucket_t Struct Reference	854
8.874.1 Detailed Description	854
8.874.2 Field Documentation	855
8.874.2.1 bucketSz	855
8.874.2.2 peakRate	855
8.874.2.3 tokenRate	855
8.875unpack_qos_Tos_t Struct Reference	855
8.875.1 Detailed Description	855
8.875.2 Field Documentation	855
8.875.2.1 mask	855
8.875.2.2 val	855
8.876unpack_QosFlowStat_t Struct Reference	855
8.876.1 Detailed Description	855
8.876.2 Field Documentation	856
8.876.2.1 bearerId	856
8.876.2.2 tx_bytes	856
8.876.2.3 tx_bytes_drp	856
8.876.2.4 tx_pkt	856
8.876.2.5 tx_pkt_drp	856
8.877unpack_result_t Struct Reference	856
8.877.1 Detailed Description	856
8.877.2 Field Documentation	856
8.877.2.1 ParamPresenceMask	856
8.877.2.2 Tlvresult	856
8.878unpack_rms_GetSMSWake_t Struct Reference	857
8.878.1 Detailed Description	857
8.878.2 Field Documentation	857
8.878.2.1 enabled	857
8.878.2.2 ParamPresenceMask	857
8.878.2.3 wake_mask	857
8.879unpack_rms_SetSMSWake_t Struct Reference	857
8.879.1 Detailed Description	857
8.879.2 Field Documentation	858
8.879.2.1 ParamPresenceMask	858
8.880unpack_RMTransferStatistics_ind_t Struct Reference	858
8.880.1 Detailed Description	858

8.880.2 Field Documentation	858
8.880.2.1 ParamPresenceMask	858
8.880.2.2 RxDropConutTlv	858
8.880.2.3 RxOkByteCountTlv	859
8.880.2.4 RxOkConutTlv	859
8.880.2.5 TxDropConutTlv	859
8.880.2.6 TxOkByteCountTlv	859
8.880.2.7 TxOkConutTlv	859
8.881unpack_sar_SLQSGetRfSarState_t Struct Reference	859
8.881.1 Detailed Description	859
8.881.2 Field Documentation	859
8.881.2.1 ParamPresenceMask	859
8.881.2.2 pSarRFState	859
8.882unpack_sms_GetSMSCAddress_t Struct Reference	859
8.882.1 Detailed Description	860
8.882.2 Field Documentation	860
8.882.2.1 addressSize	860
8.882.2.2 ParamPresenceMask	860
8.882.2.3 pSMSCAddress	860
8.882.2.4 pSMSCType	860
8.882.2.5 typeSize	860
8.883unpack_sms_SaveSMS_t Struct Reference	860
8.883.1 Detailed Description	861
8.883.2 Field Documentation	861
8.883.2.1 ParamPresenceMask	861
8.883.2.2 pMessageIndex	861
8.884unpack_sms_SendSMS_t Struct Reference	861
8.884.1 Detailed Description	861
8.884.2 Field Documentation	861
8.884.2.1 messageFailureCode	861
8.884.2.2 messageID	861
8.884.2.3 ParamPresenceMask	861
8.885unpack_sms_SetNewSMSCallback_ind_t Struct Reference	862
8.885.1 Detailed Description	862
8.885.2 Field Documentation	863
8.885.2.1 ETWSPLMNTlv	863
8.885.2.2 ETWSTlv	863
8.885.2.3 IMSTlv	863
8.885.2.4 MMTlv	863
8.885.2.5 NewMMTlv	863

8.885.2.6 ParamPresenceMask	863
8.885.2.7 SMSCTlv	863
8.885.2.8 TRMessageTlv	863
8.886unpack_sms_SetNewSMSCallback_t Struct Reference	863
8.886.1 Detailed Description	863
8.886.2 Field Documentation	863
8.886.2.1 ParamPresenceMask	863
8.887unpack_sms_SetSMSCAddress_t Struct Reference	863
8.887.1 Detailed Description	863
8.887.2 Field Documentation	864
8.887.2.1 ParamPresenceMask	864
8.888unpack_sms_SLQSDDeleteSMS_t Struct Reference	864
8.888.1 Detailed Description	864
8.888.2 Field Documentation	864
8.888.2.1 ParamPresenceMask	864
8.889unpack_sms_SLQSGetIndicationRegister_t Struct Reference	864
8.889.1 Detailed Description	864
8.889.2 Field Documentation	865
8.889.2.1 ParamPresenceMask	865
8.889.2.2 pGetIndicationRegInfo	865
8.890unpack_sms_SLQSGetMessageWaiting_t Struct Reference	865
8.890.1 Detailed Description	865
8.890.2 Field Documentation	865
8.890.2.1 ParamPresenceMask	865
8.890.2.2 pGetMsgWaitingInfoResp	865
8.891unpack_sms_SLQSGetSMS_t Struct Reference	865
8.891.1 Detailed Description	865
8.891.2 Field Documentation	866
8.891.2.1 message	866
8.891.2.2 messageFormat	866
8.891.2.3 messageSize	866
8.891.2.4 messageTag	866
8.891.2.5 ParamPresenceMask	866
8.892unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference	866
8.892.1 Detailed Description	867
8.892.2 Field Documentation	867
8.892.2.1 ParamPresenceMask	867
8.892.2.2 pBroadcastConfig	867
8.892.2.3 pCDMABroadcastConfig	867
8.893unpack_sms_SLQSGetSMSList_t Struct Reference	867

8.893.1 Detailed Description	867
8.893.2 Field Documentation	868
8.893.2.1 messageList	868
8.893.2.2 messageListSize	868
8.893.2.3 ParamPresenceMask	868
8.894unpack_sms_SLQSGetTransLayerInfo_t Struct Reference	868
8.894.1 Detailed Description	868
8.894.2 Field Documentation	868
8.894.2.1 ParamPresenceMask	868
8.894.2.2 pGetTransLayerInfo	868
8.895unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference	868
8.895.1 Detailed Description	869
8.895.2 Field Documentation	869
8.895.2.1 ParamPresenceMask	869
8.895.2.2 pGetTransNWRegInfo	869
8.896unpack_sms_SLQSModifySMSStatus_t Struct Reference	869
8.896.1 Detailed Description	869
8.896.2 Field Documentation	869
8.896.2.1 ParamPresenceMask	869
8.897unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference	869
8.897.1 Detailed Description	870
8.897.2 Field Documentation	870
8.897.2.1 NWRegStat	870
8.897.2.2 ParamPresenceMask	870
8.898unpack_sms_SLQSSendAsyncSMS_t Struct Reference	870
8.898.1 Detailed Description	870
8.898.2 Field Documentation	870
8.898.2.1 ParamPresenceMask	870
8.899unpack_sms_SLQSSetIndicationRegister_t Struct Reference	870
8.899.1 Detailed Description	871
8.899.2 Field Documentation	871
8.899.2.1 ParamPresenceMask	871
8.900unpack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference	871
8.900.1 Detailed Description	871
8.900.2 Field Documentation	871
8.900.2.1 ParamPresenceMask	871
8.901unpack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference	871
8.901.1 Detailed Description	871
8.901.2 Field Documentation	872
8.901.2.1 ParamPresenceMask	872

8.902unpack_sms_SLQSSetSmsStorage_t Struct Reference	872
8.902.1 Detailed Description	872
8.902.2 Field Documentation	872
8.902.2.1 ParamPresenceMask	872
8.903unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference	872
8.903.1 Detailed Description	872
8.903.2 Field Documentation	872
8.903.2.1 ParamPresenceMask	873
8.903.2.2 pMaxStorageSizeResp	873
8.904unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference	873
8.904.1 Detailed Description	873
8.904.2 Field Documentation	873
8.904.2.1 ParamPresenceMask	873
8.904.2.2 pMessageProtocol	873
8.905unpack_sms_SLQSSmsSetRoutes_t Struct Reference	873
8.905.1 Detailed Description	873
8.905.2 Field Documentation	873
8.905.2.1 ParamPresenceMask	873
8.906unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference	874
8.906.1 Detailed Description	874
8.906.2 Field Documentation	874
8.906.2.1 ParamPresenceMask	874
8.906.2.2 pSmsStorage	874
8.907unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference	874
8.907.1 Detailed Description	874
8.907.2 Field Documentation	875
8.907.2.1 ParamPresenceMask	875
8.907.2.2 pTransLayerInfo	875
8.907.2.3 regInd	875
8.908unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t Struct Reference	875
8.908.1 Detailed Description	875
8.908.2 Field Documentation	877
8.908.2.1 alphaIDLen	877
8.908.2.2 causeCode	877
8.908.2.3 errorClass	877
8.908.2.4 messageID	877
8.908.2.5 msgDelFailureCause	877
8.908.2.6 msgDelFailureType	877
8.908.2.7 pAlphaID	877
8.908.2.8 ParamPresenceMask	877

8.908.2.9 RPCause	877
8.908.2.10 sendStatus	877
8.908.2.11 TPCause	877
8.908.2.12 userData	877
8.909unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference	877
8.909.1 Detailed Description	877
8.909.2 Field Documentation	878
8.909.2.1 messageMode	878
8.909.2.2 ParamPresenceMask	878
8.909.2.3 storageType	878
8.910unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t Struct Reference	878
8.910.1 Detailed Description	878
8.910.2 Field Documentation	878
8.910.2.1 msgWaitInfo	878
8.910.2.2 numInstances	878
8.910.2.3 ParamPresenceMask	878
8.911unpack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference	878
8.911.1 Detailed Description	879
8.911.2 Field Documentation	879
8.911.2.1 CwtMute	879
8.911.2.2 EarMute	879
8.911.2.3 Generator	879
8.911.2.4 MicMute	880
8.911.2.5 ParamPresenceMask	880
8.911.2.6 Profile	880
8.911.2.7 Volume	880
8.912unpack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference	880
8.912.1 Detailed Description	880
8.912.2 Field Documentation	880
8.912.2.1 Level	880
8.912.2.2 ParamPresenceMask	880
8.913unpack_swiaudio_SLQSGetM2MAVMute_t Struct Reference	880
8.913.1 Detailed Description	880
8.913.2 Field Documentation	881
8.913.2.1 CwtMute	881
8.913.2.2 EarMute	881
8.913.2.3 MicMute	881
8.913.2.4 ParamPresenceMask	881
8.914unpack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference	881
8.914.1 Detailed Description	881

8.914.2 Field Documentation	882
8.914.2.1 ParamPresenceMask	882
8.914.2.2 Value	882
8.915unpack_swiaavms_SLQSAVMSEventReportInd_t Struct Reference	882
8.915.1 Detailed Description	882
8.915.2 Field Documentation	883
8.915.2.1 ParamPresenceMask	883
8.915.2.2 pBinaryUpdateSessionInfo	883
8.915.2.3 pConfig	883
8.915.2.4 pConnectionRequest	883
8.915.2.5 pDataSessionStatus	883
8.915.2.6 pHTTPStatus	883
8.915.2.7 pNotification	883
8.915.2.8 pPackageID	883
8.915.2.9 pRegStatus	883
8.915.2.10pSessionType	883
8.915.2.11pWAMSParaChanged	883
8.915.2.12resultcode	883
8.916unpack_swiaavms_SLQSAVMSSettings_t Struct Reference	883
8.916.1 Detailed Description	884
8.916.2 Field Documentation	885
8.916.2.1 AutoConnect	885
8.916.2.2 AutoReboot	885
8.916.2.3 FwAutoSDM	885
8.916.2.4 FwPromptdownload	886
8.916.2.5 FwPromptUpdate	886
8.916.2.6 OMADMEEnabled	886
8.916.2.7 pAPNInfo	886
8.916.2.8 ParamPresenceMask	886
8.916.2.9 pConnectionRetryTimers	886
8.916.2.10pNotificationStore	886
8.916.2.11pPeroidsInfo	886
8.916.2.12pPollingTimer	886
8.916.2.13resultcode	886
8.917unpack_swiaavms_SLQSAVMSSettings_v2_t Struct Reference	886
8.917.1 Detailed Description	886
8.917.2 Field Documentation	888
8.917.2.1 AutoConnect	888
8.917.2.2 FwAutoSDM	888
8.917.2.3 FwPromptdownload	888

8.917.2.4 FwPromptUpdate	888
8.917.2.5 OMADMEEnabled	888
8.917.2.6 pAPNInfo	888
8.917.2.7 ParamPresenceMask	888
8.917.2.8 pAutoReboot	888
8.917.2.9 pConnectionRetryTimers	888
8.917.2.10pNotificationStore	888
8.917.2.11pPeroidsInfo	888
8.917.2.12pPollingTimer	888
8.917.2.13resultcode	888
8.918unpack_swiaavms_SLQSAVMSSendSelection_t Struct Reference	888
8.918.1 Detailed Description	888
8.918.2 Field Documentation	889
8.918.2.1 ParamPresenceMask	889
8.918.2.2 resultcode	889
8.919unpack_swiaavms_SLQSAVMSSessionGetInfo_t Struct Reference	889
8.919.1 Detailed Description	889
8.919.2 Field Documentation	889
8.919.2.1 ParamPresenceMask	889
8.919.2.2 pBinaryUpdateSessionInfo	889
8.919.2.3 pConfig	890
8.919.2.4 pNotification	890
8.919.2.5 pPackageID	890
8.919.2.6 resultcode	890
8.920unpack_swiaavms_SLQSAvmsSetEventReport_t Struct Reference	890
8.920.1 Detailed Description	890
8.920.2 Field Documentation	890
8.920.2.1 ParamPresenceMask	890
8.920.2.2 resultcode	890
8.921unpack_swiaavms_SLQSAVMSSetSettings_t Struct Reference	890
8.921.1 Detailed Description	890
8.921.2 Field Documentation	890
8.921.2.1 ParamPresenceMask	890
8.921.2.2 resultcode	891
8.922unpack_swiaavms_SLQSAVMSSetSettings_v2_t Struct Reference	891
8.922.1 Detailed Description	891
8.922.2 Field Documentation	891
8.922.2.1 ParamPresenceMask	891
8.922.2.2 resultcode	891
8.923unpack_swiaavms_SLQSAVMSStartSession_t Struct Reference	891

8.923.1 Detailed Description	891
8.923.2 Field Documentation	891
8.923.2.1 ParamPresenceMask	891
8.923.2.2 resultcode	892
8.923.2.3 sessionResponse	892
8.924unpack_swiaVms_SLQSAVMSSStopSession_t Struct Reference	892
8.924.1 Detailed Description	892
8.924.2 Field Documentation	892
8.924.2.1 ParamPresenceMask	892
8.924.2.2 resultcode	892
8.925unpack_swidms_SLQSSwiDmsGetHWWatchdog_t Struct Reference	892
8.925.1 Detailed Description	892
8.925.2 Field Documentation	892
8.925.2.1 ParamPresenceMask	892
8.925.2.2 pHWWatchdog	893
8.926unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference	893
8.926.1 Detailed Description	893
8.926.2 Field Documentation	893
8.926.2.1 ParamPresenceMask	893
8.926.2.2 pEhrpdMTUSize	893
8.926.2.3 pHrpdMTUSize	893
8.926.2.4 pMTUSize3gpp	893
8.926.2.5 pUsbMTUSize	893
8.927unpack_swidms_SLQSSwiDmsGetSecureInfo_t Struct Reference	893
8.927.1 Detailed Description	894
8.927.2 Field Documentation	894
8.927.2.1 jtagAccessAllowed	894
8.927.2.2 memoryDumpAllowed	894
8.927.2.3 ParamPresenceMask	894
8.927.2.4 secureBootEnabled	894
8.927.2.5 TlvResult	894
8.928unpack_swidms_SLQSSwiDmsGetUsbComp_t Struct Reference	894
8.928.1 Detailed Description	895
8.928.2 Field Documentation	895
8.928.2.1 ParamPresenceMask	895
8.928.2.2 pInterfaceCfg	895
8.928.2.3 pSupportedBitmasks	895
8.929unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference	895
8.929.1 Detailed Description	895
8.929.2 Field Documentation	895

8.929.2.1 ParamPresenceMask	895
8.929.2.2 usbNetNum	895
8.930unpack_swidms_SLQSSwiDmsSetHWWatchdog_t Struct Reference	896
8.930.1 Detailed Description	896
8.930.2 Field Documentation	896
8.930.2.1 ParamPresenceMask	896
8.930.2.2 Tlvresult	896
8.931unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference	896
8.931.1 Detailed Description	896
8.931.2 Field Documentation	896
8.931.2.1 ParamPresenceMask	896
8.931.2.2 Tlvresult	897
8.932unpack_swidms_SLQSSwiDmsSetUsbComp_t Struct Reference	897
8.932.1 Detailed Description	897
8.932.2 Field Documentation	897
8.932.2.1 ParamPresenceMask	897
8.932.2.2 Tlvresult	897
8.933unpack_swiloc_SwiLocGetAutoStart_t Struct Reference	897
8.933.1 Detailed Description	897
8.933.2 Field Documentation	899
8.933.2.1 fix_rate	899
8.933.2.2 fix_rate_reported	899
8.933.2.3 fix_type	899
8.933.2.4 fix_type_reported	899
8.933.2.5 function	899
8.933.2.6 function_reported	899
8.933.2.7 max_dist	899
8.933.2.8 max_dist_reported	899
8.933.2.9 max_time	899
8.933.2.10max_time_reported	899
8.933.2.11ParamPresenceMask	899
8.934unpack_swioama_SLQSOMADMAAlertCallback_ind_t Struct Reference	899
8.934.1 Detailed Description	899
8.934.2 Field Documentation	900
8.934.2.1 eventType	900
8.934.2.2 ParamPresenceMask	900
8.934.2.3 SessionInfoConfig	900
8.934.2.4 SessionInfoFota	900
8.934.2.5 SessionInfoNotification	900
8.935unpack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference	900

8.935.1 Detailed Description	900
8.935.2 Field Documentation	902
8.935.2.1 Date	902
8.935.2.2 DateLength	903
8.935.2.3 ParamPresenceMask	903
8.935.2.4 PkgDescLength	903
8.935.2.5 PkgDescription	903
8.935.2.6 PkgName	903
8.935.2.7 PkgNameLength	903
8.935.2.8 RetryCount	903
8.935.2.9 SessionState	903
8.935.2.10 SessionType	903
8.935.2.11 Severity	903
8.935.2.12 Source	903
8.935.2.13 SourceLength	903
8.935.2.14 Status	903
8.935.2.15 Time	903
8.935.2.16 TimeLength	903
8.935.2.17 UpdateCompleteStatus	903
8.936unpack_swioama_SLQSOMADMGetSessionInfoExt_t Struct Reference	903
8.936.1 Detailed Description	904
8.936.2 Field Documentation	906
8.936.2.1 fumoResultCode	906
8.936.2.2 fumoState	906
8.936.2.3 hfaMaxRetry	906
8.936.2.4 hfaRetryIndex	906
8.936.2.5 hfaRetryInterval	906
8.936.2.6 hfaStatus	906
8.936.2.7 ParamPresenceMask	906
8.936.2.8 pkgDate	906
8.936.2.9 pkgDateLength	906
8.936.2.10 pkgDesc	906
8.936.2.11 pkgDescLength	906
8.936.2.12 pkgInstallTime	906
8.936.2.13 pkgInstallTimeLength	906
8.936.2.14 pkgName	906
8.936.2.15 pkgNameLength	906
8.936.2.16 pkgSize	906
8.936.2.17 pkgVendorName	906
8.936.2.18 pkgVendorNameLength	906

8.936.2.1	pkgVersionName	906
8.936.2.2	pkgVersionNameLength	906
8.936.2.2.1	sessionState	906
8.936.2.2.2	status	906
8.937	unpack_swioama_SLQSOMADMGetSettings_t Struct Reference	907
8.937.1	Detailed Description	907
8.937.2	Field Documentation	908
8.937.2.1	Autosdm	908
8.937.2.2	FOTAdownload	908
8.937.2.3	FOTAUpdate	908
8.937.2.4	FwAutoCheck	908
8.937.2.5	OMADMEEnabled	908
8.937.2.6	ParamPresenceMask	908
8.938	unpack_swioama_SLQSOMADMStartSession_t Struct Reference	908
8.938.1	Detailed Description	908
8.938.2	Field Documentation	909
8.938.2.1	FwAvailability	909
8.938.2.2	ParamPresenceMask	909
8.939	unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference	909
8.939.1	Detailed Description	909
8.939.2	Field Documentation	909
8.939.2.1	ParamPresenceMask	909
8.939.2.2	Tlvresult	909
8.940	unpack_tmd_SLQSTmdGetMitigationDevList_t Struct Reference	909
8.940.1	Detailed Description	909
8.940.2	Field Documentation	910
8.940.2.1	MitigationDevList	910
8.940.2.2	MitigationDevListLen	910
8.940.2.3	ParamPresenceMask	910
8.940.2.4	Tlvresult	910
8.941	unpack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference	910
8.941.1	Detailed Description	910
8.941.2	Field Documentation	911
8.941.2.1	CurrentmitigationLvl	911
8.941.2.2	ParamPresenceMask	911
8.941.2.3	ReqMitigationLvl	911
8.941.2.4	Tlvresult	911
8.942	unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t Struct Reference	911
8.942.1	Detailed Description	911
8.942.2	Field Documentation	911

8.942.2.1 deviceId	911
8.942.2.2 deviceIdLen	911
8.942.2.3 lvi	911
8.942.2.4 ParamPresenceMask	911
8.943unpack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference	911
8.943.1 Detailed Description	912
8.943.2 Field Documentation	912
8.943.2.1 ParamPresenceMask	912
8.943.2.2 Tlvresult	912
8.944unpack_uim_ChangePin_t Struct Reference	912
8.944.1 Detailed Description	912
8.944.2 Field Documentation	913
8.944.2.1 ParamPresenceMask	913
8.944.2.2 pEncryptedPIN1	913
8.944.2.3 pIndicationToken	913
8.944.2.4 pRemainingRetries	913
8.944.2.5 Tlvresult	913
8.945unpack_uim_GetCardStatus_t Struct Reference	913
8.945.1 Detailed Description	913
8.945.2 Field Documentation	913
8.945.2.1 ParamPresenceMask	913
8.945.2.2 pCardStatus	913
8.945.2.3 pHotSwapStatus	913
8.945.2.4 Tlvresult	914
8.946unpack_uim_GetCardStatusV2_t Struct Reference	914
8.946.1 Detailed Description	914
8.946.2 Field Documentation	914
8.946.2.1 ParamPresenceMask	914
8.946.2.2 pCardStatus	914
8.946.2.3 pHotSwapStatus	914
8.946.2.4 pSimBusyStatus	914
8.946.2.5 pValidCardStatus	914
8.946.2.6 Tlvresult	914
8.947unpack_uim_ReadTransparent_t Struct Reference	915
8.947.1 Detailed Description	915
8.947.2 Field Documentation	915
8.947.2.1 ParamPresenceMask	915
8.947.2.2 pCardResult	915
8.947.2.3 pEncryptedData	915
8.947.2.4 pIndicationToken	915

8.947.2.5 pReadResult	915
8.947.2.6 Tlvresult	915
8.948unpack_uim_SetPinProtection_t Struct Reference	916
8.948.1 Detailed Description	916
8.948.2 Field Documentation	916
8.948.2.1 ParamPresenceMask	916
8.948.2.2 pEncryptedPIN1	916
8.948.2.3 pIndicationToken	916
8.948.2.4 pRemainingRetries	916
8.948.2.5 Tlvresult	916
8.949unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference	916
8.949.1 Detailed Description	917
8.949.2 Field Documentation	917
8.949.2.1 bNumberOfPhySlots	917
8.949.2.2 ParamPresenceMask	917
8.949.2.3 slotsstatusChange	917
8.950unpack_uim_SLQSUIMAuthenticate_t Struct Reference	917
8.950.1 Detailed Description	917
8.950.2 Field Documentation	918
8.950.2.1 ParamPresenceMask	918
8.950.2.2 pAuthenticateResult	918
8.950.2.3 pCardResult	918
8.950.2.4 pIndicationToken	918
8.951unpack_uim_SLQSUIMDepersonalization_t Struct Reference	918
8.951.1 Detailed Description	918
8.951.2 Field Documentation	918
8.951.2.1 ParamPresenceMask	918
8.951.2.2 pRemainingRetries	918
8.952unpack_uim_SLQSUIMEventRegister_t Struct Reference	918
8.952.1 Detailed Description	918
8.952.2 Field Documentation	919
8.952.2.1 eventMask	919
8.952.2.2 ParamPresenceMask	919
8.953unpack_uim_SLQSUIMGetConfiguration_t Struct Reference	919
8.953.1 Detailed Description	919
8.953.2 Field Documentation	920
8.953.2.1 ParamPresenceMask	920
8.953.2.2 pAutoSelection	920
8.953.2.3 pHaltSubscription	920
8.953.2.4 pPersonalizationStatus	920

8.954unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference	920
8.954.1 Detailed Description	920
8.954.2 Field Documentation	921
8.954.2.1 ParamPresenceMask	921
8.954.2.2 pCardResult	921
8.954.2.3 pFileAttributes	921
8.954.2.4 pIndicationToken	921
8.955unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference	921
8.955.1 Detailed Description	921
8.955.2 Field Documentation	921
8.955.2.1 ParamPresenceMask	921
8.955.2.2 pNumberOfPhySlot	921
8.955.2.3 pUimSlotsStatus	921
8.956unpack_uim_SLQSUIMGetSlotsStatusV2_t Struct Reference	921
8.956.1 Detailed Description	922
8.956.2 Field Documentation	922
8.956.2.1 ParamPresenceMask	922
8.956.2.2 pGetSlotsInfoTlv	922
8.956.2.3 pGetSlotsStatusTlv	922
8.957unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference	922
8.957.1 Detailed Description	922
8.957.2 Field Documentation	923
8.957.2.1 ParamPresenceMask	923
8.957.2.2 refreshEvent	923
8.957.2.3 TlvPresent	923
8.958unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference	923
8.958.1 Detailed Description	923
8.958.2 Field Documentation	923
8.958.2.1 ParamPresenceMask	923
8.958.2.2 pRefreshEvent	923
8.959unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference	923
8.959.1 Detailed Description	923
8.959.2 Field Documentation	924
8.959.2.1 ParamPresenceMask	924
8.959.2.2 pCardStatus	924
8.960unpack_uim_UnblockPin_t Struct Reference	924
8.960.1 Detailed Description	924
8.960.2 Field Documentation	924
8.960.2.1 ParamPresenceMask	924
8.960.2.2 pEncryptedPIN1	925

8.960.2.3 pIndicationToken	925
8.960.2.4 pRemainingRetries	925
8.960.2.5 Tlvresult	925
8.961unpack_uim_UnblockPinV2_t Struct Reference	925
8.961.1 Detailed Description	925
8.961.2 Field Documentation	925
8.961.2.1 ParamPresenceMask	925
8.961.2.2 pCardResult	925
8.961.2.3 pEncryptedPIN1	926
8.961.2.4 pIndicationToken	926
8.961.2.5 pRemainingRetries	926
8.961.2.6 Tlvresult	926
8.962unpack_uim_VerifyPin_t Struct Reference	926
8.962.1 Detailed Description	926
8.962.2 Field Documentation	926
8.962.2.1 ParamPresenceMask	926
8.962.2.2 pEncryptedPIN1	926
8.962.2.3 pIndicationToken	926
8.962.2.4 pRemainingRetries	926
8.962.2.5 Tlvresult	926
8.963unpack_voice_allCallStatusCallback_ind_t Struct Reference	927
8.963.1 Detailed Description	927
8.963.2 Field Documentation	928
8.963.2.1 arrCallInfomation	928
8.963.2.2 ParamPresenceMask	928
8.963.2.3 pArrAlertingPattern	928
8.963.2.4 pArrAlertingType	928
8.963.2.5 pArrAlphaID	928
8.963.2.6 pArrCalledPartyNum	928
8.963.2.7 pArrCallEndReason	928
8.963.2.8 pArrConnectPartyNum	928
8.963.2.9 pArrDiagInfo	928
8.963.2.10pArrRedirPartyNum	928
8.963.2.11pArrRemotePartyName	928
8.963.2.12pArrRemotePartyNum	928
8.963.2.13pArrSvcOption	928
8.964unpack_voice_DTMFEventCallback_ind_t Struct Reference	929
8.964.1 Detailed Description	929
8.964.2 Field Documentation	929
8.964.2.1 DTMFInformation	929

8.964.2.2 ParamPresenceMask	929
8.964.2.3 pOffLength	929
8.964.2.4 pOnLength	930
8.965unpack_voice_OTASPStatusCallback_ind_t Struct Reference	930
8.965.1 Detailed Description	930
8.965.2 Field Documentation	931
8.965.2.1 callID	931
8.965.2.2 OTASPStatus	931
8.965.2.3 ParamPresenceMask	931
8.966unpack_voice_SLQSOriginateUSSD_t Struct Reference	931
8.966.1 Detailed Description	931
8.966.2 Field Documentation	933
8.966.2.1 pAlphaIDInfo	933
8.966.2.2 ParamPresenceMask	933
8.966.2.3 pCallID	933
8.966.2.4 pCcResultType	933
8.966.2.5 pCCSuppsType	933
8.966.2.6 pfailureCause	933
8.966.2.7 pUSSDInfo	933
8.967unpack_voice_SLQSVoiceAnswerCall_t Struct Reference	933
8.967.1 Detailed Description	933
8.967.2 Field Documentation	934
8.967.2.1 ParamPresenceMask	934
8.967.2.2 pCallID	934
8.968unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference	934
8.968.1 Detailed Description	934
8.968.2 Field Documentation	934
8.968.2.1 ParamPresenceMask	934
8.968.2.2 pCallID	934
8.969unpack_voice_SLQSVoiceDialCall_t Struct Reference	934
8.969.1 Detailed Description	935
8.969.2 Field Documentation	935
8.969.2.1 pAlphaIDInfo	935
8.969.2.2 ParamPresenceMask	935
8.969.2.3 pCallID	935
8.969.2.4 pCCResultType	935
8.969.2.5 pCCSUPSType	935
8.970unpack_voice_SLQSVoiceEndCall_t Struct Reference	935
8.970.1 Detailed Description	936
8.970.2 Field Documentation	936

8.970.2.1 ParamPresenceMask	936
8.970.2.2 pCallId	936
8.971unpack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference	936
8.971.1 Detailed Description	936
8.971.2 Field Documentation	939
8.971.2.1 ParamPresenceMask	939
8.971.2.2 pArrAlertingPattern	939
8.971.2.3 pArrAlertingType	939
8.971.2.4 pArrAlphaID	939
8.971.2.5 pArrCalledPartyNum	939
8.971.2.6 pArrCallEndReason	939
8.971.2.7 pArrCallInfo	939
8.971.2.8 pArrConnectPartyNum	939
8.971.2.9 pArrDiagInfo	939
8.971.2.10pArrRedirPartyNum	939
8.971.2.11pArrRemotePartyName	939
8.971.2.12pArrRemotePartyNum	939
8.971.2.13pArrSvcOption	939
8.971.2.14pArrUUSInfo	939
8.971.2.15pOTASPStatus	939
8.971.2.16pVoicePrivacy	939
8.972unpack_voice_SLQSVoiceGetCallBarring_t Struct Reference	940
8.972.1 Detailed Description	940
8.972.2 Field Documentation	941
8.972.2.1 pAlphaIDInfo	941
8.972.2.2 ParamPresenceMask	941
8.972.2.3 pCallID	941
8.972.2.4 pCCResType	941
8.972.2.5 pCCSUPSType	941
8.972.2.6 pFailCause	941
8.972.2.7 pSvcClass	941
8.973unpack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference	941
8.973.1 Detailed Description	941
8.973.2 Field Documentation	942
8.973.2.1 pAlphaIDInfo	942
8.973.2.2 ParamPresenceMask	942
8.973.2.3 pCallID	942
8.973.2.4 pCCResType	942
8.973.2.5 pCCSUPSType	942
8.973.2.6 pFailCause	943

8.973.2.7 pGetCallFWExtInfo	943
8.973.2.8 pGetCallFWInfo	943
8.974unpack_voice_SLQSVoiceGetCallInfo_t Struct Reference	943
8.974.1 Detailed Description	943
8.974.2 Field Documentation	945
8.974.2.1 pAlertingPattern	945
8.974.2.2 pAlertType	945
8.974.2.3 pAlphaIDInfo	945
8.974.2.4 ParamPresenceMask	945
8.974.2.5 pCallInfo	945
8.974.2.6 pConnectNumInfo	945
8.974.2.7 pDiagInfo	945
8.974.2.8 pOTASPStatus	945
8.974.2.9 pRemotePartyName	945
8.974.2.10pRemotePartyNum	945
8.974.2.11pSrvOpt	946
8.974.2.12pUUSInfo	946
8.974.2.13pVoicePrivacy	946
8.975unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference	946
8.975.1 Detailed Description	946
8.975.2 Field Documentation	947
8.975.2.1 pAlphaIDInfo	947
8.975.2.2 ParamPresenceMask	947
8.975.2.3 pCallID	947
8.975.2.4 pCCResType	947
8.975.2.5 pCCSUPSType	947
8.975.2.6 pFailCause	947
8.975.2.7 pSvcClass	947
8.976unpack_voice_SLQSVoiceGetCLIP_t Struct Reference	947
8.976.1 Detailed Description	947
8.976.2 Field Documentation	948
8.976.2.1 pAlphaIDInfo	948
8.976.2.2 ParamPresenceMask	948
8.976.2.3 pCallID	948
8.976.2.4 pCCResType	948
8.976.2.5 pCCSUPSType	948
8.976.2.6 pCLIPResp	949
8.976.2.7 pFailCause	949
8.977unpack_voice_SLQSVoiceGetCLIR_t Struct Reference	949
8.977.1 Detailed Description	949

8.977.2 Field Documentation	950
8.977.2.1 pAlphaIDInfo	950
8.977.2.2 ParamPresenceMask	950
8.977.2.3 pCallID	950
8.977.2.4 pCCResType	950
8.977.2.5 pCCSUPSType	950
8.977.2.6 pCLIRResp	950
8.977.2.7 pFailCause	950
8.978unpack_voice_SLQSVoiceGetCNAP_t Struct Reference	950
8.978.1 Detailed Description	950
8.978.2 Field Documentation	951
8.978.2.1 pAlphaIDInfo	951
8.978.2.2 ParamPresenceMask	951
8.978.2.3 pCallID	951
8.978.2.4 pCCResType	951
8.978.2.5 pCCSUPSType	951
8.978.2.6 pCNAPResp	951
8.978.2.7 pFailCause	951
8.979unpack_voice_SLQSVoiceGetCOLP_t Struct Reference	952
8.979.1 Detailed Description	952
8.979.2 Field Documentation	953
8.979.2.1 pAlphaIDInfo	953
8.979.2.2 ParamPresenceMask	953
8.979.2.3 pCallID	953
8.979.2.4 pCCResType	953
8.979.2.5 pCCSUPSType	953
8.979.2.6 pCOLPResp	953
8.979.2.7 pFailCause	953
8.980unpack_voice_SLQSVoiceGetCOLR_t Struct Reference	953
8.980.1 Detailed Description	953
8.980.2 Field Documentation	954
8.980.2.1 pAlphaIDInfo	954
8.980.2.2 ParamPresenceMask	954
8.980.2.3 pCallID	954
8.980.2.4 pCCResType	954
8.980.2.5 pCCSUPSType	954
8.980.2.6 pCOLRResp	954
8.980.2.7 pFailCause	954
8.981unpack_voice_SLQSVoiceGetConfig_t Struct Reference	954
8.981.1 Detailed Description	955

8.981.2 Field Documentation	956
8.981.2.1 pAirTimerCnt	956
8.981.2.2 ParamPresenceMask	956
8.981.2.3 pAutoAnswerStat	956
8.981.2.4 pCurAMRConfig	956
8.981.2.5 pCurPrefVoiceSO	956
8.981.2.6 pCurrTTYMode	956
8.981.2.7 pCurVoiceDomainPref	956
8.981.2.8 pCurVoicePrivacyPref	956
8.981.2.9 pRoamTimerCnt	956
8.982unpack_voice_SLQSVoiceManageCalls_t Struct Reference	956
8.982.1 Detailed Description	957
8.982.2 Field Documentation	957
8.982.2.1 ParamPresenceMask	957
8.982.2.2 pFailCause	957
8.983unpack_voice_SLQSVoiceSendFlash_t Struct Reference	957
8.983.1 Detailed Description	957
8.983.2 Field Documentation	957
8.983.2.1 ParamPresenceMask	957
8.983.2.2 pCallID	958
8.984unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference	958
8.984.1 Detailed Description	958
8.984.2 Field Documentation	959
8.984.2.1 pAlphaIDInfo	959
8.984.2.2 ParamPresenceMask	959
8.984.2.3 pCallID	959
8.984.2.4 pCCResType	959
8.984.2.5 pCCSUPSType	959
8.984.2.6 pFailCause	959
8.985unpack_voice_SLQSVoiceSetConfig_t Struct Reference	959
8.985.1 Detailed Description	959
8.985.2 Field Documentation	960
8.985.2.1 pAirTimerStatus	960
8.985.2.2 ParamPresenceMask	960
8.985.2.3 pAutoAnsStatus	960
8.985.2.4 pPrefVoiceSOStatus	960
8.985.2.5 pRoamTimerStatus	960
8.985.2.6 pTTYConfigStatus	960
8.985.2.7 pVoiceDomainPrefStatus	960
8.986unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference	961

8.986.1 Detailed Description	961
8.986.2 Field Documentation	962
8.986.2.1 pAlphaIDInfo	962
8.986.2.2 ParamPresenceMask	962
8.986.2.3 pCallID	962
8.986.2.4 pCCResultType	962
8.986.2.5 pCCSUPSType	962
8.986.2.6 pFailCause	962
8.987unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference	962
8.987.1 Detailed Description	962
8.987.2 Field Documentation	962
8.987.2.1 ParamPresenceMask	962
8.987.2.2 pCallID	962
8.988unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference	962
8.988.1 Detailed Description	963
8.988.2 Field Documentation	963
8.988.2.1 callID	963
8.988.2.2 ParamPresenceMask	963
8.989unpack_voice_SLQSVoiceSUPSCallback_ind_t Struct Reference	963
8.989.1 Detailed Description	963
8.989.2 Field Documentation	965
8.989.2.1 pAlphaIDInfo	965
8.989.2.2 ParamPresenceMask	965
8.989.2.3 pCallBarPasswd	965
8.989.2.4 pCallFwdInfo	966
8.989.2.5 pCallFWNum	966
8.989.2.6 pCallFWTimerVal	966
8.989.2.7 pCallID	966
8.989.2.8 pCLIPstatus	966
8.989.2.9 pCLIRstatus	966
8.989.2.10pCNAPstatus	966
8.989.2.11pCOLPstatus	966
8.989.2.12pCOLRstatus	966
8.989.2.13pDataSrc	966
8.989.2.14pFailCause	966
8.989.2.15pNewPwdData	966
8.989.2.16pReason	966
8.989.2.17pSvcClass	966
8.989.2.18pUSSInfo	966
8.989.2.19SUPSInformation	966

8.990unpack_voice_SUPSNotificationCallback_ind_t Struct Reference	966
8.990.1 Detailed Description	966
8.990.2 Field Documentation	968
8.990.2.1 callID	968
8.990.2.2 notifType	968
8.990.2.3 ParamPresenceMask	968
8.990.2.4 pCUGIndex	968
8.990.2.5 pECTNum	968
8.991unpack_voice_USSDNotificationCallback_ind_t Struct Reference	968
8.991.1 Detailed Description	968
8.991.2 Field Documentation	968
8.991.2.1 notification_Type	968
8.991.2.2 ParamPresenceMask	968
8.991.2.3 USSDNotificationNetworkInfo	968
8.992unpack_voice_VoiceInfoRecCallback_ind_t Struct Reference	968
8.992.1 Detailed Description	969
8.992.2 Field Documentation	971
8.992.2.1 callID	971
8.992.2.2 ParamPresenceMask	971
8.992.2.3 pCalledPartyInfo	971
8.992.2.4 pCallerIDInfo	971
8.992.2.5 pCallerNameInfo	971
8.992.2.6 pCallingPartyInfo	971
8.992.2.7 pCallWaitInd	971
8.992.2.8 pCLIRCause	971
8.992.2.9 pConnectNumInfo	971
8.992.2.10pDispInfo	971
8.992.2.11pExtDispInfo	971
8.992.2.12pExtDispRecInfo	971
8.992.2.13pLineCtrlInfo	971
8.992.2.14pNSSAudioCtrl	971
8.992.2.15pNSSRelease	971
8.992.2.16pRedirNumInfo	971
8.992.2.17pSignalInfo	971
8.993unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference	971
8.993.1 Detailed Description	971
8.993.2 Field Documentation	972
8.993.2.1 callID	972
8.993.2.2 ParamPresenceMask	972
8.993.2.3 voicePrivacy	972

8.994unpack_wds_DHCPv4ClientLease_ind_t Struct Reference	972
8.994.1 Detailed Description	972
8.994.2 Field Documentation	973
8.994.2.1 DHCPv4LeaseOptTlv	973
8.994.2.2 DHCPv4LeaseStateTlv	973
8.994.2.3 IPv4AddrTlv	973
8.994.2.4 ParamPresenceMask	973
8.994.2.5 ProfileIdTlv	973
8.995unpack_wds_GetAutoconnect_t Struct Reference	973
8.995.1 Detailed Description	973
8.995.2 Field Documentation	973
8.995.2.1 ParamPresenceMask	973
8.995.2.2 psetting	973
8.996unpack_wds_GetByteTotals_t Struct Reference	973
8.996.1 Detailed Description	973
8.996.2 Field Documentation	974
8.996.2.1 ParamPresenceMask	974
8.996.2.2 pRXTotalBytes	974
8.996.2.3 pTXTotalBytes	974
8.997unpack_wds_GetConnectionRate_t Struct Reference	974
8.997.1 Detailed Description	974
8.997.2 Field Documentation	975
8.997.2.1 currentChannelRXRate	975
8.997.2.2 currentChannelTXRate	975
8.997.2.3 maxChannelRXRate	975
8.997.2.4 maxChannelTXRate	975
8.997.2.5 ParamPresenceMask	975
8.998unpack_wds_GetDataBearerTechnology_t Struct Reference	975
8.998.1 Detailed Description	975
8.998.2 Field Documentation	976
8.998.2.1 ParamPresenceMask	976
8.998.2.2 pDataBearer	976
8.999unpack_wds_GetDefaultProfile_t Struct Reference	976
8.999.1 Detailed Description	977
8.999.2 Field Documentation	978
8.999.2.1 apnname	978
8.999.2.2 apnsize	978
8.999.2.3 auth	978
8.999.2.4 ipaddr	979
8.999.2.5 ipaddrv6	979

8.999.2.6 name	979
8.999.2.7 namesize	979
8.999.2.8 ParamPresenceMask	979
8.999.2.9 pdptype	979
8.999.2.10 pridns	979
8.999.2.11 pridnsv6	979
8.999.2.12 secdns	979
8.999.2.13 secdnsv6	979
8.999.2.14 username	979
8.999.2.15 usersize	979
8.1000. inpack_wds_GetDefaultProfileNum_t Struct Reference	979
8.1000.1 Detailed Description	979
8.1000.2 Field Documentation	979
8.1000.2.1 index	979
8.1000.2.2 ParamPresenceMask	979
8.1001. inpack_wds_GetDefaultProfileV2_t Struct Reference	979
8.1001.1 Detailed Description	980
8.1001.2 Field Documentation	982
8.1001.2.1 apnname	982
8.1001.2.2 apnsize	982
8.1001.2.3 auth	982
8.1001.2.4 paddr	982
8.1001.2.5 paddrv6	982
8.1001.2.6 name	982
8.1001.2.7 namesize	982
8.1001.2.8 ParamPresenceMask	982
8.1001.2.9 pdptype	982
8.1001.2.10 pridns	982
8.1001.2.11 pridnsv6	982
8.1001.2.12 pvd	982
8.1001.2.13 pvdsize	982
8.1001.2.14 secdns	982
8.1001.2.15 secdnsv6	982
8.1001.2.16 username	982
8.1001.2.17 usersize	982
8.1002. inpack_wds_GetDormancyState_t Struct Reference	982
8.1002.1 Detailed Description	983
8.1002.2 Field Documentation	983
8.1002.2.1 dormancyState	983
8.1002.2.2 ParamPresenceMask	983

8.1003	inpack_wds_GetLastMobileIPError_t Struct Reference	983
8.1003.1	Detailed Description	983
8.1003.2	Field Documentation	983
8.1003.2.1	error	983
8.1003.2.2	ParamPresenceMask	983
8.1004	inpack_wds_GetMobileIP_t Struct Reference	984
8.1004.1	Detailed Description	984
8.1004.2	Field Documentation	984
8.1004.2.1	mipMode	984
8.1004.2.2	ParamPresenceMask	984
8.1005	inpack_wds_GetMobileIPProfile_t Struct Reference	984
8.1005.1	Detailed Description	984
8.1005.2	Field Documentation	986
8.1005.2.1	AAASPI	986
8.1005.2.2	AAASState	986
8.1005.2.3	address	986
8.1005.2.4	enabled	986
8.1005.2.5	HASPI	986
8.1005.2.6	HASState	986
8.1005.2.7	NAI	986
8.1005.2.8	naiSize	986
8.1005.2.9	ParamPresenceMask	986
8.1005.2.10	primaryHA	986
8.1005.2.11	revTunneling	986
8.1005.2.12	secondaryHA	986
8.1006	inpack_wds_GetPacketStatistics_t Struct Reference	986
8.1006.1	Detailed Description	987
8.1006.2	Field Documentation	988
8.1006.2.1	ParamPresenceMask	988
8.1006.2.2	RXDroppedCount	988
8.1006.2.3	RXOkBytesCount	988
8.1006.2.4	RXOKBytesLastCall	988
8.1006.2.5	RXPacketErrors	988
8.1006.2.6	RXPacketOverflows	988
8.1006.2.7	RXPacketSuccesses	988
8.1006.2.8	TXDroppedCount	988
8.1006.2.9	TXOkBytesCount	988
8.1006.2.10	TXOKBytesLastCall	988
8.1006.2.11	TXPacketErrors	988
8.1006.2.12	TXPacketOverflows	988

8.1006.2.1 TX PacketSuccesses	988
8.1007 in pack_wds_GetPacketStatus_t Struct Reference	988
8.1007.1Detailed Description	989
8.1007.2Field Documentation	990
8.1007.2.1ParamPresenceMask	990
8.1007.2.2XDroppedCount	990
8.1007.2.3XOkBytesCount	990
8.1007.2.4XOKBytesLastCall	990
8.1007.2.5XPacketErrors	990
8.1007.2.6XPacketOverflows	990
8.1007.2.7XPacketSuccesses	990
8.1007.2.8XDroppedCount	990
8.1007.2.9XOkBytesCount	990
8.1007.2.10XOKBytesLastCall	990
8.1007.2.11XPacketErrors	990
8.1007.2.12XPacketOverflows	990
8.1007.2.13XPacketSuccesses	990
8.1008 in pack_wds_GetSessionDuration_t Struct Reference	990
8.1008.1Detailed Description	990
8.1008.2Field Documentation	991
8.1008.2.1callDuration	991
8.1008.2.2ParamPresenceMask	991
8.1009 in pack_wds_GetSessionDurationV2_t Struct Reference	991
8.1009.1Detailed Description	991
8.1009.2Field Documentation	992
8.1009.2.1callDuration	992
8.1009.2.2ParamPresenceMask	992
8.1009.2.3pCallActiveDuration	992
8.1009.2.4pLastCallActiveDuration	992
8.1009.2.5pLastCallDuration	992
8.1010 in pack_wds_GetSessionState_t Struct Reference	992
8.1010.1Detailed Description	992
8.1010.2Field Documentation	992
8.1010.2.1connectionStatus	992
8.1010.2.2ParamPresenceMask	992
8.1011 in pack_wds_RMSetTransferStatistics_t Struct Reference	992
8.1011.1Detailed Description	992
8.1011.2Field Documentation	993
8.1011.2.1ParamPresenceMask	993
8.1012 in pack_wds_SetMobileIPProfile_t Struct Reference	993

8.1012.1Detailed Description	993
8.1012.2Field Documentation	993
8.1012.2.1ParamPresenceMask	993
8.1013unpack_wds_SLQSCreateProfile_t Struct Reference	993
8.1013.1Detailed Description	993
8.1013.2Field Documentation	994
8.1013.2.1ParamPresenceMask	994
8.1013.2.2pCreateProfileOut	994
8.1013.2.3pProfileID	994
8.1013.2.4Tlvresult	994
8.1014unpack_wds_SLQSDeleteProfile_t Struct Reference	994
8.1014.1Detailed Description	994
8.1014.2Field Documentation	994
8.1014.2.1extendedErrorCode	994
8.1014.2.2ParamPresenceMask	994
8.1015unpack_wds_SLQSDUNCallInfoCallBack_ind_t Struct Reference	994
8.1015.1Detailed Description	995
8.1015.2Field Documentation	996
8.1015.2.1CRTlv	996
8.1015.2.2CSTlv	996
8.1015.2.3DBTTlv	996
8.1015.2.4DSTlv	996
8.1015.2.5MCERTlv	996
8.1015.2.6ParamPresenceMask	996
8.1015.2.7RXBOTlv	996
8.1015.2.8TXBOTlv	996
8.1016unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference	996
8.1016.1Detailed Description	996
8.1016.2Field Documentation	997
8.1016.2.1_3gppRelease	997
8.1016.2.2defaultPDNEnabled	997
8.1016.2.3TEAttachProfile	998
8.1016.2.4TEAttachProfileList	998
8.1016.2.5TEAttachProfileListLen	998
8.1016.2.6ParamPresenceMask	998
8.1016.2.7profileList	998
8.1017unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	998
8.1017.1Detailed Description	998
8.1017.2Field Documentation	998
8.1017.2.1currNetworkInfo	998

8.1017.2.2networkInfoLen	998
8.1017.2.3ParamPresenceMask	999
8.1017.2.4prefNetwork	999
8.1018npack_wds_SLQSGetCurrentChannelRate_t Struct Reference	999
8.1018.1Detailed Description	999
8.1018.2Field Documentation	1000
8.1018.2.1current_channel_rx_rate	1000
8.1018.2.2current_channel_tx_rate	1000
8.1018.2.3max_channel_rx_rate	1000
8.1018.2.4max_channel_tx_rate	1000
8.1018.2.5ParamPresenceMask	1000
8.1019npack_wds_SLQSGetDataBearerTechnology_t Struct Reference	1000
8.1019.1Detailed Description	1000
8.1019.2Field Documentation	1000
8.1019.2.1curDataBearerTechnology	1000
8.1019.2.2dataBearerMask	1001
8.1019.2.3astCallDataBearerTechnology	1001
8.1019.2.4ParamPresenceMask	1001
8.1020npack_wds_SLQSGetDUNCallInfo_t Struct Reference	1001
8.1020.1Detailed Description	1001
8.1020.2Field Documentation	1003
8.1020.2.1callEndReason	1003
8.1020.2.2channelRate	1003
8.1020.2.3connectionStatus	1003
8.1020.2.4dataBearerTech	1003
8.1020.2.5dormancyStatus	1003
8.1020.2.6astCallDataBearerTech	1003
8.1020.2.7astCallRXOKBytesCnt	1003
8.1020.2.8astCallTXOKBytesCnt	1004
8.1020.2.9mdmCallDurationActive	1004
8.1020.2.10ParamPresenceMask	1004
8.1020.2.11kOKBytesCount	1004
8.1020.2.12OKBytesCount	1004
8.1021npack_wds_SLQSGetProfileSettings_t Struct Reference	1004
8.1021.1Detailed Description	1004
8.1021.2Field Documentation	1004
8.1021.2.1ParamPresenceMask	1004
8.1021.2.2pProfileSettings	1004
8.1021.2.3ProfileType	1004
8.1021.2.4Ivresult	1004

8.1022npack_wds_SLQSGetProfileSettingsV2_t Struct Reference	1004
8.1022.1Detailed Description	1005
8.1022.2Field Documentation	1005
8.1022.2.1ParamPresenceMask	1005
8.1022.2.2pProfileSettings	1005
8.1022.2.3ProfileType	1005
8.1022.2.4Ivresult	1005
8.1023npack_wds_SLQSGetRuntimeSettings_t Struct Reference	1005
8.1023.1Detailed Description	1006
8.1023.2Field Documentation	1007
8.1023.2.1APNName	1007
8.1023.2.2Authentication	1007
8.1023.2.3DomainList	1007
8.1023.2.4GPRSGrantedQoS	1007
8.1023.2.5GWAddressV4	1008
8.1023.2.6MCNflag	1008
8.1023.2.7PFamilyPreference	1008
8.1023.2.8Pv4	1008
8.1023.2.9PV6AddrInfo	1008
8.1023.2.10PV6GWAddrInfo	1008
8.1023.2.11Mtu	1008
8.1023.2.12ParamPresenceMask	1008
8.1023.2.13CSCFAddrPCO	1008
8.1023.2.14CSCFFQDNAddrList	1008
8.1023.2.15DPTtype	1008
8.1023.2.16PrimaryDNSV4	1008
8.1023.2.17PrimaryDNSV6	1008
8.1023.2.18ProfileID	1008
8.1023.2.19ProfileName	1008
8.1023.2.20SecondaryDNSV4	1008
8.1023.2.21SecondaryDNSV6	1008
8.1023.2.22ServerAddrList	1008
8.1023.2.23SubnetMaskV4	1008
8.1023.2.24Technology	1008
8.1023.2.25MTSGrantedQoS	1008
8.1023.2.26Username	1008
8.1024npack_wds_SLQSModifyProfile_t Struct Reference	1008
8.1024.1Detailed Description	1008
8.1024.2Field Documentation	1009
8.1024.2.1ParamPresenceMask	1009

8.1024.2.2pExtErrorCode	1009
8.1025unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference	1009
8.1025.1Detailed Description	1009
8.1025.2Field Documentation	1009
8.1025.2.1ParamPresenceMask	1009
8.1025.2.2Tlvresult	1009
8.1026unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference	1009
8.1026.1Detailed Description	1010
8.1026.2Field Documentation	1011
8.1026.2.1bearerID	1011
8.1026.2.2conn_status	1011
8.1026.2.3pFamily	1011
8.1026.2.4ParamPresenceMask	1011
8.1026.2.5reconfigReqd	1011
8.1026.2.6sessionEndReason	1011
8.1026.2.7techName	1011
8.1026.2.8verboseSessnEndReason	1011
8.1026.2.9verboseSessnEndReasonType	1011
8.1027unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference	1011
8.1027.1Detailed Description	1012
8.1027.2Field Documentation	1016
8.1027.2.1currDBTechAvail	1016
8.1027.2.2currNWInfo	1016
8.1027.2.3dataSysStatAvail	1016
8.1027.2.4dBTechAvail	1016
8.1027.2.5dBTechExtAvail	1017
8.1027.2.6dBTechExtRatValue	1017
8.1027.2.7dBTechExtSoMask	1017
8.1027.2.8dBTechology	1017
8.1027.2.9dBTechologyExt	1017
8.1027.2.10dormancyStatAvail	1017
8.1027.2.11dormancyStatus	1017
8.1027.2.12ipstatAvail	1017
8.1027.2.13ipStatus	1017
8.1027.2.14etInfoLen	1017
8.1027.2.15ParamPresenceMask	1017
8.1027.2.16refNetwork	1017
8.1027.2.17atMask	1017
8.1027.2.18_bytes	1017
8.1027.2.19_pkts	1017

8.1027.2.20Mask	1017
8.1027.2.21bytes	1017
8.1027.2.22pkts	1017
8.1027.2.23erStatAvail	1017
8.1028unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	1017
8.1028.1Detailed Description	1017
8.1028.2Field Documentation	1018
8.1028.2.1ParamPresenceMask	1018
8.1028.2.2pHwConfig	1018
8.1028.2.3pRequestOptionList	1018
8.1029unpack_wds_SLQSSGetLoopback_t Struct Reference	1018
8.1029.1Detailed Description	1018
8.1029.2Field Documentation	1019
8.1029.2.1ByteLoopbackMode	1019
8.1029.2.2ByteLoopbackMultiplier	1019
8.1029.2.3ParamPresenceMask	1019
8.1030unpack_wds_SLQSSStartDataSession_t Struct Reference	1019
8.1030.1Detailed Description	1019
8.1030.2Field Documentation	1019
8.1030.2.1ParamPresenceMask	1019
8.1030.2.2pFailureReason	1019
8.1030.2.3psid	1020
8.1030.2.4pVerboseFailReasonType	1020
8.1030.2.5pVerboseFailureReason	1020
8.1031unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	1020
8.1031.1Detailed Description	1020
8.1031.2Field Documentation	1022
8.1031.2.1apnName	1022
8.1031.2.2bearerId	1022
8.1031.2.3contextId	1022
8.1031.2.4pv4Address	1022
8.1031.2.5pv4GWAddress	1022
8.1031.2.6pv6Address	1022
8.1031.2.7pv6GWAddress	1022
8.1031.2.8ParamPresenceMask	1022
8.1031.2.9pDNSIPv4Address	1022
8.1031.2.10pDNSIPv6Address	1022
8.1031.2.11pPCSCFIPv4Address	1022
8.1031.2.12pPCSCFIPv6Address	1022
8.1031.2.13pDNSIPv4Address	1022

8.1031.2.1	1e DNSIPv6Address	1022
8.1031.2.1	5e PCSCFIPv4Address	1022
8.1031.2.1	6e PCSCFIPv6Address	1022
8.1032	InPackGetProfileSettingOut Struct Reference	1022
8.1032.1	Detailed Description	1022
8.1032.2	Field Documentation	1023
8.1032.2.1	curProfile	1023
8.1032.2.2	ExtErrCode	1023
8.1033	InPackGetProfileSettingOutV2 Struct Reference	1023
8.1033.1	Detailed Description	1023
8.1033.2	Field Documentation	1023
8.1033.2.1	curProfile	1023
8.1033.2.2	ExtErrCode	1023
8.1034	UnpackSwiAvmsEventReportBinaryUpdateSessionInfo Struct Reference	1023
8.1034.1	Detailed Description	1024
8.1034.2	Field Documentation	1025
8.1034.2.1	BinaryType	1025
8.1034.2.2	Serity	1025
8.1034.2.3	State	1025
8.1034.2.4	UserInputRequest	1025
8.1034.2.5	szDescription	1025
8.1034.2.6	szName	1025
8.1034.2.7	szVersion	1025
8.1034.2.8	TlvPresent	1025
8.1034.2.9	ulPkgDownloadComplete	1025
8.1034.2.10	ulPkgDownloadSize	1025
8.1034.2.11	DescriptionLength	1025
8.1034.2.12	NameLength	1025
8.1034.2.13	UpdateCompeteStatus	1025
8.1034.2.14	UserInputTimeout	1025
8.1034.2.15	VersionLength	1025
8.1035	InpackSwiAvmsEventReportConfig Struct Reference	1025
8.1035.1	Detailed Description	1025
8.1035.2	Field Documentation	1026
8.1035.2.1	State	1026
8.1035.2.2	UserInputRequest	1026
8.1035.2.3	szAlertMsg	1026
8.1035.2.4	TlvPresent	1026
8.1035.2.5	wAlertMsgLength	1026
8.1035.2.6	wUserInputTimeout	1026

8.1036	InpackSwiAvmsEventReportConnectionRequest Struct Reference	1026
8.1036.1	Detailed Description	1026
8.1036.2	Field Documentation	1027
8.1036.2.1	bUserInputRequest	1027
8.1036.2.2	TlvPresent	1027
8.1036.2.3	wUserInputTimeout	1027
8.1037	InpackSwiAvmsEventReportDataSessionStatus Struct Reference	1027
8.1037.1	Detailed Description	1027
8.1037.2	Field Documentation	1027
8.1037.2.1	bType	1027
8.1037.2.2	TlvPresent	1027
8.1037.2.3	wErrorCode	1027
8.1038	InpackSwiAvmsEventReportHTTPStatus Struct Reference	1027
8.1038.1	Detailed Description	1027
8.1038.2	Field Documentation	1028
8.1038.2.1	TlvPresent	1028
8.1038.2.2	wHTTPStatus	1029
8.1039	InpackSwiAvmsEventReportNotification Struct Reference	1029
8.1039.1	Detailed Description	1029
8.1039.2	Field Documentation	1029
8.1039.2.1	bNotification	1030
8.1039.2.2	TlvPresent	1030
8.1039.2.3	wSessionStatus	1030
8.1040	InpackSwiAvmsEventReportPackageID Struct Reference	1030
8.1040.1	Detailed Description	1030
8.1040.2	Field Documentation	1030
8.1040.2.1	bPackageID	1030
8.1040.2.2	TlvPresent	1030
8.1041	InpackSwiAvmsEventReportRegStatus Struct Reference	1030
8.1041.1	Detailed Description	1030
8.1041.2	Field Documentation	1030
8.1041.2.1	bRegStatus	1030
8.1041.2.2	TlvPresent	1031
8.1042	InpackSwiAvmsEventReportSessionType Struct Reference	1031
8.1042.1	Detailed Description	1031
8.1042.2	Field Documentation	1031
8.1042.2.1	bType	1031
8.1042.2.2	TlvPresent	1031
8.1043	InpackSwiAvmsEventReportWAMSPParamChange Struct Reference	1031
8.1043.1	Detailed Description	1031

8.1043.2Field Documentation	1032
8.1043.2.1TlvPresent	1032
8.1043.2.2wWamsChangeMask	1032
8.1044InpackWdsProfileParam Union Reference	1032
8.1044.1Detailed Description	1032
8.1044.2Field Documentation	1032
8.1044.2.1SlqsProfile3GPP	1032
8.1044.2.2SlqsProfile3GPP2	1032
8.1045InpackWdsProfileParamV2 Union Reference	1032
8.1045.1Detailed Description	1032
8.1045.2Field Documentation	1033
8.1045.2.1SlqsProfile3GPP	1033
8.1045.2.2SlqsProfile3GPP2	1033
8.1046Voice_airTimer Struct Reference	1033
8.1046.1Detailed Description	1033
8.1046.2Field Documentation	1033
8.1046.2.1airTimerValue	1033
8.1046.2.2hamID	1033
8.1047Voice_allCallsAlphaIDInfo Struct Reference	1033
8.1047.1Detailed Description	1033
8.1047.2Field Documentation	1034
8.1047.2.1AlphaIDInfo	1034
8.1047.2.2callID	1034
8.1048Voice_allCallsDiagInfo Struct Reference	1034
8.1048.1Detailed Description	1034
8.1048.2Field Documentation	1034
8.1048.2.1callID	1034
8.1048.2.2DiagInfo	1034
8.1049Voice_allCallsUUSInfo Struct Reference	1034
8.1049.1Detailed Description	1034
8.1049.2Field Documentation	1035
8.1049.2.1callID	1035
8.1049.2.2uusInfo	1035
8.1050Voice_alphaIDInfo Struct Reference	1035
8.1050.1Detailed Description	1035
8.1050.2Field Documentation	1035
8.1050.2.1alphaDcs	1035
8.1050.2.2alphaLen	1035
8.1050.2.3alphaText	1035
8.1051Voice_arrAlertingPattern Struct Reference	1035

8.1051.1Detailed Description	1036
8.1051.2Field Documentation	1036
8.1051.2.1alertingPattern	1036
8.1051.2.2callID	1036
8.1051.2.3numInstances	1036
8.1052voice_arrAlertingType Struct Reference	1036
8.1052.1Detailed Description	1036
8.1052.2Field Documentation	1037
8.1052.2.1AlertingType	1037
8.1052.2.2callID	1037
8.1052.2.3numInstances	1037
8.1053voice_arrAlphaID Struct Reference	1037
8.1053.1Detailed Description	1037
8.1053.2Field Documentation	1037
8.1053.2.1allCallsAlphaIDInfoArr	1037
8.1053.2.2numInstances	1037
8.1054voice_arrCalledPartyNum Struct Reference	1037
8.1054.1Detailed Description	1038
8.1054.2Field Documentation	1038
8.1054.2.1CalledPartyNum	1038
8.1054.2.2numInstances	1038
8.1055voice_arrCallEndReason Struct Reference	1038
8.1055.1Detailed Description	1038
8.1055.2Field Documentation	1039
8.1055.2.1callEndReason	1039
8.1055.2.2callID	1039
8.1055.2.3numInstances	1039
8.1056voice_arrCallInfo Struct Reference	1039
8.1056.1Detailed Description	1039
8.1056.2Field Documentation	1039
8.1056.2.1getAllCallInfo	1039
8.1056.2.2numInstances	1039
8.1057voice_arrConnectPartyNum Struct Reference	1039
8.1057.1Detailed Description	1039
8.1057.2Field Documentation	1040
8.1057.2.1ConnectedPartyNum	1040
8.1057.2.2numInstances	1040
8.1058voice_arrDiagInfo Struct Reference	1040
8.1058.1Detailed Description	1040
8.1058.2Field Documentation	1040

8.1058.2.1DiagInfo	1040
8.1058.2.2numInstances	1040
8.1059voice_arrRedirPartyNum Struct Reference	1040
8.1059.1Detailed Description	1040
8.1059.2Field Documentation	1041
8.1059.2.1numInstances	1041
8.1059.2.2RedirPartyNum	1041
8.1060voice_arrRemotePartyName Struct Reference	1041
8.1060.1Detailed Description	1041
8.1060.2Field Documentation	1041
8.1060.2.1GetAllCallRmtPtyName	1041
8.1060.2.2numInstances	1041
8.1061voice_arrRemotePartyNum Struct Reference	1041
8.1061.1Detailed Description	1042
8.1061.2Field Documentation	1042
8.1061.2.1numInstances	1042
8.1061.2.2RmtPtyNum	1042
8.1062voice_arrSvcOption Struct Reference	1042
8.1062.1Detailed Description	1042
8.1062.2Field Documentation	1042
8.1062.2.1callID	1042
8.1062.2.2numInstances	1042
8.1062.2.3srvOption	1043
8.1063voice_arrUUSInfo Struct Reference	1043
8.1063.1Detailed Description	1043
8.1063.2Field Documentation	1043
8.1063.2.1AllCallsUUSInfo	1043
8.1063.2.2numInstances	1043
8.1064voice_burstDTMFInfo Struct Reference	1043
8.1064.1Detailed Description	1043
8.1064.2Field Documentation	1044
8.1064.2.1digitCnt	1044
8.1064.2.2pCallID	1044
8.1064.2.3pDigitBuff	1044
8.1065voice_calledPartyInfo Struct Reference	1044
8.1065.1Detailed Description	1044
8.1065.2Field Documentation	1045
8.1065.2.1number	1045
8.1065.2.2numLen	1045
8.1065.2.3numPlan	1045

8.1065.2.4numType	1045
8.1065.2.5PI	1045
8.1065.2.6SI	1045
8.1066voice_calledPartySubAdd Struct Reference	1045
8.1066.1Detailed Description	1046
8.1066.2Field Documentation	1046
8.1066.2.1extBit	1046
8.1066.2.2oddEvenInd	1046
8.1066.2.3subAddr	1046
8.1066.2.4subAddrLen	1046
8.1066.2.5subAddrType	1046
8.1067voice_callerIDInfo Struct Reference	1046
8.1067.1Detailed Description	1047
8.1067.2Field Documentation	1047
8.1067.2.1callerID	1047
8.1067.2.2callerIDLen	1047
8.1067.2.3PI	1047
8.1068voice_callFwdTypeAndPlan Struct Reference	1047
8.1068.1Detailed Description	1047
8.1068.2Field Documentation	1048
8.1068.2.1numberPlan	1048
8.1068.2.2numberType	1048
8.1069voice_callFWExtInfo Struct Reference	1048
8.1069.1Detailed Description	1048
8.1069.2Field Documentation	1049
8.1069.2.1noReplyTimer	1049
8.1069.2.2number	1049
8.1069.2.3numLen	1050
8.1069.2.4numPlan	1050
8.1069.2.5numType	1050
8.1069.2.6PI	1050
8.1069.2.7SI	1050
8.1069.2.8SvcClass	1050
8.1069.2.9SvcStatus	1050
8.1070voice_callFWInfo Struct Reference	1050
8.1070.1Detailed Description	1050
8.1070.2Field Documentation	1050
8.1070.2.1noReplyTimer	1050
8.1070.2.2number	1051
8.1070.2.3numLen	1051

8.1070.2.4SvcClass	1051
8.1070.2.5SvcStatus	1051
8.1071voice_callInfo Struct Reference	1051
8.1071.1Detailed Description	1051
8.1071.2Field Documentation	1052
8.1071.2.1callID	1052
8.1071.2.2callState	1052
8.1071.2.3callType	1052
8.1071.2.4direction	1052
8.1071.2.5mode	1052
8.1072voice_callingPartyInfo Struct Reference	1052
8.1072.1Detailed Description	1053
8.1072.2Field Documentation	1054
8.1072.2.1number	1054
8.1072.2.2numLen	1054
8.1072.2.3numPlan	1054
8.1072.2.4numType	1054
8.1072.2.5PI	1054
8.1072.2.6SI	1054
8.1073voice_ccSUPSType Struct Reference	1054
8.1073.1Detailed Description	1054
8.1073.2Field Documentation	1054
8.1073.2.1reason	1055
8.1073.2.2svcType	1055
8.1074voice_CLIPResp Struct Reference	1055
8.1074.1Detailed Description	1055
8.1074.2Field Documentation	1055
8.1074.2.1ActiveStatus	1055
8.1074.2.2ProvisionStatus	1055
8.1075voice_CLIRResp Struct Reference	1055
8.1075.1Detailed Description	1055
8.1075.2Field Documentation	1056
8.1075.2.1ActiveStatus	1056
8.1075.2.2ProvisionStatus	1056
8.1076voice_CNAPResp Struct Reference	1056
8.1076.1Detailed Description	1056
8.1076.2Field Documentation	1057
8.1076.2.1ActiveStatus	1057
8.1076.2.2ProvisionStatus	1057
8.1077voice_COLPResp Struct Reference	1057

8.1077.1Detailed Description	1057
8.1077.2Field Documentation	1057
8.1077.2.1ActiveStatus	1057
8.1077.2.2ProvisionStatus	1057
8.1078voice_COLRResp Struct Reference	1057
8.1078.1Detailed Description	1057
8.1078.2Field Documentation	1058
8.1078.2.1ActiveStatus	1058
8.1078.2.2ProvisionStatus	1058
8.1079voice_connectNumInfo Struct Reference	1058
8.1079.1Detailed Description	1058
8.1079.2Field Documentation	1059
8.1079.2.1callerID	1059
8.1079.2.2callerIDLen	1059
8.1079.2.3numPlan	1059
8.1079.2.4numPresInd	1060
8.1079.2.5numType	1060
8.1079.2.6screeningInd	1060
8.1080voice_CUGInfo Struct Reference	1060
8.1080.1Detailed Description	1060
8.1080.2Field Documentation	1060
8.1080.2.1CUGIndex	1060
8.1080.2.2SuppOA	1060
8.1080.2.3SuppPrefCUG	1060
8.1081voice_curAMRConfig Struct Reference	1060
8.1081.1Detailed Description	1060
8.1081.2Field Documentation	1061
8.1081.2.1gsmAmrStat	1061
8.1081.2.2wcdmaAmrStat	1061
8.1082voice_diagInfo Struct Reference	1061
8.1082.1Detailed Description	1061
8.1082.2Field Documentation	1061
8.1082.2.1diagInfoLen	1061
8.1082.2.2diagnosticInfo	1061
8.1083voice_DTMFInfo Struct Reference	1062
8.1083.1Detailed Description	1062
8.1083.2Field Documentation	1062
8.1083.2.1callID	1062
8.1083.2.2digitBuff	1062
8.1083.2.3digitCnt	1062

8.1083.2.4DTMFEvent	1062
8.1084voice_DTMFLengths Struct Reference	1062
8.1084.1Detailed Description	1063
8.1084.2Field Documentation	1063
8.1084.2.1DTMFInterdigitInterval	1063
8.1084.2.2DTMFPulseWidth	1063
8.1085voice_ECTNum Struct Reference	1063
8.1085.1Detailed Description	1063
8.1085.2Field Documentation	1064
8.1085.2.1ECTCallState	1064
8.1085.2.2number	1064
8.1085.2.3presentationInd	1064
8.1086voice_extDispRecInfo Struct Reference	1064
8.1086.1Detailed Description	1064
8.1086.2Field Documentation	1064
8.1086.2.1dispType	1064
8.1086.2.2extDispInfo	1064
8.1086.2.3extDispInfoLen	1065
8.1087voice_getAllCallInformation Struct Reference	1065
8.1087.1Detailed Description	1065
8.1087.2Field Documentation	1065
8.1087.2.1ALS	1065
8.1087.2.2Callinfo	1065
8.1087.2.3sEmpty	1065
8.1088voice_getAllCallRmtPtyName Struct Reference	1065
8.1088.1Detailed Description	1065
8.1088.2Field Documentation	1066
8.1088.2.1callID	1066
8.1088.2.2RemotePartyName	1066
8.1089voice_getAllCallRmtPtyNum Struct Reference	1066
8.1089.1Detailed Description	1066
8.1089.2Field Documentation	1066
8.1089.2.1callID	1066
8.1089.2.2RemotePartyNum	1066
8.1090voice_getCallFWExtInfo Struct Reference	1066
8.1090.1Detailed Description	1066
8.1090.2Field Documentation	1067
8.1090.2.1CallFWExtInfo	1067
8.1090.2.2numInstances	1067
8.1091voice_getCallFWInfo Struct Reference	1067

8.1091.1 Detailed Description	1067
8.1091.2 Field Documentation	1067
8.1091.2.1 CallFWInfo	1067
8.1091.2.2 numInstances	1067
8.1092 Voice_lineCtrlInfo Struct Reference	1067
8.1092.1 Detailed Description	1068
8.1092.2 Field Documentation	1068
8.1092.2.1 polarityIncluded	1068
8.1092.2.2 pwrDenialTime	1068
8.1092.2.3 revPolarity	1068
8.1092.2.4 toggleMode	1068
8.1093 Voice_newPwdData Struct Reference	1068
8.1093.1 Detailed Description	1068
8.1093.2 Field Documentation	1069
8.1093.2.1 newPwd	1069
8.1093.2.2 newPwdAgain	1069
8.1094 Voice_NSSAudioCtrl Struct Reference	1069
8.1094.1 Detailed Description	1069
8.1094.2 Field Documentation	1069
8.1094.2.1 downLink	1069
8.1094.2.2 upLink	1069
8.1095 Voice_peerNumberInfo Struct Reference	1069
8.1095.1 Detailed Description	1069
8.1095.2 Field Documentation	1070
8.1095.2.1 callID	1071
8.1095.2.2 number	1071
8.1095.2.3 numLen	1071
8.1095.2.4 numPI	1071
8.1095.2.5 numPlan	1071
8.1095.2.6 numSI	1071
8.1095.2.7 numType	1071
8.1096 Voice_prefVoiceSO Struct Reference	1071
8.1096.1 Detailed Description	1071
8.1096.2 Field Documentation	1073
8.1096.2.1 evrcCapability	1073
8.1096.2.2 homeOrigVoiceSO	1073
8.1096.2.3 homePageVoiceSO	1073
8.1096.2.4 namID	1073
8.1096.2.5 roamOrigVoiceSO	1073
8.1097 Voice_redirNumInfo Struct Reference	1073

8.1097.1Detailed Description	1073
8.1097.2Field Documentation	1074
8.1097.2.1number	1074
8.1097.2.2numLen	1074
8.1097.2.3numPlan	1074
8.1097.2.4numType	1074
8.1097.2.5PI	1074
8.1097.2.6reason	1074
8.1097.2.7SI	1074
8.1098voice_remotePartyName Struct Reference	1075
8.1098.1Detailed Description	1075
8.1098.2Field Documentation	1075
8.1098.2.1callerName	1075
8.1098.2.2codingScheme	1075
8.1098.2.3nameLen	1075
8.1098.2.4namePI	1075
8.1099voice_remotePartyNum Struct Reference	1075
8.1099.1Detailed Description	1076
8.1099.2Field Documentation	1076
8.1099.2.1numLen	1076
8.1099.2.2presentationInd	1076
8.1099.2.3remPartyNumber	1076
8.1100voice_roamTimer Struct Reference	1076
8.1100.1Detailed Description	1076
8.1100.2Field Documentation	1077
8.1100.2.1namID	1077
8.1100.2.2roamTimerValue	1077
8.1101voice_signalInfo Struct Reference	1077
8.1101.1Detailed Description	1077
8.1101.2Field Documentation	1077
8.1101.2.1alertPitch	1077
8.1101.2.2signal	1077
8.1101.2.3signalType	1077
8.1102voice_SUPSInfo Struct Reference	1078
8.1102.1Detailed Description	1078
8.1102.2Field Documentation	1078
8.1102.2.1isModByCC	1078
8.1102.2.2svcType	1078
8.1103voice_USSDNotificationNetworkInfo Struct Reference	1078
8.1103.1Detailed Description	1078

8.1103.2Field Documentation	1079
8.1103.2.1networkInfo	1079
8.1103.2.2lvPresent	1079
8.1104oice_USSInfo Struct Reference	1079
8.1104.1Detailed Description	1079
8.1104.2Field Documentation	1079
8.1104.2.1ussData	1079
8.1104.2.2ussDCS	1079
8.1104.2.3ussLen	1079
8.1105oice_UUSInfo Struct Reference	1079
8.1105.1Detailed Description	1080
8.1105.2Field Documentation	1080
8.1105.2.1UUSData	1080
8.1105.2.2JUUSDataLen	1080
8.1105.2.3JUUSDCS	1080
8.1105.2.4JUUSType	1080
8.1106nds_channelRate Struct Reference	1081
8.1106.1Detailed Description	1081
8.1106.2Field Documentation	1081
8.1106.2.1CurrChanRxRate	1081
8.1106.2.2CurrChanTxRate	1081
8.1107nds_ChannelRateTlv Struct Reference	1081
8.1107.1Detailed Description	1081
8.1107.2Field Documentation	1081
8.1107.2.1ChannelRate	1081
8.1107.2.2TlvPresent	1081
8.1108nds_ConnStatusTlv Struct Reference	1082
8.1108.1Detailed Description	1082
8.1108.2Field Documentation	1082
8.1108.2.1MDMConnStatus	1082
8.1108.2.2TlvPresent	1082
8.1109nds_currNetworkInfo Struct Reference	1082
8.1109.1Detailed Description	1082
8.1109.2Field Documentation	1083
8.1109.2.1NetworkType	1083
8.1109.2.2RATMask	1083
8.1109.2.3SOMask	1083
8.1110nds_DataBearTechTlv Struct Reference	1083
8.1110.1Detailed Description	1083
8.1110.2Field Documentation	1084

8.1110.2.1DataBearerTech	1084
8.1110.2.2TlvPresent	1084
8.1111vds_DataULongLongTlv Struct Reference	1084
8.1111.1Detailed Description	1084
8.1111.2Field Documentation	1085
8.1111.2.1TlvPresent	1085
8.1111.2.2AllData	1085
8.1112vds_DataULongTlv Struct Reference	1085
8.1112.1Detailed Description	1085
8.1112.2Field Documentation	1085
8.1112.2.1TlvPresent	1085
8.1112.2.2AllData	1085
8.1113vds_DHCPLeaseOptTlv Struct Reference	1085
8.1113.1Detailed Description	1085
8.1113.2Field Documentation	1085
8.1113.2.1numOpt	1085
8.1113.2.2optList	1086
8.1113.2.3optListData	1086
8.1113.2.4TlvPresent	1086
8.1114vds_DHCPLeaseStateTlv Struct Reference	1086
8.1114.1Detailed Description	1086
8.1114.2Field Documentation	1086
8.1114.2.1leaseState	1086
8.1114.2.2TlvPresent	1086
8.1115vds_DHCPOpt Struct Reference	1086
8.1115.1Detailed Description	1086
8.1115.2Field Documentation	1087
8.1115.2.1optCode	1087
8.1115.2.2optValLen	1087
8.1115.2.3optVal	1087
8.1116vds_DHCPProfileIdTlv Struct Reference	1087
8.1116.1Detailed Description	1087
8.1116.2Field Documentation	1087
8.1116.2.1profileId	1087
8.1116.2.2profileType	1087
8.1116.2.3TlvPresent	1087
8.1117vds_DHCPv4HWConfig Struct Reference	1087
8.1117.1Detailed Description	1088
8.1117.2Field Documentation	1088
8.1117.2.1chaddr	1088

8.1117.2.2chaddrLen	1088
8.1117.2.3hwType	1088
8.1118.1nds_DHCPv4Option Struct Reference	1088
8.1118.1.1Detailed Description	1088
8.1118.2Field Documentation	1089
8.1118.2.1optCode	1089
8.1118.2.2optVal	1089
8.1118.2.3optValLen	1089
8.1119.1nds_DHCPv4OptionList Struct Reference	1089
8.1119.1.1Detailed Description	1089
8.1119.2Field Documentation	1089
8.1119.2.1numOpt	1089
8.1119.2.2pOptList	1089
8.1120.1nds_DHCPv4Profile Struct Reference	1089
8.1120.1.1Detailed Description	1089
8.1120.2Field Documentation	1090
8.1120.2.1profileId	1090
8.1120.2.2profileType	1090
8.1121.1nds_Domain Struct Reference	1090
8.1121.1.1Detailed Description	1090
8.1121.2Field Documentation	1090
8.1121.2.1domainLen	1090
8.1121.2.2domainName	1090
8.1122.1nds_DomainNameList Struct Reference	1090
8.1122.1.1Detailed Description	1090
8.1122.2Field Documentation	1091
8.1122.2.1domain	1091
8.1122.2.2numInstances	1091
8.1123.1nds_DormStatTlv Struct Reference	1091
8.1123.1.1Detailed Description	1091
8.1123.2Field Documentation	1091
8.1123.2.1DormancyStat	1091
8.1123.2.2TlvPresent	1091
8.1124.1nds_GPRSQoS Struct Reference	1091
8.1124.1.1Detailed Description	1092
8.1124.2Field Documentation	1092
8.1124.2.1delayClass	1092
8.1124.2.2meanThroughputClass	1092
8.1124.2.3peakThroughputClass	1092
8.1124.2.4precedenceClass	1092

8.1124.2.5reliabilityClass	1092
8.1125.1ds_IPv4AdTlv Struct Reference	1092
8.1125.1Detailed Description	1092
8.1125.2Field Documentation	1093
8.1125.2.1IPv4Addr	1093
8.1125.2.2TlvPresent	1093
8.1126.1ds_IPV6AddressInfo Struct Reference	1093
8.1126.1Detailed Description	1093
8.1126.2Field Documentation	1093
8.1126.2.1IPAddressV6	1093
8.1126.2.2IPv6PrefixLen	1093
8.1127.1ds_IPV6GWAddressInfo Struct Reference	1093
8.1127.1Detailed Description	1093
8.1127.2Field Documentation	1094
8.1127.2.1gwAddressV6	1094
8.1127.2.2gwV6PrefixLen	1094
8.1128.1ds_LastMdmCallEndRsnTlv Struct Reference	1094
8.1128.1Detailed Description	1094
8.1128.2Field Documentation	1094
8.1128.2.1CallEndReason	1094
8.1128.2.2TlvPresent	1094
8.1129.1ds_PCSCFFQDNAddress Struct Reference	1094
8.1129.1Detailed Description	1095
8.1129.2Field Documentation	1095
8.1129.2.1fqdnAddr	1095
8.1129.2.2fqdnLen	1095
8.1130.1ds_PCSCFFQDNAddressList Struct Reference	1095
8.1130.1Detailed Description	1095
8.1130.2Field Documentation	1095
8.1130.2.1numInstances	1095
8.1130.2.2pcsfQDNAddress	1095
8.1131.1ds_PCSCFIPv4ServerAddressList Struct Reference	1095
8.1131.1Detailed Description	1096
8.1131.2Field Documentation	1096
8.1131.2.1numInstances	1096
8.1131.2.2pcsfIPv4Addr	1096
8.1132.1ds_ProfileIdentifier Struct Reference	1096
8.1132.1Detailed Description	1096
8.1132.2Field Documentation	1096
8.1132.2.1profileIndex	1097

8.1132.2.2profileType	1097
8.1133.1vds_profileInfo Union Reference	1097
8.1133.1Detailed Description	1097
8.1133.2Field Documentation	1097
8.1133.2.1SlqsProfile3GPP	1097
8.1133.2.2SlqsProfile3GPP2	1097
8.1134.1vds_RXBytesOKTlv Struct Reference	1097
8.1134.1Detailed Description	1097
8.1134.2Field Documentation	1097
8.1134.2.1RxByteOKCnt	1097
8.1134.2.2TlvPresent	1097
8.1135.1vds_transferStatInd Struct Reference	1098
8.1135.1Detailed Description	1098
8.1135.2Field Documentation	1098
8.1135.2.1StatsMask	1098
8.1135.2.2StatsPeriod	1098
8.1136.1vds_TrStatInd Struct Reference	1098
8.1136.1Detailed Description	1098
8.1136.2Field Documentation	1099
8.1136.2.1statsMask	1099
8.1136.2.2statsPeriod	1099
8.1137.1vds_TXBytesOKTlv Struct Reference	1099
8.1137.1Detailed Description	1099
8.1137.2Field Documentation	1099
8.1137.2.1TlvPresent	1099
8.1137.2.2TxByteOKCnt	1100
8.1138.1vds_UMTSMinQoS Struct Reference	1100
8.1138.1Detailed Description	1100
8.1138.2Field Documentation	1101
8.1138.2.1deliveryErrSDU	1101
8.1138.2.2grntDownlinkBitrate	1102
8.1138.2.3grntUplinkBitrate	1102
8.1138.2.4maxDownlinkBitrate	1102
8.1138.2.5maxSDUSize	1102
8.1138.2.6maxUplinkBitrate	1102
8.1138.2.7qosDeliveryOrder	1102
8.1138.2.8resBerRatio	1102
8.1138.2.9sduErrorRatio	1102
8.1138.2.10trafficClass	1102
8.1138.2.11trafficPriority	1102

8.1138.2.1	TransferDelay	1102
8.1139	ndsDhcpv4HwConfig Struct Reference	1102
8.1139.1	Detailed Description	1102
8.1139.2	Field Documentation	1102
8.1139.2.1	chaddr	1102
8.1139.2.2	chaddrLen	1102
8.1139.2.3	hwType	1102
8.1140	ndsDhcpv4Option Struct Reference	1103
8.1140.1	Detailed Description	1103
8.1140.2	Field Documentation	1103
8.1140.2.1	optCode	1103
8.1140.2.2	optVal	1103
8.1140.2.3	optValLen	1103
8.1141	ndsDhcpv4OptionList Struct Reference	1103
8.1141.1	Detailed Description	1103
8.1141.2	Field Documentation	1103
8.1141.2.1	numOpt	1104
8.1141.2.2	optList	1104
8.1142	ndsDhcpv4ProfileId Struct Reference	1104
8.1142.1	Detailed Description	1104
8.1142.2	Field Documentation	1104
8.1142.2.1	profileId	1104
8.1142.2.2	profileType	1104
9	File Documentation	1105
9.1	apdoxypages.c File Reference	1105
9.1.1	Detailed Description	1105
9.2	audio.h File Reference	1105
9.2.1	Function Documentation	1106
9.2.1.1	pack_audio_SLQSGetAudioPathConfig	1106
9.2.1.2	pack_audio_SLQSGetAudioProfile	1106
9.2.1.3	pack_audio_SLQSGetAudioVolTLBConfig	1107
9.2.1.4	pack_audio_SLQSSetAudioPathConfig	1107
9.2.1.5	pack_audio_SLQSSetAudioProfile	1108
9.2.1.6	pack_audio_SLQSSetAudioVolTLBConfig	1108
9.2.1.7	unpack_audio_SLQSGetAudioPathConfig	1109
9.2.1.8	unpack_audio_SLQSGetAudioProfile	1109
9.2.1.9	unpack_audio_SLQSGetAudioVolTLBConfig	1109
9.2.1.10	unpack_audio_SLQSSetAudioPathConfig	1110
9.2.1.11	unpack_audio_SLQSSetAudioProfile	1110

9.2.1.12	unpack_audio_SLQSSetAudioVoTLBConfig	1110
9.3	cat.h File Reference	1111
9.3.1	Macro Definition Documentation	1112
9.3.1.1	CAN_COMMON_EVENT_TLV_NUMBER	1112
9.3.1.2	CAT_EVENT_DATA_MAX_LENGTH	1112
9.3.2	Typedef Documentation	1112
9.3.2.1	unpack_cat_CATSendEnvelopeCommand_t	1112
9.3.2.2	unpack_cat_CATSendTerminalResponse_t	1112
9.3.3	Function Documentation	1112
9.3.3.1	pack_cat_CATSendEnvelopeCommand	1112
9.3.3.2	pack_cat_CATSendTerminalResponse	1112
9.3.3.3	pack_cat_SetCATEventCallback	1113
9.3.3.4	unpack_cat_CATSendEnvelopeCommand	1113
9.3.3.5	unpack_cat_CATSendTerminalResponse	1113
9.3.3.6	unpack_cat_SetCATEventCallback	1114
9.3.3.7	unpack_cat_SetCatEventCallback_ind	1114
9.4	common.h File Reference	1114
9.4.1	Macro Definition Documentation	1116
9.4.1.1	DEFAULT_LOC_TIMEOUT_IN_SEC	1116
9.4.1.2	MINREQBKLEN	1116
9.4.1.3	MSGID_AND_LEN	1116
9.4.1.4	MSGID_DONT_CARE	1116
9.4.1.5	SDK_VALIDATE_INPUT_PACK_PARAM	1116
9.4.1.6	SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID	1117
9.4.1.7	SDK_VALIDATE_INPUT_UNPACK_PARAM	1117
9.4.1.8	SDU_HDR_LEN	1117
9.4.1.9	SWI_INIT_UNPACK_RESULT_VALUE	1117
9.4.1.10	UNUSEDPARAM	1117
9.4.2	Typedef Documentation	1117
9.4.2.1	logger	1117
9.4.3	Enumeration Type Documentation	1117
9.4.3.1	eLOG_LEVEL	1117
9.4.3.2	eQMI_SVC	1117
9.4.3.3	eTimeout	1118
9.4.3.4	msgtype	1118
9.4.4	Function Documentation	1119
9.4.4.1	fill_pack_ctx	1119
9.4.4.2	fill_sdu_hdr	1119
9.4.4.3	get_version	1119
9.4.4.4	helper_get_error_code	1119

9.4.4.5	helper_get_error_reason	1119
9.4.4.6	helper_get_req_str	1119
9.4.4.7	helper_get_resp_ctx	1119
9.4.4.8	helper_get_xid	1119
9.4.4.9	helper_isBootLoader_DebugEnabled	1119
9.4.4.10	helper_set_log_func	1120
9.4.4.11	helper_set_log_lvl	1120
9.4.4.12	liteqmi_GetVersion	1120
9.4.4.13	liteqmi_helper_decode7bitAsciiEncString	1120
9.4.4.14	liteqmi_log	1120
9.4.4.15	unpack_result_code_only	1120
9.4.5	Variable Documentation	1120
9.4.5.1	glog	1120
9.4.5.2	gloglvl	1120
9.5	dms.h File Reference	1120
9.5.1	Macro Definition Documentation	1128
9.5.1.1	ACT_CODE_MAX_SIZE	1128
9.5.1.2	CK_MAX_SIZE	1128
9.5.1.3	DMS_IMGDETAILS_LEN	1128
9.5.1.4	DMS_MAX_CUST_ID_LEN	1128
9.5.1.5	DMS_MAX_CUST_VALUE_LEN	1128
9.5.1.6	DMS_MAX_FWUPDATE_LOG_STR_SZ	1128
9.5.1.7	DMS_MAX_FWUPDATE_REF_STR_SZ	1129
9.5.1.8	DMS_MAX_RADIO_IFCS_SIZE	1129
9.5.1.9	DMS_MAX_SUBS_CFG_LIST_SIZE	1129
9.5.1.10	DMS_MAX_SUBS_LIST_SIZE	1129
9.5.1.11	DMS_MAX_SUPPORTED_LTE_BANDS	1129
9.5.1.12	DMS_PM_FACTORY	1129
9.5.1.13	DMS_PM_LOW	1129
9.5.1.14	DMS_PM_OFFLINE	1129
9.5.1.15	DMS_PM_ONLINE	1129
9.5.1.16	DMS_PM_PERSISTENT_LOW	1129
9.5.1.17	DMS_PM_RESET	1129
9.5.1.18	DMS_PM_SHUT_DOWN	1129
9.5.1.19	DMS_SET_REG_IND_DISABLE	1129
9.5.1.20	DMS_SET_REG_IND_ENABLE	1129
9.5.1.21	DMS_SET_REG_IND_NO_CHANGE	1129
9.5.1.22	DMS_SET_REPORT_DISABLE	1129
9.5.1.23	DMS_SET_REPORT_ENABLE	1129
9.5.1.24	DMS_SLQSFWINFO_APPVERSION_SZ	1129

9.5.1.25	DMS_SLQSFWINFO_BOOTVERSION_SZ	1129
9.5.1.26	DMS_SLQSFWINFO_CARRIER_SZ	1129
9.5.1.27	DMS_SLQSFWINFO_CUR_CARR_NAME	1129
9.5.1.28	DMS_SLQSFWINFO_CUR_CARR_REV	1129
9.5.1.29	DMS_SLQSFWINFO_MODELID_SZ	1129
9.5.1.30	DMS_SLQSFWINFO_PACKAGEID_SZ	1129
9.5.1.31	DMS_SLQSFWINFO_PRIVERSION_SZ	1129
9.5.1.32	DMS_SLQSFWINFO_SKU_SZ	1129
9.5.1.33	DMS_SWI_SET_IND_DISABLE	1129
9.5.1.34	DMS_SWI_SET_IND_ENABLE	1129
9.5.1.35	DMS_UINT8_MAX_STRING_SZ	1130
9.5.1.36	DMS_VALID_FSN_LEN	1130
9.5.1.37	ERI_DATA_MAX_SIZE	1130
9.5.1.38	MAX_BUILD_ID_LEN	1130
9.5.1.39	MEID_MAX_SIZE	1130
9.5.1.40	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	1130
9.5.1.41	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	1130
9.5.1.42	SPC_SIZE	1130
9.5.1.43	UNIQUE_ID_LEN	1130
9.5.2	Function Documentation	1130
9.5.2.1	pack_dms_ActivateAutomatic	1130
9.5.2.2	pack_dms_GetActivationState	1130
9.5.2.3	pack_dms_GetBandCapability	1131
9.5.2.4	pack_dms_GetCrashAction	1131
9.5.2.5	pack_dms_GetCustFeature	1131
9.5.2.6	pack_dms_GetCustFeaturesV2	1132
9.5.2.7	pack_dms_GetDeviceCap	1132
9.5.2.8	pack_dms_GetDeviceCapabilities	1133
9.5.2.9	pack_dms_GetDeviceCapabilitiesV2	1133
9.5.2.10	pack_dms_GetDeviceHardwareRev	1133
9.5.2.11	pack_dms_GetDeviceMfr	1134
9.5.2.12	pack_dms_GetDeviceSerialNumbers	1134
9.5.2.13	pack_dms_GetFirmwareInfo	1134
9.5.2.14	pack_dms_GetFirmwareRevision	1135
9.5.2.15	pack_dms_GetFirmwareRevisions	1135
9.5.2.16	pack_dms_GetFSN	1135
9.5.2.17	pack_dms_GetHardwareRevision	1136
9.5.2.18	pack_dms_GetIMSI	1136
9.5.2.19	pack_dms_GetManufacturer	1136
9.5.2.20	pack_dms_GetModelID	1137

9.5.2.21	pack_dms_GetNetworkTime	1137
9.5.2.22	pack_dms_GetNetworkTimeV2	1137
9.5.2.23	pack_dms_GetOfflineReason	1138
9.5.2.24	pack_dms_GetPower	1138
9.5.2.25	pack_dms_GetPRLVersion	1138
9.5.2.26	pack_dms_GetSerialNumbers	1139
9.5.2.27	pack_dms_GetUSBComp	1139
9.5.2.28	pack_dms_GetVoiceNumber	1140
9.5.2.29	pack_dms_ResetToFactoryDefaults	1140
9.5.2.30	pack_dms_SetActivationStatusCallback	1140
9.5.2.31	pack_dms_SetCrashAction	1141
9.5.2.32	pack_dms_SetCustFeature	1141
9.5.2.33	pack_dms_SetCustFeaturesV2	1141
9.5.2.34	pack_dms_SetEventReport	1142
9.5.2.35	pack_dms_SetFirmwarePreference	1142
9.5.2.36	pack_dms_SetIndicationRegister	1142
9.5.2.37	pack_dms_SetPower	1143
9.5.2.38	pack_dms_SetUSBComp	1143
9.5.2.39	pack_dms_SLQSDmsSwiGetPCInfo	1144
9.5.2.40	pack_dms_SLQSDmsSwiGetResetInfo	1144
9.5.2.41	pack_dms_SLQSDmsSwiGetUimSelection	1144
9.5.2.42	pack_dms_SLQSDmsSwiIndicationRegister	1145
9.5.2.43	pack_dms_SLQSGetBandCapability	1145
9.5.2.44	pack_dms_SLQSGetERIFile	1145
9.5.2.45	pack_dms_SLQSGetPowerSaveModeConfig	1146
9.5.2.46	pack_dms_SLQSSetPowerSaveModeConfig	1146
9.5.2.47	pack_dms_SLQSSwiClearDyingGaspStatistics	1147
9.5.2.48	pack_dms_SLQSSwiGetCrashInfo	1147
9.5.2.49	pack_dms_SLQSSwiGetDyingGaspCfg	1147
9.5.2.50	pack_dms_SLQSSwiGetDyingGaspStatistics	1148
9.5.2.51	pack_dms_SLQSSwiGetFirmwareCurr	1148
9.5.2.52	pack_dms_SLQSSwiGetFwUpdateStatus	1148
9.5.2.53	pack_dms_SLQSSwiGetHostDevInfo	1149
9.5.2.54	pack_dms_SLQSSwiGetOSInfo	1149
9.5.2.55	pack_dms_SLQSSwiGetSerialNoExt	1149
9.5.2.56	pack_dms_SLQSSwiSetDyingGaspCfg	1150
9.5.2.57	pack_dms_SLQSSwiSetHostDevInfo	1150
9.5.2.58	pack_dms_SLQSSwiSetOSInfo	1150
9.5.2.59	pack_dms_SLQSUIMGetState	1151
9.5.2.60	pack_dms_SwiSetEventReport	1151

9.5.2.61	pack_dms_SwiUimSelect	1151
9.5.2.62	pack_dms_UIMChangePIN	1152
9.5.2.63	pack_dms_UIMGetControlKeyStatus	1152
9.5.2.64	pack_dms_UIMGetICCID	1153
9.5.2.65	pack_dms_UIMGetPINStatus	1153
9.5.2.66	pack_dms_UIMSetControlKeyProtection	1153
9.5.2.67	pack_dms_UIMSetPINProtection	1154
9.5.2.68	pack_dms_UIMUnblockControlKey	1154
9.5.2.69	pack_dms_UIMUnblockPIN	1154
9.5.2.70	pack_dms_UIMVerifyPIN	1155
9.5.2.71	pack_dms_ValidateSPC	1155
9.5.2.72	unpack_dms_ActivateAutomatic	1156
9.5.2.73	unpack_dms_GetActivationState	1156
9.5.2.74	unpack_dms_GetBandCapability	1156
9.5.2.75	unpack_dms_GetCrashAction	1157
9.5.2.76	unpack_dms_GetCustFeature	1157
9.5.2.77	unpack_dms_GetCustFeaturesV2	1157
9.5.2.78	unpack_dms_GetDeviceCap	1158
9.5.2.79	unpack_dms_GetDeviceCapabilities	1158
9.5.2.80	unpack_dms_GetDeviceCapabilitiesV2	1158
9.5.2.81	unpack_dms_GetDeviceHardwareRev	1159
9.5.2.82	unpack_dms_GetDeviceMfr	1159
9.5.2.83	unpack_dms_GetDeviceSerialNumbers	1159
9.5.2.84	unpack_dms_GetFirmwareInfo	1160
9.5.2.85	unpack_dms_GetFirmwareRevision	1160
9.5.2.86	unpack_dms_GetFirmwareRevisions	1161
9.5.2.87	unpack_dms_GetFSN	1161
9.5.2.88	unpack_dms_GetHardwareRevision	1161
9.5.2.89	unpack_dms_GetIMSI	1162
9.5.2.90	unpack_dms_GetManufacturer	1162
9.5.2.91	unpack_dms_GetModelID	1162
9.5.2.92	unpack_dms_GetNetworkTime	1163
9.5.2.93	unpack_dms_GetNetworkTimeV2	1163
9.5.2.94	unpack_dms_GetOfflineReason	1163
9.5.2.95	unpack_dms_GetPower	1164
9.5.2.96	unpack_dms_GetPRLVersion	1164
9.5.2.97	unpack_dms_GetSerialNumbers	1164
9.5.2.98	unpack_dms_GetUSBComp	1165
9.5.2.99	unpack_dms_GetVoiceNumber	1165
9.5.2.100	unpack_dms_PSMCfgChange_ind	1165

9.5.2.101 unpack_dms_ResetToFactoryDefaults	1166
9.5.2.102 unpack_dms_SetActivationStatusCallback	1166
9.5.2.103 unpack_dms_SetCrashAction	1166
9.5.2.104 unpack_dms_SetCustFeature	1167
9.5.2.105 unpack_dms_SetCustFeaturesV2	1167
9.5.2.106 unpack_dms_SetEventReport	1167
9.5.2.107 unpack_dms_SetEventReport_ind	1168
9.5.2.108 unpack_dms_SetFirmwarePreference	1168
9.5.2.109 unpack_dms_SetIndicationRegister	1169
9.5.2.110 unpack_dms_SetPower	1169
9.5.2.111 unpack_dms_SetUSBComp	1169
9.5.2.112 unpack_dms_SLQSDmsSwiGetPCInfo	1170
9.5.2.113 unpack_dms_SLQSDmsSwiGetResetInfo	1170
9.5.2.114 unpack_dms_SLQSDmsSwiGetResetInfo_Ind	1170
9.5.2.115 unpack_dms_SLQSDmsSwiGetUimSelection	1171
9.5.2.116 unpack_dms_SLQSDmsSwiIndicationRegister	1171
9.5.2.117 unpack_dms_SLQSGetBandCapability	1172
9.5.2.118 unpack_dms_SLQSGetBandCapabilityExt	1172
9.5.2.119 unpack_dms_SLQSGetERIFile	1172
9.5.2.120 unpack_dms_SLQSGetPowerSaveModeConfig	1173
9.5.2.121 unpack_dms_SLQSSetPowerSaveModeConfig	1173
9.5.2.122 unpack_dms_SLQSSwiClearDyingGaspStatistics	1173
9.5.2.123 unpack_dms_SLQSSwiGetCrashInfo	1174
9.5.2.124 unpack_dms_SLQSSwiGetDyingGaspCfg	1174
9.5.2.125 unpack_dms_SLQSSwiGetDyingGaspStatistics	1174
9.5.2.126 unpack_dms_SLQSSwiGetFirmwareCurr	1175
9.5.2.127 unpack_dms_SLQSSwiGetFwUpdateStatus	1175
9.5.2.128 unpack_dms_SLQSSwiGetHostDevInfo	1175
9.5.2.129 unpack_dms_SLQSSwiGetOSInfo	1176
9.5.2.130 unpack_dms_SLQSSwiGetSerialNoExt	1176
9.5.2.131 unpack_dms_SLQSSwiSetDyingGaspCfg	1176
9.5.2.132 unpack_dms_SLQSSwiSetHostDevInfo	1177
9.5.2.133 unpack_dms_SLQSSwiSetOSInfo	1177
9.5.2.134 unpack_dms_SLQSUIMGetState	1177
9.5.2.135 unpack_dms_SwiEventReportCallBack_ind	1178
9.5.2.136 unpack_dms_SwiSetEventReport	1178
9.5.2.137 unpack_dms_SwiUimSelect	1178
9.5.2.138 unpack_dms_UIMChangePIN	1179
9.5.2.139 unpack_dms_UIMGetControlKeyStatus	1179
9.5.2.140 unpack_dms_UIMGetICCID	1179

9.5.2.141	unpack_dms_UIMGetPINStatus	1180
9.5.2.142	unpack_dms_UIMSetControlKeyProtection	1180
9.5.2.143	unpack_dms_UIMSetPINProtection	1181
9.5.2.144	unpack_dms_UIMUnblockControlKey	1181
9.5.2.145	unpack_dms_UIMUnblockPIN	1181
9.5.2.146	unpack_dms_UIMVerifyPIN	1182
9.5.2.147	unpack_dms_ValidateSPC	1182
9.6	fms.h File Reference	1182
9.6.1	Macro Definition Documentation	1183
9.6.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	1183
9.6.1.2	FMS_GOBI_LISTENTRIES_MAX	1183
9.6.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	1183
9.6.1.4	FMS_GOBI_MBN_IMG_ID_SIZE	1183
9.6.1.5	FMS_GOBI_MBN_IMG_ID_STR_LEN	1183
9.6.1.6	FMS_IMAGE_ID_BUILD_ID_LEN	1183
9.6.1.7	FMS_IMAGE_ID_IMG_ID_LEN	1183
9.6.1.8	FMS_IMAGE_ID_MAX_ENTRIES	1184
9.6.1.9	FMS_IMAGE_ID_PRI_IMGTYPE	1184
9.6.1.10	FMS_MAX_IMAGE_ID_ELEMENT	1184
9.6.1.11	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	1184
9.6.1.12	LITE_TOTAL_IMAGE_ID_STRING_SIZE	1184
9.6.2	Function Documentation	1184
9.6.2.1	GetValidFwPriCombinations	1184
9.6.2.2	pack_fms_GetImagesPreference	1184
9.6.2.3	pack_fms_GetStoredImages	1185
9.6.2.4	pack_fms_SetImagesPreference	1185
9.6.2.5	unpack_fms_GetImagesPreference	1185
9.6.2.6	unpack_fms_GetStoredImages	1185
9.6.2.7	unpack_fms_SetImagesPreference	1185
9.7	ims.h File Reference	1186
9.7.1	Macro Definition Documentation	1188
9.7.1.1	MAX_NAME_LEN	1188
9.7.2	Typedef Documentation	1188
9.7.2.1	unpack_ims_SLQSImsConfigIndicationRegister_t	1188
9.7.3	Function Documentation	1188
9.7.3.1	pack_ims_SLQSGetIMSSMSConfig	1188
9.7.3.2	pack_ims_SLQSGetIMSUserConfig	1188
9.7.3.3	pack_ims_SLQSGetIMSVoIPConfig	1189
9.7.3.4	pack_ims_SLQSGetRegMgrConfig	1189
9.7.3.5	pack_ims_SLQSGetSIPConfig	1189

9.7.3.6	pack_ims_SLQSImsConfigIndicationRegister	1190
9.7.3.7	pack_ims_SLQSSetIMSSMSConfig	1190
9.7.3.8	pack_ims_SLQSSetIMSUserConfig	1191
9.7.3.9	pack_ims_SLQSSetIMSVoIPConfig	1191
9.7.3.10	pack_ims_SLQSSetRegMgrConfig	1192
9.7.3.11	pack_ims_SLQSSetSIPConfig	1192
9.7.3.12	unpack_ims_SLQSGetIMSSMSConfig	1193
9.7.3.13	unpack_ims_SLQSGetIMSUserConfig	1193
9.7.3.14	unpack_ims_SLQSGetIMSVoIPConfig	1193
9.7.3.15	unpack_ims_SLQSGetRegMgrConfig	1194
9.7.3.16	unpack_ims_SLQSGetSIPConfig	1194
9.7.3.17	unpack_ims_SLQSImsConfigIndicationRegister	1194
9.7.3.18	unpack_ims_SLQSRegMgrCfgCallBack_ind	1195
9.7.3.19	unpack_ims_SLQSSetIMSSMSConfig	1195
9.7.3.20	unpack_ims_SLQSSetIMSUserConfig	1195
9.7.3.21	unpack_ims_SLQSSetIMSVoIPConfig	1196
9.7.3.22	unpack_ims_SLQSSetRegMgrConfig	1196
9.7.3.23	unpack_ims_SLQSSetSIPConfig	1196
9.7.3.24	unpack_ims_SLQSSIPCfgCallBack_ind	1197
9.7.3.25	unpack_ims_SLQSSMSCfgCallBack_ind	1197
9.7.3.26	unpack_ims_SLQSUserCfgCallBack_ind	1197
9.7.3.27	unpack_ims_SLQSVoIPCfgCallBack_ind	1198
9.8	imsa.h File Reference	1198
9.8.1	Macro Definition Documentation	1199
9.8.1.1	MAX_ERROR_CODE_LEN	1199
9.8.2	Typedef Documentation	1199
9.8.2.1	unpack_imsa_SLQSRegisterIMSAIndication_t	1199
9.8.3	Function Documentation	1199
9.8.3.1	pack_imsa_SLQSGetIMSARegStatus	1199
9.8.3.2	pack_imsa_SLQSGetIMSAServiceStatus	1200
9.8.3.3	pack_imsa_SLQSRegisterIMSAIndication	1200
9.8.3.4	unpack_imsa_SLQSGetIMSARegStatus	1201
9.8.3.5	unpack_imsa_SLQSGetIMSAServiceStatus	1201
9.8.3.6	unpack_imsa_SLQSImsaPdpStatusCallBack_ind	1202
9.8.3.7	unpack_imsa_SLQSImsaRatStatusCallBack_ind	1202
9.8.3.8	unpack_imsa_SLQSImsaRegStatusCallBack_ind	1202
9.8.3.9	unpack_imsa_SLQSImsaSvcStatusCallBack_ind	1203
9.8.3.10	unpack_imsa_SLQSRegisterIMSAIndication	1203
9.9	lte-fw.h File Reference	1203
9.9.1	Detailed Description	1206

9.9.2	Macro Definition Documentation	1206
9.9.2.1	FIRMWARE_BCHDATESIZE	1206
9.9.2.2	FIRMWARE_BCHVERSTRSIZE	1206
9.9.2.3	FIRMWARE_IMAGE_SIZE_MAX	1206
9.9.2.4	FIRMWARE_INFO_STRING_SIZE	1206
9.9.2.5	IMG_MASK_CLEAR	1206
9.9.2.6	IMG_MASK_GENERIC	1206
9.9.2.7	IMG_MASK_MDM	1206
9.9.2.8	IMG_MASK_PRI	1207
9.9.2.9	libSDP_BuildImagesPreferenceRequest	1207
9.9.2.10	libSDP_CalculateImageMask	1207
9.9.2.11	LIBSDP_CARRIER_PACKAGE_SKU	1207
9.9.2.12	libSDP_CheckValidFirmwareInfo	1207
9.9.2.13	libSDP_DownloadFW	1207
9.9.2.14	libSDP_ExtractFirmwareParametersByPath	1207
9.9.2.15	libSDP_getFileType	1207
9.9.2.16	libSDP_GetModelFamily	1207
9.9.2.17	libSDP_GetVersion	1207
9.9.2.18	libsdp_set_log_func	1207
9.9.2.19	libsdp_SetReadBlockSize	1207
9.9.2.20	LIBSDP_SKU_STRING_LENGTH	1207
9.9.2.21	LITEFW_CARRIER_PACKAGE_SKU	1207
9.9.2.22	LITEFW_SKU_STRING_LENGTH	1207
9.9.2.23	MAX_IMAGE_PRODUCT_LENGTH	1207
9.9.2.24	NULL_TERMINATOR_CHAR_SIZE	1207
9.9.3	Typedef Documentation	1207
9.9.3.1	libSDP_FirmwareInfo	1207
9.9.3.2	libsdplogger	1207
9.9.3.3	litefw_FirmwareFileInfo	1207
9.9.3.4	litefw_FirmwareInfo	1208
9.9.3.5	litefw_FirmwarePartNo	1209
9.9.3.6	litefwlogger	1209
9.9.4	Enumeration Type Documentation	1209
9.9.4.1	litefw_fileimgtype	1209
9.9.4.2	litefw_Fw_Type	1209
9.9.4.3	litefw_fwdwl_error_codes	1210
9.9.4.4	litefw_imagetype	1210
9.9.4.5	litefw_Models	1210
9.9.4.6	litefw_QDL_FLOW_CONTROLS	1211
9.9.4.7	litefw_QDL_MODEs	1211

9.9.5	Function Documentation	1211
9.9.5.1	litefw_BuildImagesPreferenceRequest	1211
9.9.5.2	litefw_CalculateImageMask	1211
9.9.5.3	litefw_CheckValidFirmwareInfo	1212
9.9.5.4	litefw_DownloadFW	1212
9.9.5.5	litefw_ExtractFirmwareParametersByPath	1213
9.9.5.6	litefw_ExtractFirmwarePartNoByPath	1213
9.9.5.7	litefw_getFileType	1213
9.9.5.8	litefw_GetModelFamily	1214
9.9.5.9	litefw_GetQTLDownloadMode	1214
9.9.5.10	litefw_GetQTLHWFlowControl	1214
9.9.5.11	litefw_GetVersion	1214
9.9.5.12	litefw_logsenable	1215
9.9.5.13	litefw_set_log_func	1215
9.9.5.14	litefw_SetQTLDownloadMode	1215
9.9.5.15	litefw_SetQTLHWFlowControl	1215
9.9.5.16	litefw_SetReadBlockSize	1216
9.9.5.17	litefw_SLQSGetFirmwareFileInfo	1216
9.9.5.18	litefw_switch_9x07_to_downloadmode	1216
9.9.5.19	litefw_switch_to_BootHoldMode	1216
9.10	loc.h File Reference	1217
9.10.1	Macro Definition Documentation	1221
9.10.1.1	LOC_UINT8_MAX_STRING_SZ	1221
9.10.1.2	LOCEVENTMASKBATCHFULLNOTIFICATION	1221
9.10.1.3	LOCEVENTMASKENGINESTATE	1221
9.10.1.4	LOCEVENTMASKFIXSESSIONSTATE	1221
9.10.1.5	LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION	1221
9.10.1.6	LOCEVENTMASKGEOFENCEBREACHNOTIFICATION	1221
9.10.1.7	LOCEVENTMASKGEOFENCEGENALERT	1221
9.10.1.8	LOCEVENTMASKGNSSMEASUREMENTREPORT	1221
9.10.1.9	LOCEVENTMASKGNSSSVINFO	1221
9.10.1.10	LOCEVENTMASKINJECTPOSITIONREQ	1221
9.10.1.11	LOCEVENTMASKINJECTPREDICTEDORBITSREQ	1221
9.10.1.12	LOCEVENTMASKINJECTTIMERREQ	1222
9.10.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	1222
9.10.1.14	LOCEVENTMASKINVALIDVALUE	1222
9.10.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	1222
9.10.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	1222
9.10.1.17	LOCEVENTMASKMOTIONDATACONTROL	1222
9.10.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	1222

9.10.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	1222
9.10.1.20	LOCEVENTMASKNMEA	1222
9.10.1.21	LOCEVENTMASKPEDOMETERCONTROL	1222
9.10.1.22	LOCEVENTMASKPOSITIONREPORT	1222
9.10.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	1223
9.10.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	1223
9.10.1.25	LOCEVENTMASKTIMESYNCREQ	1223
9.10.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	1223
9.10.1.27	LOCEVENTMASKWIFIREQ	1223
9.10.1.28	MAX_LOC_NMEA_STR_LEN	1223
9.10.1.29	MAX_SENSOR_DATA_LEN	1223
9.10.1.30	MAX_TEMP_DATA_LEN	1223
9.10.2	Typedef Documentation	1223
9.10.2.1	unpack_loc_DeleteAssistData_t	1223
9.10.2.2	unpack_loc_SLQSLOCGetServer_t	1223
9.10.2.3	unpack_loc_SLQSLOCInjectPosition_t	1223
9.10.2.4	unpack_loc_SLQSLOCInjectSensorData_t	1223
9.10.2.5	unpack_loc_SLQSLOCInjectUTCTime_t	1223
9.10.2.6	unpack_loc_SLQSLOCSetCradleMountConfig_t	1223
9.10.3	Enumeration Type Documentation	1223
9.10.3.1	anonymous enum	1223
9.10.4	Function Documentation	1224
9.10.4.1	pack_loc_DeleteAssistData	1224
9.10.4.2	pack_loc_EventRegister	1224
9.10.4.3	pack_loc_SetExtPowerState	1224
9.10.4.4	pack_loc_SetOperationMode	1225
9.10.4.5	pack_loc_SLQSLOCGetBestAvailPos	1225
9.10.4.6	pack_loc_SLQSLOCGetOpMode	1225
9.10.4.7	pack_loc_SLQSLOCGetServer	1226
9.10.4.8	pack_loc_SLQSLOCInjectPosition	1226
9.10.4.9	pack_loc_SLQSLOCInjectSensorData	1227
9.10.4.10	pack_loc_SLQSLOCInjectUTCTime	1227
9.10.4.11	pack_loc_SLQSLOCSetCradleMountConfig	1227
9.10.4.12	pack_loc_SLQSLOCSetServer	1228
9.10.4.13	pack_loc_Start	1228
9.10.4.14	pack_loc_Stop	1229
9.10.4.15	unpack_loc_BestAvailPos_Ind	1229
9.10.4.16	unpack_loc_CradleMountCallback_Ind	1229
9.10.4.17	unpack_loc_DeleteAssistData	1230
9.10.4.18	unpack_loc_DeleteAssistData_Ind	1230

9.10.4.19 unpack_loc_EngineState_Ind	1230
9.10.4.20 unpack_loc_EventNMEA_Ind	1231
9.10.4.21 unpack_loc_EventRegister	1231
9.10.4.22 unpack_loc_EventTimeSyncCallback_Ind	1231
9.10.4.23 unpack_loc_GetOpMode_Ind	1232
9.10.4.24 unpack_loc_GetServer_Ind	1232
9.10.4.25 unpack_loc_GnssSvInfo_Ind	1232
9.10.4.26 unpack_loc_InjectPositionCallback_Ind	1233
9.10.4.27 unpack_loc_InjectSensorDataCallback_Ind	1233
9.10.4.28 unpack_loc_InjectTimeSyncDataCallback_Ind	1233
9.10.4.29 unpack_loc_InjectUTCTimeCallback_Ind	1234
9.10.4.30 unpack_loc_PositionRpt_Ind	1234
9.10.4.31 unpack_loc_SensorStreamingCallback_Ind	1234
9.10.4.32 unpack_loc_SetExtPowerConfig_Ind	1235
9.10.4.33 unpack_loc_SetExtPowerState	1235
9.10.4.34 unpack_loc_SetOperationMode	1235
9.10.4.35 unpack_loc_SetOperationMode_Ind	1236
9.10.4.36 unpack_loc_SetServer_Ind	1236
9.10.4.37 unpack_loc_SLQSLOCGetBestAvailPos	1236
9.10.4.38 unpack_loc_SLQSLOCGetOpMode	1237
9.10.4.39 unpack_loc_SLQSLOCGetServer	1237
9.10.4.40 unpack_loc_SLQSLOCInjectPosition	1237
9.10.4.41 unpack_loc_SLQSLOCInjectSensorData	1238
9.10.4.42 unpack_loc_SLQSLOCInjectUTCTime	1238
9.10.4.43 unpack_loc_SLQSLOCSetCradleMountConfig	1238
9.10.4.44 unpack_loc_SLQSLOCSetServer	1239
9.10.4.45 unpack_loc_Start	1239
9.10.4.46 unpack_loc_Stop	1239
9.11 nas.h File Reference	1240
9.11.1 Macro Definition Documentation	1252
9.11.1.1 LITE_NW_SCAN_LTE_OP_MODE_MAX_LENGTH	1252
9.11.1.2 NAS_IMSI_M_S1_LENGTH	1252
9.11.1.3 NAS_IMSI_M_S2_LENGTH	1252
9.11.1.4 NAS_MAX_DESCRIPTION_LENGTH	1252
9.11.1.5 NAS_MAX_NUM_NETWORKS	1252
9.11.1.6 NAS_MAX_PHY_CA_AGG_SCELL_ARRRY_SIZE	1252
9.11.1.7 NAS_MAX_SCC_RX_INFO_INSTANCES	1252
9.11.1.8 NAS_MAX_SLQS_NAS_PCI_INFO_LENGTH	1252
9.11.1.9 NAS_MAX_SLQS_NAS_PCI_INFO_PLMN_LENGTH	1252
9.11.1.10 NAS_MCC_MNC_INST_LENGTH	1252

9.11.1.11	NAS_NAM_NAME_LENGTH	1252
9.11.1.12	NAS_OTA_MESSAGE_MAX_BUF_SIZE	1252
9.11.1.13	NAS_PLMN_LENGTH	1252
9.11.1.14	NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST	1252
9.11.2	Typedef Documentation	1252
9.11.2.1	nas_LTEBandPrefExtTlv	1252
9.11.2.2	nas_NR5gBandPrefTlv	1252
9.11.2.3	unpack_nas_InitiateDomainAttach_t	1253
9.11.2.4	unpack_nas_SetACCOLC_t	1253
9.11.2.5	unpack_nas_SetCDMANetworkParameters_t	1253
9.11.2.6	unpack_nas_SetLURejectCallback_t	1253
9.11.2.7	unpack_nas_SetRFInfoCallback_t	1253
9.11.2.8	unpack_nas_SLQSConfigSigInfo_t	1253
9.11.2.9	unpack_nas_SLQSInitiateNetworkRegistration_t	1253
9.11.2.10	unpack_nas_SLQSNasConfigSigInfo2_t	1253
9.11.2.11	unpack_nas_SLQSNasIndicationRegisterExt_t	1253
9.11.2.12	unpack_nas_SLQSNASSeteDRXParams_t	1253
9.11.2.13	unpack_nas_SLQSNasSwiIndicationRegister_t	1253
9.11.2.14	unpack_nas_SLQSNASSwiSetChannelLock_t	1253
9.11.2.15	unpack_nas_SLQSSetBandPreference_t	1253
9.11.2.16	unpack_nas_SLQSSetSignalStrengthsCallback_t	1253
9.11.2.17	unpack_nas_SLQSSetSysSelectionPref_t	1253
9.11.2.18	unpack_nas_SLQSSetSysSelectionPrefExt_t	1253
9.11.2.19	unpack_nas_SLQSSwiPSDetach_t	1253
9.11.3	Enumeration Type Documentation	1253
9.11.3.1	LITE_TYPE_OF_SERVICE_DOMAIN	1253
9.11.3.2	LITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND	1253
9.11.3.3	LITEQMI_NAS_LTE_CPHY_CA_BW_NRB	1255
9.11.3.4	LITEQMI_NAS_LTE_CPHY_SCELL_STATE	1255
9.11.3.5	NAS_LTE_CPHY_CA_BW_NRB_LITE	1255
9.11.3.6	NAS_LTE_CPHY_SCELL_STATE_LITE	1255
9.11.4	Function Documentation	1255
9.11.4.1	pack_nas_GetACCOLC	1255
9.11.4.2	pack_nas_GetANAAAAAuthenticationStatus	1256
9.11.4.3	pack_nas_GetCDMANetworkParameters	1256
9.11.4.4	pack_nas_GetHomeNetwork	1256
9.11.4.5	pack_nas_GetHomeNetwork3GPP2	1257
9.11.4.6	pack_nas_GetNetworkPreference	1257
9.11.4.7	pack_nas_GetRFInfo	1257
9.11.4.8	pack_nas_GetServingNetwork	1258

9.11.4.9 pack_nas_GetServingNetworkCapabilities	1258
9.11.4.10 pack_nas_GetSignalStrengths	1258
9.11.4.11 pack_nas_InitiateDomainAttach	1259
9.11.4.12 pack_nas_PerformNetworkScan	1259
9.11.4.13 pack_nas_PerformNetworkScanPCI	1259
9.11.4.14 pack_nas_SetACCOLC	1260
9.11.4.15 pack_nas_SetCDMANetworkParameters	1260
9.11.4.16 pack_nas_SetLURejectCallback	1260
9.11.4.17 pack_nas_SetNetworkPreference	1261
9.11.4.18 pack_nas_SetRFInfoCallback	1261
9.11.4.19 pack_nas_SLQSConfigSigInfo	1261
9.11.4.20 pack_nas_SLQSGetErrorRate	1262
9.11.4.21 pack_nas_SlqsGetLTECphyCAInfo	1262
9.11.4.22 pack_nas_SLQSGetNetworkTime	1262
9.11.4.23 pack_nas_SLQSGetOperatorNameData	1263
9.11.4.24 pack_nas_SLQSGetPLMNName	1263
9.11.4.25 pack_nas_SLQSGetServingSystem	1263
9.11.4.26 pack_nas_SLQSGetSignalStrength	1264
9.11.4.27 pack_nas_SLQSGetSysInfo	1264
9.11.4.28 pack_nas_SLQSGetSysSelectionPref	1264
9.11.4.29 pack_nas_SLQSGetSysSelectionPrefExt	1265
9.11.4.30 pack_nas_SLQSInitiateNetworkRegistration	1265
9.11.4.31 pack_nas_SLQSNasConfigSigInfo2	1266
9.11.4.32 pack_nas_SLQSNasGet3GPP2Subscription	1266
9.11.4.33 pack_nas_SLQSNasGetCellLocationInfo	1266
9.11.4.34 pack_nas_SLQSNASGeteDRXParams	1267
9.11.4.35 pack_nas_SLQSNASGeteDRXParamsExt	1267
9.11.4.36 pack_nas_SLQSNASGetForbiddenNetworks	1267
9.11.4.37 pack_nas_SLQSNasGetHDRColorCode	1268
9.11.4.38 pack_nas_SLQSNasGetRFInfo	1268
9.11.4.39 pack_nas_SLQSNasGetSigInfo	1268
9.11.4.40 pack_nas_SLQSNasGetTxRxInfo	1269
9.11.4.41 pack_nas_SLQSNasIndicationRegisterExt	1269
9.11.4.42 pack_nas_SLQSNASSeteDRXParams	1270
9.11.4.43 pack_nas_SLQSNASSwiGetChannelLock	1270
9.11.4.44 pack_nas_SLQSNasSwiIndicationRegister	1270
9.11.4.45 pack_nas_SLQSNasSwiModemStatus	1271
9.11.4.46 pack_nas_SLQSNASSwiSetChannelLock	1271
9.11.4.47 pack_nas_SLQSSetBandPreference	1271
9.11.4.48 pack_nas_SLQSSetSignalStrengthsCallback	1272

9.11.4.49 pack_nas_SLQSSetSysSelectionPref	1272
9.11.4.50 pack_nas_SLQSSetSysSelectionPrefExt	1272
9.11.4.51 pack_nas_SLQSSwiGetHDRPersonality	1273
9.11.4.52 pack_nas_SLQSSwiGetHDRProtSubtype	1273
9.11.4.53 pack_nas_SLQSSwiGetHRPDStats	1273
9.11.4.54 pack_nas_SLQSSwiGetLteCQI	1274
9.11.4.55 pack_nas_SLQSSwiGetLteSccRxInfo	1274
9.11.4.56 pack_nas_SLQSSwiNetworkDebug	1274
9.11.4.57 pack_nas_SLQSSwiPSDetach	1275
9.11.4.58 unpack_nas_GetACCOLC	1275
9.11.4.59 unpack_nas_GetANAAAAuthenticationStatus	1276
9.11.4.60 unpack_nas_GetCDMANetworkParameters	1276
9.11.4.61 unpack_nas_GetHomeNetwork	1276
9.11.4.62 unpack_nas_GetHomeNetwork3GPP2	1277
9.11.4.63 unpack_nas_GetNetworkPreference	1277
9.11.4.64 unpack_nas_GetRFInfo	1277
9.11.4.65 unpack_nas_GetServingNetwork	1278
9.11.4.66 unpack_nas_GetServingNetworkCapabilities	1278
9.11.4.67 unpack_nas_GetSignalStrengths	1278
9.11.4.68 unpack_nas_InitiateDomainAttach	1279
9.11.4.69 unpack_nas_PerformNetworkScan	1279
9.11.4.70 unpack_nas_SetACCOLC	1280
9.11.4.71 unpack_nas_SetCDMANetworkParameters	1280
9.11.4.72 unpack_nas_SetDataCapabilitiesCallback_ind	1280
9.11.4.73 unpack_nas_SetEventReportInd	1281
9.11.4.74 unpack_nas_SetLURRejectCallback	1281
9.11.4.75 unpack_nas_SetNasLTECphyCaIndCallback_ind	1281
9.11.4.76 unpack_nas_SetNetworkPreference	1282
9.11.4.77 unpack_nas_SetRFInfoCallback	1282
9.11.4.78 unpack_nas_SetRoamingIndicatorCallback_ind	1282
9.11.4.79 unpack_nas_SetServingSystemCallback_ind	1283
9.11.4.80 unpack_nas_SLQSConfigSigInfo	1283
9.11.4.81 unpack_nas_SLQSGetErrorRate	1283
9.11.4.82 unpack_nas_SlqsGetLTECphyCAInfo	1284
9.11.4.83 unpack_nas_SLQSGetNetworkTime	1284
9.11.4.84 unpack_nas_SLQSGetOperatorNameData	1284
9.11.4.85 unpack_nas_SLQSGetPLMNName	1285
9.11.4.86 unpack_nas_SLQSGetServingSystem	1285
9.11.4.87 unpack_nas_SLQSGetSignalStrength	1285
9.11.4.88 unpack_nas_SLQSGetSysInfo	1286

9.11.4.89 unpack_nas_SLQSGetSysSelectionPref	1286
9.11.4.90 unpack_nas_SLQSGetSysSelectionPrefExt	1287
9.11.4.91 unpack_nas_SLQSInitiateNetworkRegistration	1287
9.11.4.92 unpack_nas_SLQSNasConfigSigInfo2	1287
9.11.4.93 unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind	1288
9.11.4.94 unpack_nas_SLQSNasGet3GPP2Subscription	1288
9.11.4.95 unpack_nas_SLQSNasGetCellLocationInfo	1289
9.11.4.96 unpack_nas_SLQSNASGeteDRXParams	1289
9.11.4.97 unpack_nas_SLQSNASGeteDRXParamsExt	1289
9.11.4.98 unpack_nas_SLQSNASGetForbiddenNetworks	1290
9.11.4.99 unpack_nas_SLQSNasGetHDRColorCode	1290
9.11.4.100unpack_nas_SLQSNasGetRFInfo	1290
9.11.4.101unpack_nas_SLQSNasGetSigInfo	1291
9.11.4.102unpack_nas_SLQSNasGetTxRxInfo	1291
9.11.4.103unpack_nas_SLQSNasIndicationRegisterExt	1292
9.11.4.104unpack_nas_SLQSNasNetworkRejectCallback_Ind	1292
9.11.4.105unpack_nas_SLQSNasNetworkTimeCallBack_ind	1292
9.11.4.106unpack_nas_SLQSNASSeteDRXParams	1293
9.11.4.107unpack_nas_SLQSNasSigInfoCallback_ind	1293
9.11.4.108unpack_nas_SLQSNASSwiGetChannelLock	1293
9.11.4.109unpack_nas_SLQSNasSwiIndicationRegister	1294
9.11.4.110unpack_nas_SLQSNasSwiModemStatus	1294
9.11.4.111unpack_nas_SLQSNasSwiOTAMessageCallback_ind	1294
9.11.4.112unpack_nas_SLQSNASSwiSetChannelLock	1295
9.11.4.113unpack_nas_SLQSNasSysInfoCallback_ind	1295
9.11.4.114unpack_nas_SLQSNasTimerCallback_ind	1295
9.11.4.115unpack_nas_SLQSSetBandPreference	1296
9.11.4.116unpack_nas_SLQSSetSignalStrengthsCallback	1296
9.11.4.117unpack_nas_SLQSSetSysSelectionPref	1296
9.11.4.118unpack_nas_SLQSSetSysSelectionPrefCallBack_ind	1297
9.11.4.119unpack_nas_SLQSSetSysSelectionPrefExt	1297
9.11.4.120unpack_nas_SLQSSwiGetHDRPersonality	1297
9.11.4.121unpack_nas_SLQSSwiGetHDRProtSubtype	1298
9.11.4.122unpack_nas_SLQSSwiGetHRPDStats	1298
9.11.4.123unpack_nas_SLQSSwiGetLteCQI	1298
9.11.4.124unpack_nas_SLQSSwiGetLteSccRxInfo	1299
9.11.4.125unpack_nas_SLQSSwiHDRPersonalityCallback_Ind	1299
9.11.4.126unpack_nas_SLQSSwiNetworkDebug	1299
9.11.4.127unpack_nas_SLQSSwiPSDetach	1300
9.11.4.128unpack_nas_SLQSSwiRandIndicatorCallback_Ind	1300

9.11.4.129unpack_valid_nas_GetCDMANetworkParameters	1300
9.11.4.130unpack_valid_nas_SLQSGetServingSystem	1301
9.11.4.131unpack_valid_nas_SLQSGetSignalStrength	1302
9.11.4.132unpack_valid_nas_SLQSNasGetSigInfo	1302
9.12 pds.h File Reference	1303
9.12.1 Typedef Documentation	1305
9.12.1.1 unpack_pds_ForceXTRADownload_t	1305
9.12.1.2 unpack_pds_PDSInjectTimeReference_t	1305
9.12.1.3 unpack_pds_ResetPDSDData_t	1305
9.12.1.4 unpack_pds_SetEventReportCallback_t	1305
9.12.1.5 unpack_pds_SetPDSDDefaults_t	1305
9.12.1.6 unpack_pds_SetPDSSState_t	1305
9.12.1.7 unpack_pds_SetPortAutomaticTracking_t	1306
9.12.1.8 unpack_pds_SetServiceAutomaticTracking_t	1306
9.12.1.9 unpack_pds_SetXTRAAutomaticDownload_t	1306
9.12.1.10 unpack_pds_SetXTRANetwork_t	1306
9.12.1.11 unpack_pds_SLQSPDSDeterminePosition_t	1306
9.12.1.12 unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t	1306
9.12.1.13 unpack_pds_SLQSPDSInjectPositionData_t	1306
9.12.1.14 unpack_pds_SLQSSetAGPSConfig_t	1306
9.12.1.15 unpack_pds_SLQSSetPositionMethodState_t	1306
9.12.1.16 unpack_pds_StartPDSTrackingSessionExt_t	1306
9.12.1.17 unpack_pds_StopPDSTrackingSession_t	1306
9.12.2 Function Documentation	1306
9.12.2.1 pack_pds_ForceXTRADownload	1306
9.12.2.2 pack_pds_GetPDSDDefaults	1306
9.12.2.3 pack_pds_GetPDSSState	1307
9.12.2.4 pack_pds_GetPortAutomaticTracking	1307
9.12.2.5 pack_pds_GetServiceAutomaticTracking	1307
9.12.2.6 pack_pds_GetXTRAAutomaticDownload	1308
9.12.2.7 pack_pds_GetXTRANetwork	1308
9.12.2.8 pack_pds_GetXTRAVality	1308
9.12.2.9 pack_pds_PDSInjectTimeReference	1309
9.12.2.10 pack_pds_ResetPDSDData	1309
9.12.2.11 pack_pds_SetEventReportCallback	1309
9.12.2.12 pack_pds_SetPDSDDefaults	1310
9.12.2.13 pack_pds_SetPDSSState	1310
9.12.2.14 pack_pds_SetPortAutomaticTracking	1310
9.12.2.15 pack_pds_SetServiceAutomaticTracking	1311
9.12.2.16 pack_pds_SetXTRAAutomaticDownload	1311

9.12.2.17	pack_pds_SetXTRANetwork	1312
9.12.2.18	pack_pds_SLQSGetAGPSConfig	1312
9.12.2.19	pack_pds_SLQSGetGPSSStateInfo	1312
9.12.2.20	pack_pds_SLQSPDSDeterminePosition	1313
9.12.2.21	pack_pds_SLQSPDSInjectAbsoluteTimeReference	1313
9.12.2.22	pack_pds_SLQSPDSInjectPositionData	1313
9.12.2.23	pack_pds_SLQSSetAGPSConfig	1314
9.12.2.24	pack_pds_SLQSSetPositionMethodState	1314
9.12.2.25	pack_pds_StartPDSTrackingSessionExt	1314
9.12.2.26	pack_pds_StopPDSTrackingSession	1315
9.12.2.27	unpack_pds_ForceXTRADownload	1315
9.12.2.28	unpack_pds_GetPDSDefaults	1316
9.12.2.29	unpack_pds_GetPDSSState	1316
9.12.2.30	unpack_pds_GetPortAutomaticTracking	1316
9.12.2.31	unpack_pds_GetServiceAutomaticTracking	1317
9.12.2.32	unpack_pds_GetXTRAAutomaticDownload	1317
9.12.2.33	unpack_pds_GetXTRANetwork	1317
9.12.2.34	unpack_pds_GetXTRAValidity	1318
9.12.2.35	unpack_pds_PDSInjectTimeReference	1318
9.12.2.36	unpack_pds_ResetPDSDData	1318
9.12.2.37	unpack_pds_SetEventReport_Ind	1319
9.12.2.38	unpack_pds_SetEventReportCallback	1319
9.12.2.39	unpack_pds_SetPDSDefaults	1319
9.12.2.40	unpack_pds_SetPDSSState	1320
9.12.2.41	unpack_pds_SetPdsState_Ind	1320
9.12.2.42	unpack_pds_SetPortAutomaticTracking	1320
9.12.2.43	unpack_pds_SetServiceAutomaticTracking	1321
9.12.2.44	unpack_pds_SetXTRAAutomaticDownload	1321
9.12.2.45	unpack_pds_SetXTRANetwork	1321
9.12.2.46	unpack_pds_SLQSGetAGPSConfig	1322
9.12.2.47	unpack_pds_SLQSGetGPSSStateInfo	1322
9.12.2.48	unpack_pds_SLQSPDSDeterminePosition	1322
9.12.2.49	unpack_pds_SLQSPDSInjectAbsoluteTimeReference	1323
9.12.2.50	unpack_pds_SLQSPDSInjectPositionData	1323
9.12.2.51	unpack_pds_SLQSSetAGPSConfig	1323
9.12.2.52	unpack_pds_SLQSSetPositionMethodState	1324
9.12.2.53	unpack_pds_StartPDSTrackingSessionExt	1324
9.12.2.54	unpack_pds_StopPDSTrackingSession	1324
9.13	qaGobiApiTableBandClasses.h File Reference	1325
9.13.1	Detailed Description	1325

9.13.2	Band Classes (Value - Description)	1325
9.13.2.1	LTE Bands	1326
9.14	qaGobiApiTableCallControlReturnReasons.h File Reference	1328
9.14.1	Detailed Description	1328
9.14.2	S1	1328
9.15	qaGobiApiTableCallEndReasons.h File Reference	1329
9.15.1	Detailed Description	1329
9.15.2	Call end reason codes (Code - Reason)	1329
9.15.2.1	Technology-agnostic call end reasons	1329
9.15.2.2	EVDO CDMA 1xEV-DO	1330
9.15.2.3	WCDMA/GSM call end reasons	1330
9.15.2.4	EVDO CDMA 1xEV-DO	1332
9.15.2.5	call end reason type	1333
9.15.2.6	Mobile IP call end reasons (Type=1)	1333
9.15.2.7	Internal call end reasons (Type=2)	1335
9.15.2.8	Call Manager defined call end reasons (Type=3)	1337
9.15.2.9	3GPP specification defined call end reasons (Type=6)	1342
9.15.2.10	PPP call end reasons (Type=7)	1344
9.15.2.11	EHRPD call end reasons (Type=8)	1344
9.15.2.12	IPv6 call end reasons (Type=9)	1345
9.16	qaGobiApiTableCarrierCodes.h File Reference	1345
9.16.1	Detailed Description	1345
9.16.2	Carrier Codes (Number - Carrier)	1346
9.17	qaGobiApiTableCodingScheme.h File Reference	1347
9.17.1	Detailed Description	1347
9.17.2	S1	1347
9.17.2.1	Use of bits 3..0	1347
9.17.3	Coding Group Bits 7..4(0001)	1348
9.17.3.1	use of bits 3..0	1348
9.17.4	Coding Group Bits 7..4(0010)	1348
9.17.4.1	use of bits 3..0	1348
9.17.5	Coding Group Bits 7..4(0011)	1349
9.17.5.1	use of bits 3..0	1349
9.17.6	Coding Group Bits 7..4(01xx)	1349
9.17.6.1	use of bits 3..0	1349
9.17.7	Coding Group Bits 7..4(1001)	1349
9.17.7.1	Reserved coding groups	1349
9.17.8	Coding Group Bits 7..4(1010..1101)	1350
9.17.8.1	Reserved coding groups	1350
9.17.9	Coding Group Bits 7..4(1110)	1350

9.17.9.1	Defined by the WAP Forum	1350
9.17.10	Coding Group Bits 7..4 (1111)	1350
9.17.10.1	Data coding / message handling	1350
9.18	qaGobiApiTableGpsCapabilityCodes.h File Reference	1350
9.18.1	Detailed Description	1350
9.18.2	GPS capability (Value - Capability)	1350
9.19	qaGobiApiTablePowerModes.h File Reference	1350
9.19.1	Detailed Description	1351
9.19.2	Power Modes (Value - Description)	1351
9.20	qaGobiApiTableRadioInterfaces.h File Reference	1351
9.20.1	Detailed Description	1351
9.20.2	Radio interface	1351
9.20.2.1	Technology (Value - Radio Interface Technology)	1351
9.21	qaGobiApiTableRegionCodes.h File Reference	1352
9.21.1	Detailed Description	1352
9.21.2	Region Codes (Code - Region)	1352
9.22	qaGobiApiTableServiceOptions.h File Reference	1352
9.22.1	Detailed Description	1352
9.22.2	Service Option codes (Code - Reason)	1353
9.22.2.1	Description	1353
9.23	qaGobiApiTableSupServiceInfoClasses.h File Reference	1355
9.23.1	Detailed Description	1355
9.23.2	Supplementary Service Information Classes (Value - Service Class)	1355
9.24	qaGobiApiTableSwiAudio.h File Reference	1355
9.24.1	Detailed Description	1355
9.24.2	ACDB Device (Device ID - description)	1355
9.24.3	Physical Interface (Device ID - description - Interface parameters)	1356
9.25	qaGobiApiTableSwiOMADMSessionStatus.h File Reference	1356
9.25.1	Detailed Description	1356
9.25.2	OMA DM Session Status (Session Status - Meaning - Usage)	1356
9.26	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	1357
9.26.1	Detailed Description	1357
9.26.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	1357
9.27	qaGobiApiTableVoiceCallEndReasons.h File Reference	1358
9.27.1	Detailed Description	1358
9.27.2	Voice Call and supplementary services end reason codes (Code - Reason)	1358
9.27.2.1	General	1358
9.27.2.2	service Errors	1360
9.27.2.3	control cause values	1361
9.27.2.4	reject causes	1363

9.27.2.5	reject causes	1363
9.27.2.6	reject causes	1363
9.27.2.7	stratum reject causes	1364
9.27.2.8	reject causes	1364
9.27.2.9	IP end reasons	1364
9.28	qmerrno.h File Reference	1365
9.28.1	Enumeration Type Documentation	1367
9.28.1.1	eQCWWANError	1367
9.28.1.2	qm_wds_ds_profile_extended_err_codes	1372
9.29	qos.h File Reference	1372
9.29.1	Macro Definition Documentation	1374
9.29.1.1	LITEQMI_MAX_QOS_FILTERS	1374
9.29.1.2	LITEQMI_MAX_QOS_FLOW_PER_APN_STATS	1374
9.29.1.3	LITEQMI_MAX_QOS_FLOWS	1374
9.29.2	Function Documentation	1374
9.29.2.1	pack_qos_BindDataPort	1374
9.29.2.2	pack_qos_SLQSQosGetNetworkStatus	1374
9.29.2.3	pack_qos_SLQSQosSwiReadApnExtraParams	1375
9.29.2.4	pack_qos_SLQSQosSwiReadDataStats	1376
9.29.2.5	pack_qos_SLQSSetQosEventCallback	1376
9.29.2.6	unpack_qos_BindDataPort	1377
9.29.2.7	unpack_qos_SLQSQosGetNetworkStatus	1377
9.29.2.8	unpack_qos_SLQSQosSwiReadApnExtraParams	1378
9.29.2.9	unpack_qos_SLQSQosSwiReadDataStats	1378
9.29.2.10	unpack_qos_SLQSSetQosEventCallback	1379
9.29.2.11	unpack_qos_SLQSSetQosEventCallback_ind	1379
9.29.2.12	unpack_qos_SLQSSetQosNWStatusCallback_ind	1380
9.29.2.13	unpack_qos_SLQSSetQosPriEventCallback_ind	1381
9.29.2.14	unpack_qos_SLQSSetQosStatusCallback_ind	1381
9.30	rms.h File Reference	1382
9.30.1	Macro Definition Documentation	1382
9.30.1.1	__LITEQMI_RMS_H__	1382
9.30.2	Function Documentation	1382
9.30.2.1	pack_rms_GetSMSWake	1382
9.30.2.2	pack_rms_SetSMSWake	1383
9.30.2.3	unpack_rms_GetSMSWake	1383
9.30.2.4	unpack_rms_SetSMSWake	1383
9.31	sar.h File Reference	1384
9.31.1	Typedef Documentation	1384
9.31.1.1	unpack_sar_SLQSSetRfSarState_t	1384

9.31.2	Function Documentation	1384
9.31.2.1	pack_sar_SLQSGetRfSarState	1384
9.31.2.2	pack_sar_SLQSSetRfSarState	1385
9.31.2.3	unpack_sar_SLQSGetRfSarState	1385
9.31.2.4	unpack_sar_SLQSSetRfSarState	1385
9.32	sms.h File Reference	1386
9.32.1	Macro Definition Documentation	1389
9.32.1.1	MAX_CDMA_ENC_MO_TXT_MSG_SIZE	1389
9.32.1.2	MAX_MS_TRANSFER_ROUTE_MSG	1389
9.32.1.3	MAX_MSC_ADDRESS_SIZE	1389
9.32.1.4	MAX_MSE_TWS_MSG	1389
9.32.1.5	MAX_SMS_LIST_SIZE	1389
9.32.1.6	MAX_SMS_MESSAGE_SIZE	1389
9.32.1.7	SMS_CONFIG_LEN	1390
9.32.1.8	SMS_MAX_SMS_ROUTES	1390
9.32.1.9	SMS_NUM_OF_SET	1390
9.32.1.10	SMSC_TYPE_LEN	1390
9.32.2	Enumeration Type Documentation	1390
9.32.2.1	eqmiCbkJSetStatus	1390
9.32.3	Function Documentation	1390
9.32.3.1	pack_sms_GetSMSCAddress	1390
9.32.3.2	pack_sms_SaveSMS	1390
9.32.3.3	pack_sms_SendSMS	1391
9.32.3.4	pack_sms_SetNewSMSCallback	1391
9.32.3.5	pack_sms_SetSMSCAddress	1391
9.32.3.6	pack_sms_SLQSDelateSMS	1392
9.32.3.7	pack_sms_SLQSGetIndicationRegister	1392
9.32.3.8	pack_sms_SLQSGetMessageWaiting	1392
9.32.3.9	pack_sms_SLQSGetSMS	1393
9.32.3.10	pack_sms_SLQSGetSmsBroadcastConfig	1393
9.32.3.11	pack_sms_SLQSGetSMSList	1393
9.32.3.12	pack_sms_SLQSGetTransLayerInfo	1394
9.32.3.13	pack_sms_SLQSGetTransNWRegInfo	1394
9.32.3.14	pack_sms_SLQSModifySMSStatus	1394
9.32.3.15	pack_sms_SLQSSendAsyncSMS	1395
9.32.3.16	pack_sms_SLQSSetIndicationRegister	1395
9.32.3.17	pack_sms_SLQSSetSmsBroadcastActivation	1396
9.32.3.18	pack_sms_SLQSSetSmsBroadcastConfig	1396
9.32.3.19	pack_sms_SLQSSetSmsStorage	1396
9.32.3.20	pack_sms_SLQSSmsGetMaxStorageSize	1397

9.32.3.21	pack_sms_SLQSSmsGetMessageProtocol	1397
9.32.3.22	pack_sms_SLQSSmsSetRoutes	1397
9.32.3.23	pack_sms_SLQSSwiGetSMSStorage	1398
9.32.3.24	unpack_sms_GetSMSCAddress	1398
9.32.3.25	unpack_sms_SaveSMS	1398
9.32.3.26	unpack_sms_SendSMS	1399
9.32.3.27	unpack_sms_SetNewSMSCallback	1399
9.32.3.28	unpack_sms_SetNewSMSCallback_ind	1399
9.32.3.29	unpack_sms_SetSMSCAddress	1400
9.32.3.30	unpack_sms_SLQSDeleteSMS	1400
9.32.3.31	unpack_sms_SLQSGetIndicationRegister	1401
9.32.3.32	unpack_sms_SLQSGetMessageWaiting	1401
9.32.3.33	unpack_sms_SLQSGetSMS	1401
9.32.3.34	unpack_sms_SLQSGetSmsBroadcastConfig	1402
9.32.3.35	unpack_sms_SLQSGetSMSList	1402
9.32.3.36	unpack_sms_SLQSGetTransLayerInfo	1402
9.32.3.37	unpack_sms_SLQSGetTransNWRegInfo	1403
9.32.3.38	unpack_sms_SLQSModifySMSStatus	1403
9.32.3.39	unpack_sms_SLQSNWRegInfoCallback_ind	1403
9.32.3.40	unpack_sms_SLQSSendAsyncSMS	1404
9.32.3.41	unpack_sms_SLQSSetIndicationRegister	1404
9.32.3.42	unpack_sms_SLQSSetSmsBroadcastActivation	1404
9.32.3.43	unpack_sms_SLQSSetSmsBroadcastConfig	1405
9.32.3.44	unpack_sms_SLQSSetSmsStorage	1405
9.32.3.45	unpack_sms_SLQSSmsGetMaxStorageSize	1405
9.32.3.46	unpack_sms_SLQSSmsGetMessageProtocol	1406
9.32.3.47	unpack_sms_SLQSSmsSetRoutes	1406
9.32.3.48	unpack_sms_SLQSSwiGetSMSStorage	1407
9.32.3.49	unpack_sms_SLQSTransLayerInfoCallback_ind	1407
9.32.3.50	unpack_sms_SLQSWmsAsyncRawSendCallBack_ind	1407
9.32.3.51	unpack_sms_SLQSWmsMemoryFullCallBack_ind	1408
9.32.3.52	unpack_sms_SLQSWmsMessageWaitingCallBack_ind	1408
9.33	swiaudio.h File Reference	1408
9.33.1	Macro Definition Documentation	1410
9.33.1.1	SWIAUDIO_MAX_LEN_IFACE_TABLE	1410
9.33.2	Typedef Documentation	1410
9.33.2.1	unpack_swiaudio_SLQSSetM2MAudioAVCFG_t	1410
9.33.2.2	unpack_swiaudio_SLQSSetM2MAudioLPBK_t	1410
9.33.2.3	unpack_swiaudio_SLQSSetM2MAudioNVDef_t	1410
9.33.2.4	unpack_swiaudio_SLQSSetM2MAudioProfile_t	1410

9.33.2.5	unpack_swiaudio_SLQSSetM2MAudioVolume_t	1410
9.33.2.6	unpack_swiaudio_SLQSSetM2MAVMute_t	1410
9.33.2.7	unpack_swiaudio_SLQSSetM2MSprkGain_t	1410
9.33.3	Function Documentation	1410
9.33.3.1	pack_swiaudio_SLQSGetM2MAudioProfile	1410
9.33.3.2	pack_swiaudio_SLQSGetM2MAudioVolume	1411
9.33.3.3	pack_swiaudio_SLQSGetM2MAVMute	1411
9.33.3.4	pack_swiaudio_SLQSGetM2MSprkGain	1411
9.33.3.5	pack_swiaudio_SLQSSetM2MAudioAVCFG	1412
9.33.3.6	pack_swiaudio_SLQSSetM2MAudioLPBK	1412
9.33.3.7	pack_swiaudio_SLQSSetM2MAudioNVDef	1412
9.33.3.8	pack_swiaudio_SLQSSetM2MAudioProfile	1413
9.33.3.9	pack_swiaudio_SLQSSetM2MAudioVolume	1413
9.33.3.10	pack_swiaudio_SLQSSetM2MAVMute	1414
9.33.3.11	pack_swiaudio_SLQSSetM2MSprkGain	1414
9.33.3.12	unpack_swiaudio_SLQSGetM2MAudioProfile	1414
9.33.3.13	unpack_swiaudio_SLQSGetM2MAudioVolume	1415
9.33.3.14	unpack_swiaudio_SLQSGetM2MAVMute	1415
9.33.3.15	unpack_swiaudio_SLQSGetM2MSprkGain	1415
9.33.3.16	unpack_swiaudio_SLQSSetM2MAudioAVCFG	1416
9.33.3.17	unpack_swiaudio_SLQSSetM2MAudioLPBK	1416
9.33.3.18	unpack_swiaudio_SLQSSetM2MAudioNVDef	1416
9.33.3.19	unpack_swiaudio_SLQSSetM2MAudioProfile	1417
9.33.3.20	unpack_swiaudio_SLQSSetM2MAudioVolume	1417
9.33.3.21	unpack_swiaudio_SLQSSetM2MAVMute	1417
9.33.3.22	unpack_swiaudio_SLQSSetM2MSprkGain	1418
9.34	swiavms.h File Reference	1418
9.34.1	Macro Definition Documentation	1420
9.34.1.1	LITEQMI_MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH	1420
9.34.1.2	LITEQMI_MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH	1420
9.34.1.3	LITEQMI_MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH	1420
9.34.1.4	LITEQMI_MAX_SWIOMA_STR_LEN	1420
9.34.1.5	MAX_AVMS_SETTINGS_RETRY_TIMER_NUMBER	1420
9.34.1.6	MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH	1420
9.34.1.7	MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH	1420
9.34.1.8	MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH	1420
9.34.1.9	MAX_PACK_SWI_AVMS_SESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH	1420

9.34.1.10	MAX_PACK_SWI_AVMS_SET_SETTING_APN_LENGTH	1420
9.34.1.11	MAX_PACK_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS	1420
9.34.1.12	MAX_PACK_SWI_AVMS_SET_SETTING_PWD_LENGTH	1420
9.34.1.13	MAX_PACK_SWI_AVMS_SET_SETTING_UNAME_LENGTH	1420
9.34.2	Typedef Documentation	1420
9.34.2.1	unpack_swiaavms_SLQSAVMSStopSession_avc2_t	1420
9.34.3	Function Documentation	1420
9.34.3.1	pack_swiaavms_SLQSAVMSSetSettings	1421
9.34.3.2	pack_swiaavms_SLQSAVMSSetSettings_v2	1421
9.34.3.3	pack_swiaavms_SLQSAVMSSendSelection	1422
9.34.3.4	pack_swiaavms_SLQSAVMSSessionGetInfo	1422
9.34.3.5	pack_swiaavms_SLQSAvmsSetEventReport	1423
9.34.3.6	pack_swiaavms_SLQSAVMSSetSettings	1423
9.34.3.7	pack_swiaavms_SLQSAVMSSetSettings_v2	1424
9.34.3.8	pack_swiaavms_SLQSAVMSSetSettingsNoAutoRebootField	1425
9.34.3.9	pack_swiaavms_SLQSAVMSStartSession	1425
9.34.3.10	pack_swiaavms_SLQSAVMSStopSession	1426
9.34.3.11	unpack_swiaavms_SLQSAVMSEventReportInd	1427
9.34.3.12	unpack_swiaavms_SLQSAVMSSetSettings	1427
9.34.3.13	unpack_swiaavms_SLQSAVMSSetSettings_v2	1428
9.34.3.14	unpack_swiaavms_SLQSAVMSSendSelection	1428
9.34.3.15	unpack_swiaavms_SLQSAVMSSessionGetInfo	1428
9.34.3.16	unpack_swiaavms_SLQSAvmsSetEventReport	1429
9.34.3.17	unpack_swiaavms_SLQSAVMSSetSettings	1429
9.34.3.18	unpack_swiaavms_SLQSAVMSSetSettings_v2	1430
9.34.3.19	unpack_swiaavms_SLQSAVMSStartSession	1430
9.34.3.20	unpack_swiaavms_SLQSAVMSStopSession	1431
9.34.3.21	unpack_swiaavms_SLQSAVMSStopSession_avc2	1431
9.35	SwiDataTypes.h File Reference	1432
9.35.1	Detailed Description	1432
9.35.2	Macro Definition Documentation	1432
9.35.2.1	MEMSET_STATIC_OUTPUT_STRUCT	1432
9.35.2.2	QMI_NO_LTE_FW_SUPPORT	1433
9.35.2.3	QMI_TLV_PLACEHOLDER	1433
9.35.2.4	SWI_API	1433
9.35.2.5	UNUSEDPARAM	1433
9.35.3	Typedef Documentation	1433
9.35.3.1	BOOL	1433
9.35.3.2	BYTE	1433
9.35.3.3	CHAR	1433

9.35.3.4	FLOAT	1433
9.35.3.5	INT32	1433
9.35.3.6	INT8	1433
9.35.3.7	LPCSTR	1433
9.35.3.8	qmuint16	1433
9.35.3.9	qmulong	1433
9.35.3.10	SHORT	1433
9.35.3.11	ULONG	1433
9.35.3.12	ULONGLONG	1433
9.35.3.13	USHORT	1433
9.35.3.14	WORD	1433
9.36	swidms.h File Reference	1433
9.36.1	Typedef Documentation	1435
9.36.1.1	unpack_swidms_SLQSSwiDmsSetUsbNetNum_t	1435
9.36.2	Function Documentation	1435
9.36.2.1	pack_swidms_SLQSSwiDmsGetHWWatchdog	1435
9.36.2.2	pack_swidms_SLQSSwiDmsGetMTU	1435
9.36.2.3	pack_swidms_SLQSSwiDmsGetSecureInfo	1435
9.36.2.4	pack_swidms_SLQSSwiDmsGetUsbComp	1436
9.36.2.5	pack_swidms_SLQSSwiDmsGetUsbNetNum	1436
9.36.2.6	pack_swidms_SLQSSwiDmsSetHWWatchdog	1436
9.36.2.7	pack_swidms_SLQSSwiDmsSetMTU	1437
9.36.2.8	pack_swidms_SLQSSwiDmsSetUsbComp	1437
9.36.2.9	pack_swidms_SLQSSwiDmsSetUsbNetNum	1437
9.36.2.10	unpack_swidms_SLQSSwiDmsGetHWWatchdog	1438
9.36.2.11	unpack_swidms_SLQSSwiDmsGetMTU	1438
9.36.2.12	unpack_swidms_SLQSSwiDmsGetSecureInfo	1439
9.36.2.13	unpack_swidms_SLQSSwiDmsGetUsbComp	1439
9.36.2.14	unpack_swidms_SLQSSwiDmsGetUsbNetNum	1439
9.36.2.15	unpack_swidms_SLQSSwiDmsSetHWWatchdog	1440
9.36.2.16	unpack_swidms_SLQSSwiDmsSetMTU	1440
9.36.2.17	unpack_swidms_SLQSSwiDmsSetUsbComp	1440
9.36.2.18	unpack_swidms_SLQSSwiDmsSetUsbNetNum	1441
9.37	swiloc.h File Reference	1441
9.37.1	Typedef Documentation	1441
9.37.1.1	unpack_swiloc_SwiLocSetAutoStart_t	1441
9.37.2	Function Documentation	1442
9.37.2.1	pack_swiloc_SwiLocGetAutoStart	1442
9.37.2.2	pack_swiloc_SwiLocSetAutoStart	1442
9.37.2.3	unpack_swiloc_SwiLocGetAutoStart	1442

9.37.2.4	unpack_swiloc_SwiLocSetAutoStart	1443
9.38	swioma.h File Reference	1443
9.38.1	Macro Definition Documentation	1444
9.38.1.1	LITEQMI_MAX_SWIOMA_STR_LEN	1444
9.38.2	Typedef Documentation	1444
9.38.2.1	unpack_swioma_SLQSOMADMAAlertCallback_t	1444
9.38.2.2	unpack_swioma_SLQSOMADMCancelSession_t	1444
9.38.2.3	unpack_swioma_SLQSOMADMSendSelection_t	1444
9.38.2.4	unpack_swioma_SLQSOMADMSetSettings_t	1444
9.38.3	Function Documentation	1444
9.38.3.1	pack_swioma_SLQSOMADMAAlertCallback	1444
9.38.3.2	pack_swioma_SLQSOMADMCancelSession	1445
9.38.3.3	pack_swioma_SLQSOMADMGetSessionInfo	1446
9.38.3.4	pack_swioma_SLQSOMADMGetSettings	1446
9.38.3.5	pack_swioma_SLQSOMADMSendSelection	1447
9.38.3.6	pack_swioma_SLQSOMADMSetSettings	1448
9.38.3.7	pack_swioma_SLQSOMADMStartSession	1448
9.38.3.8	unpack_swioma_SLQSOMADMAAlertCallback	1449
9.38.3.9	unpack_swioma_SLQSOMADMAAlertCallback_ind	1449
9.38.3.10	unpack_swioma_SLQSOMADMCancelSession	1450
9.38.3.11	unpack_swioma_SLQSOMADMGetSessionInfo	1450
9.38.3.12	unpack_swioma_SLQSOMADMGetSettings	1451
9.38.3.13	unpack_swioma_SLQSOMADMSendSelection	1451
9.38.3.14	unpack_swioma_SLQSOMADMSetSettings	1452
9.38.3.15	unpack_swioma_SLQSOMADMStartSession	1452
9.39	swiomaext.h File Reference	1453
9.39.1	Macro Definition Documentation	1453
9.39.1.1	LITE_SWIOMAEXT_MAX_UCS2_DATA_LEN	1453
9.39.1.2	LITEQMI_MAX_SWIOMA_STR_LEN	1453
9.39.2	Typedef Documentation	1453
9.39.2.1	unpack_swioma_SLQSOMADMCancelSessionExt_t	1454
9.39.2.2	unpack_swioma_SLQSOMADMSendSelectionExt_t	1454
9.39.2.3	unpack_swioma_SLQSOMADMSetSettingsExt_t	1454
9.39.2.4	unpack_swioma_SLQSOMADMStartSessionExt_t	1454
9.39.3	Function Documentation	1454
9.39.3.1	pack_swioma_SLQSOMADMCancelSessionExt	1454
9.39.3.2	pack_swioma_SLQSOMADMGetSessionInfoExt	1454
9.39.3.3	pack_swioma_SLQSOMADMSendSelectionExt	1455
9.39.3.4	pack_swioma_SLQSOMADMSetSettingsExt	1456
9.39.3.5	pack_swioma_SLQSOMADMStartSessionExt	1456

9.39.3.6	unpack_swioma_SLQSOMADMCancelSessionExt	1457
9.39.3.7	unpack_swioma_SLQSOMADMGetSessionInfoExt	1457
9.39.3.8	unpack_swioma_SLQSOMADMSendSelectionExt	1458
9.39.3.9	unpack_swioma_SLQSOMADMSetSettingsExt	1458
9.39.3.10	unpack_swioma_SLQSOMADMStartSessionExt	1459
9.40	switype_256bit.h File Reference	1459
9.40.1	Macro Definition Documentation	1460
9.40.1.1	SWI_UINT256_BIT_OFFSET	1460
9.40.1.2	SWI_UINT256_BITS_PER_WORD	1460
9.40.1.3	SWI_UINT256_INT_VALUE	1460
9.40.1.4	SWI_UINT256_WORD_COUNT	1460
9.40.1.5	SWI_UINT256_WORD_OFFSET	1460
9.40.2	Function Documentation	1460
9.40.2.1	swi_uint256_clear_bit	1460
9.40.2.2	swi_uint256_get_bit	1460
9.40.2.3	swi_uint256_print_mask	1460
9.40.2.4	swi_uint256_set_bit	1460
9.41	SWIWWANCMAPI.h File Reference	1461
9.42	tmd.h File Reference	1461
9.42.1	Macro Definition Documentation	1461
9.42.1.1	MAX_MITIGATION_DEV_ID_LEN	1461
9.42.1.2	MAX_MITIGATION_DEV_LIST_LEN	1461
9.42.1.3	TMD_MAX_DEV_LIST	1461
9.42.2	Function Documentation	1461
9.42.2.1	pack_tmd_SLQSTmdDeRegNotMitigationLvl	1462
9.42.2.2	pack_tmd_SLQSTmdGetMitigationDevList	1462
9.42.2.3	pack_tmd_SLQSTmdGetMitigationLvl	1462
9.42.2.4	pack_tmd_SLQSTmdRegNotMitigationLvl	1463
9.42.2.5	unpack_tmd_SLQSTmdDeRegNotMitigationLvl	1463
9.42.2.6	unpack_tmd_SLQSTmdGetMitigationDevList	1463
9.42.2.7	unpack_tmd_SLQSTmdGetMitigationLvl	1464
9.42.2.8	unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind	1464
9.42.2.9	unpack_tmd_SLQSTmdRegNotMitigationLvl	1464
9.43	uim.h File Reference	1465
9.43.1	Macro Definition Documentation	1468
9.43.1.1	MAX_ATR_LENGTH	1468
9.43.1.2	MAX_DESCRIPTION_LENGTH	1468
9.43.1.3	MAX_ICCID_LENGTH	1468
9.43.1.4	MAX_NO_OF_APPLICATIONS	1468
9.43.1.5	MAX_NO_OF_SLOTS	1468

9.43.1.6	MAX_PHY_SLOTS_INFO	1468
9.43.1.7	MAX_SLOTS_STATUS	1468
9.43.1.8	UIM_MAX_ACTIVE_PERS_FEATURES	1468
9.43.1.9	UIM_MAX_CONTENT_LENGTH	1468
9.43.1.10	UIM_MAX_DESCRIPTION_LENGTH	1468
9.43.1.11	UIM_MAX_NO_OF_APPLICATIONS	1468
9.43.1.12	UIM_MAX_NO_OF_SLOTS	1469
9.43.1.13	UIM_UINT8_MAX_STRING_SZ	1469
9.43.2	Typedef Documentation	1469
9.43.2.1	unpack_uim_SLQSUIMPowerDown_t	1469
9.43.2.2	unpack_uim_SLQSUIMPowerUp_t	1469
9.43.2.3	unpack_uim_SLQSUIMRefreshComplete_t	1469
9.43.2.4	unpack_uim_SLQSUIMRefreshOK_t	1469
9.43.2.5	unpack_uim_SLQSUIMRefreshRegister_t	1469
9.43.2.6	unpack_uim_SLQSUIMReset_t	1469
9.43.2.7	unpack_uim_SLQSUIMSwitchSlot_t	1469
9.43.3	Function Documentation	1469
9.43.3.1	pack_uim_ChangePin	1469
9.43.3.2	pack_uim_GetCardStatus	1469
9.43.3.3	pack_uim_ReadTransparent	1470
9.43.3.4	pack_uim_SetPinProtection	1470
9.43.3.5	pack_uim_SLQSUIMAuthenticate	1470
9.43.3.6	pack_uim_SLQSUIMDepersonalization	1471
9.43.3.7	pack_uim_SLQSUIMEventRegister	1471
9.43.3.8	pack_uim_SLQSUIMGetConfiguration	1472
9.43.3.9	pack_uim_SLQSUIMGetFileAttributes	1472
9.43.3.10	pack_uim_SLQSUIMGetSlotsStatus	1472
9.43.3.11	pack_uim_SLQSUIMPowerDown	1473
9.43.3.12	pack_uim_SLQSUIMPowerUp	1473
9.43.3.13	pack_uim_SLQSUIMRefreshComplete	1473
9.43.3.14	pack_uim_SLQSUIMRefreshGetLastEvent	1474
9.43.3.15	pack_uim_SLQSUIMRefreshOK	1474
9.43.3.16	pack_uim_SLQSUIMRefreshRegister	1474
9.43.3.17	pack_uim_SLQSUIMReset	1475
9.43.3.18	pack_uim_SLQSUIMSwitchSlot	1475
9.43.3.19	pack_uim_UnblockPin	1476
9.43.3.20	pack_uim_VerifyPin	1476
9.43.3.21	unpack_uim_ChangePin	1476
9.43.3.22	unpack_uim_GetCardStatus	1477
9.43.3.23	unpack_uim_GetCardStatusV2	1477

9.43.3.24	unpack_uim_ReadTransparent	1477
9.43.3.25	unpack_uim_SetPinProtection	1478
9.43.3.26	unpack_uim_SetUimSlotStatusChangeCallback_ind	1478
9.43.3.27	unpack_uim_SLQSUIMAuthenticate	1478
9.43.3.28	unpack_uim_SLQSUIMDepersonalization	1479
9.43.3.29	unpack_uim_SLQSUIMEventRegister	1479
9.43.3.30	unpack_uim_SLQSUIMGetConfiguration	1480
9.43.3.31	unpack_uim_SLQSUIMGetFileAttributes	1480
9.43.3.32	unpack_uim_SLQSUIMGetSlotsStatus	1480
9.43.3.33	unpack_uim_SLQSUIMGetSlotsStatusV2	1481
9.43.3.34	unpack_uim_SLQSUIMPowerDown	1481
9.43.3.35	unpack_uim_SLQSUIMPowerUp	1481
9.43.3.36	unpack_uim_SLQSUIMRefreshCallback_Ind	1482
9.43.3.37	unpack_uim_SLQSUIMRefreshComplete	1482
9.43.3.38	unpack_uim_SLQSUIMRefreshGetLastEvent	1482
9.43.3.39	unpack_uim_SLQSUIMRefreshOK	1483
9.43.3.40	unpack_uim_SLQSUIMRefreshRegister	1483
9.43.3.41	unpack_uim_SLQSUIMReset	1483
9.43.3.42	unpack_uim_SLQSUIMSetStatusChangeCallBack_ind	1484
9.43.3.43	unpack_uim_SLQSUIMSwitchSlot	1484
9.43.3.44	unpack_uim_UnblockPin	1484
9.43.3.45	unpack_uim_UnblockPinV2	1485
9.43.3.46	unpack_uim_VerifyPin	1485
9.44	voice.h File Reference	1485
9.44.1	Detailed Description	1491
9.44.2	S1	1491
9.44.3	Macro Definition Documentation	1491
9.44.3.1	BARRING_PASSWORD_LENGTH	1491
9.44.3.2	MAX_VOICE_CALL_NO_LEN	1491
9.44.3.3	MAX_VOICE_DESCRIPTION_LENGTH	1491
9.44.3.4	MAXVOICEUSSDLENGTH	1491
9.44.3.5	VOICE_MAX_NO_OF_CALLS	1491
9.44.4	Typedef Documentation	1491
9.44.4.1	unpack_voice_AnswerUSSD_t	1491
9.44.4.2	unpack_voice_CancelUSSD_t	1491
9.44.4.3	unpack_voice_OriginateUSSD_t	1491
9.44.4.4	unpack_voice_SLQSVoiceALSSelectLine_t	1491
9.44.4.5	unpack_voice_SLQSVoiceALSSetLineSwitching_t	1491
9.44.4.6	unpack_voice_SLQSVoiceBindSubscription_t	1491
9.44.4.7	unpack_voice_SLQSVoiceIndicationRegister_t	1491

9.44.4.8	unpack_voice_SLQSVoiceOrigUSSDNoWait_t	1491
9.44.4.9	unpack_voice_SLQSVoiceSetPreferredPrivacy_t	1491
9.44.5	Enumeration Type Documentation	1491
9.44.5.1	liteServiceClassInformation	1492
9.44.6	Function Documentation	1492
9.44.6.1	pack_voice_AnswerUSSD	1492
9.44.6.2	pack_voice_CancelUSSD	1492
9.44.6.3	pack_voice_OriginateUSSD	1493
9.44.6.4	pack_voice_SLQSVoiceOriginateUSSD	1493
9.44.6.5	pack_voice_SLQSVoiceALSSelectLine	1493
9.44.6.6	pack_voice_SLQSVoiceALSSetLineSwitching	1494
9.44.6.7	pack_voice_SLQSVoiceAnswerCall	1494
9.44.6.8	pack_voice_SLQSVoiceBindSubscription	1494
9.44.6.9	pack_voice_SLQSVoiceBurstDTMF	1495
9.44.6.10	pack_voice_SLQSVoiceDialCall	1495
9.44.6.11	pack_voice_SLQSVoiceEndCall	1496
9.44.6.12	pack_voice_SLQSVoiceGetAllCallInfo	1496
9.44.6.13	pack_voice_SLQSVoiceGetCallBarring	1496
9.44.6.14	pack_voice_SLQSVoiceGetCallForwardingStatus	1497
9.44.6.15	pack_voice_SLQSVoiceGetCallInfo	1497
9.44.6.16	pack_voice_SLQSVoiceGetCallWaiting	1497
9.44.6.17	pack_voice_SLQSVoiceGetCLIP	1498
9.44.6.18	pack_voice_SLQSVoiceGetCLIR	1498
9.44.6.19	pack_voice_SLQSVoiceGetCNAP	1498
9.44.6.20	pack_voice_SLQSVoiceGetCOLP	1499
9.44.6.21	pack_voice_SLQSVoiceGetCOLR	1499
9.44.6.22	pack_voice_SLQSVoiceGetConfig	1499
9.44.6.23	pack_voice_SLQSVoiceIndicationRegister	1500
9.44.6.24	pack_voice_SLQSVoiceManageCalls	1500
9.44.6.25	pack_voice_SLQSVoiceOrigUSSDNoWait	1500
9.44.6.26	pack_voice_SLQSVoiceSendFlash	1501
9.44.6.27	pack_voice_SLQSVoiceSetCallBarringPassword	1501
9.44.6.28	pack_voice_SLQSVoiceSetConfig	1502
9.44.6.29	pack_voice_SLQSVoiceSetPreferredPrivacy	1502
9.44.6.30	pack_voice_SLQSVoiceSetSUPSService	1502
9.44.6.31	pack_voice_SLQSVoiceStartContDTMF	1503
9.44.6.32	pack_voice_SLQSVoiceStopContDTMF	1503
9.44.6.33	unpack_voice_allCallStatusCallback_ind	1503
9.44.6.34	unpack_voice_AnswerUSSD	1504
9.44.6.35	unpack_voice_CancelUSSD	1504

9.44.6.36 unpack_voice_DTMFEventCallback_ind	1504
9.44.6.37 unpack_voice_OriginateUSSD	1505
9.44.6.38 unpack_voice_OTASPStatusCallback_ind	1505
9.44.6.39 unpack_voice_SLQSOriginateUSSD	1506
9.44.6.40 unpack_voice_SLQSVoiceALSSelectLine	1506
9.44.6.41 unpack_voice_SLQSVoiceALSSetLineSwitching	1506
9.44.6.42 unpack_voice_SLQSVoiceAnswerCall	1507
9.44.6.43 unpack_voice_SLQSVoiceBindSubscription	1507
9.44.6.44 unpack_voice_SLQSVoiceBurstDTMF	1507
9.44.6.45 unpack_voice_SLQSVoiceDialCall	1508
9.44.6.46 unpack_voice_SLQSVoiceEndCall	1508
9.44.6.47 unpack_voice_SLQSVoiceGetAllCallInfo	1508
9.44.6.48 unpack_voice_SLQSVoiceGetCallBarring	1509
9.44.6.49 unpack_voice_SLQSVoiceGetCallForwardingStatus	1509
9.44.6.50 unpack_voice_SLQSVoiceGetCallInfo	1509
9.44.6.51 unpack_voice_SLQSVoiceGetCallWaiting	1510
9.44.6.52 unpack_voice_SLQSVoiceGetCLIP	1510
9.44.6.53 unpack_voice_SLQSVoiceGetCLIR	1510
9.44.6.54 unpack_voice_SLQSVoiceGetCNAP	1511
9.44.6.55 unpack_voice_SLQSVoiceGetCOLP	1511
9.44.6.56 unpack_voice_SLQSVoiceGetCOLR	1511
9.44.6.57 unpack_voice_SLQSVoiceGetConfig	1512
9.44.6.58 unpack_voice_SLQSVoiceIndicationRegister	1512
9.44.6.59 unpack_voice_SLQSVoiceManageCalls	1512
9.44.6.60 unpack_voice_SLQSVoiceOrigUSSDNoWait	1513
9.44.6.61 unpack_voice_SLQSVoiceSendFlash	1513
9.44.6.62 unpack_voice_SLQSVoiceSetCallBarringPassword	1513
9.44.6.63 unpack_voice_SLQSVoiceSetConfig	1514
9.44.6.64 unpack_voice_SLQSVoiceSetPreferredPrivacy	1514
9.44.6.65 unpack_voice_SLQSVoiceSetSUPSService	1514
9.44.6.66 unpack_voice_SLQSVoiceStartContDTMF	1515
9.44.6.67 unpack_voice_SLQSVoiceStopContDTMF	1515
9.44.6.68 unpack_voice_SLQSVoiceSUPSCallback_ind	1515
9.44.6.69 unpack_voice_SUPSNotificationCallback_ind	1516
9.44.6.70 unpack_voice_USSDNotificationCallback_ind	1516
9.44.6.71 unpack_voice_VoiceInfoRecCallback_ind	1517
9.44.6.72 unpack_voice_voicePrivacyChangeCallback_ind	1517
9.45 wds.h File Reference	1517
9.45.1 Detailed Description	1524
9.45.2 SO Mask	1524

9.45.3	RAT Mask	1525
9.45.4	Macro Definition Documentation	1525
9.45.4.1	BYT_STAT_STAT_MASK	1525
9.45.4.2	IPV6_ADDRESS_ARRAY_SIZE	1525
9.45.4.3	LITE_MAX_PCOID_LIST	1525
9.45.4.4	LITE_MAX_PDN_THROTTLE_TIMER	1525
9.45.4.5	MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	1526
9.45.4.6	MAX_WDS_NAME_ARRAY_SIZE	1526
9.45.4.7	PACK_WDS_IPV4	1526
9.45.4.8	PACK_WDS_IPV6	1526
9.45.4.9	WDS_DHCP_MAX_NUM_OPTIONS	1526
9.45.4.10	WDS_DHCP_OPTION_DATA_BUF_SIZE	1526
9.45.4.11	WDS_PROFILE_3GPP	1526
9.45.4.12	WDS_PROFILE_3GPP2	1526
9.45.4.13	WDS_TFTID_SOURCE_IP_SIZE	1526
9.45.5	Typedef Documentation	1526
9.45.5.1	unpack_wds_DHCPv4ClientLeaseChange_t	1526
9.45.5.2	unpack_wds_RMTransferStatistics_ind_t	1526
9.45.5.3	unpack_wds_SetAutoconnect_t	1526
9.45.5.4	unpack_wds_SetDefaultProfile_t	1526
9.45.5.5	unpack_wds_SetDefaultProfileNum_t	1526
9.45.5.6	unpack_wds_SetMobileIP_t	1526
9.45.5.7	unpack_wds_SetMobileIPParameters_t	1526
9.45.5.8	unpack_wds_SetMuxID_t	1526
9.45.5.9	unpack_wds_SLQSResetPacketStatics_t	1526
9.45.5.10	unpack_wds_SLQSSet3GPPConfigItem_t	1526
9.45.5.11	unpack_wds_SLQSSetWdsEventCallback_t	1526
9.45.5.12	unpack_wds_SLQSSetDHCPv4ClientConfig_t	1526
9.45.5.13	unpack_wds_SLQSSetLoopback_t	1526
9.45.5.14	unpack_wds_SLQSStopDataSession_t	1526
9.45.5.15	unpack_wds_SLQSWdsGoActive_t	1526
9.45.5.16	unpack_wds_SLQSWdsGoDormant_t	1526
9.45.5.17	unpack_wds_SLQSWdsSetEventReport_t	1526
9.45.5.18	UnpackQmiProfileInfo	1527
9.45.5.19	UnpackQmiProfileInfoV2	1527
9.45.6	Enumeration Type Documentation	1527
9.45.6.1	lteQmiDataBearerMasks	1527
9.45.7	Function Documentation	1527
9.45.7.1	pack_wds_DHCPv4ClientLeaseChange	1527
9.45.7.2	pack_wds_GetAutoconnect	1528

9.45.7.3	pack_wds_GetByteTotals	1528
9.45.7.4	pack_wds_GetConnectionRate	1529
9.45.7.5	pack_wds_GetDataBearerTechnology	1529
9.45.7.6	pack_wds_GetDefaultProfile	1529
9.45.7.7	pack_wds_GetDefaultProfileNum	1530
9.45.7.8	pack_wds_GetDefaultProfileV2	1530
9.45.7.9	pack_wds_GetDormancyState	1531
9.45.7.10	pack_wds_GetLastMobileIPError	1531
9.45.7.11	pack_wds_GetMobileIP	1531
9.45.7.12	pack_wds_GetMobileIPProfile	1532
9.45.7.13	pack_wds_GetPacketStatistics	1532
9.45.7.14	pack_wds_GetPacketStatus	1533
9.45.7.15	pack_wds_GetSessionDuration	1533
9.45.7.16	pack_wds_GetSessionDurationV2	1534
9.45.7.17	pack_wds_GetSessionState	1534
9.45.7.18	pack_wds_RMSetTransferStatistics	1534
9.45.7.19	pack_wds_SetAutoconnect	1535
9.45.7.20	pack_wds_SetDefaultProfile	1535
9.45.7.21	pack_wds_SetDefaultProfileNum	1536
9.45.7.22	pack_wds_SetMobileIP	1536
9.45.7.23	pack_wds_SetMobileIPParameters	1536
9.45.7.24	pack_wds_SetMobileIPProfile	1537
9.45.7.25	pack_wds_SetMuxID	1537
9.45.7.26	pack_wds_SLQSCreateProfile	1538
9.45.7.27	pack_wds_SLQSDeleteProfile	1538
9.45.7.28	pack_wds_SLQSGet3GPPConfigItem	1538
9.45.7.29	pack_wds_SLQSGetCurrDataSystemStat	1539
9.45.7.30	pack_wds_SLQSGetCurrentChannelRate	1539
9.45.7.31	pack_wds_SLQSGetDataBearerTechnology	1540
9.45.7.32	pack_wds_SLQSGetDUNCallInfo	1540
9.45.7.33	pack_wds_SLQSGetProfileSettings	1541
9.45.7.34	pack_wds_SLQSGetProfileSettingsV2	1541
9.45.7.35	pack_wds_SLQSGetRuntimeSettings	1542
9.45.7.36	pack_wds_SLQSModifyProfile	1542
9.45.7.37	pack_wds_SLQSResetPacketStatics	1542
9.45.7.38	pack_wds_SLQSSet3GPPConfigItem	1543
9.45.7.39	pack_wds_SLQSSetIPFamilyPreference	1543
9.45.7.40	pack_wds_SLQSSetWdsEventCallback	1544
9.45.7.41	pack_wds_SLQSSetDHCPv4ClientConfig	1544
9.45.7.42	pack_wds_SLQSSetLoopback	1544

9.45.7.43 pack_wds_SLQSSetDHCPv4ClientConfig	1545
9.45.7.44 pack_wds_SLQSSetLoopback	1545
9.45.7.45 pack_wds_SLQSStartDataSession	1546
9.45.7.46 pack_wds_SLQSStopDataSession	1546
9.45.7.47 pack_wds_SLQSWdsGoActive	1546
9.45.7.48 pack_wds_SLQSWdsGoDormant	1547
9.45.7.49 pack_wds_SLQSWdsSetEventReport	1547
9.45.7.50 pack_wds_SLQSWdsSwiPDPRuntimeSettings	1548
9.45.7.51 unpack_wds_DHCPv4ClientLease_ind	1548
9.45.7.52 unpack_wds_DHCPv4ClientLeaseChange	1548
9.45.7.53 unpack_wds_GetAutoconnect	1549
9.45.7.54 unpack_wds_GetByteTotals	1549
9.45.7.55 unpack_wds_GetConnectionRate	1549
9.45.7.56 unpack_wds_GetDataBearerTechnology	1550
9.45.7.57 unpack_wds_GetDefaultProfile	1550
9.45.7.58 unpack_wds_GetDefaultProfileNum	1551
9.45.7.59 unpack_wds_GetDefaultProfileV2	1551
9.45.7.60 unpack_wds_GetDormancyState	1551
9.45.7.61 unpack_wds_GetLastMobileIPError	1552
9.45.7.62 unpack_wds_GetMobileIP	1552
9.45.7.63 unpack_wds_GetMobileIPProfile	1552
9.45.7.64 unpack_wds_GetPacketStatistics	1553
9.45.7.65 unpack_wds_GetPacketStatus	1553
9.45.7.66 unpack_wds_GetSessionDuration	1553
9.45.7.67 unpack_wds_GetSessionDurationV2	1554
9.45.7.68 unpack_wds_GetSessionState	1554
9.45.7.69 unpack_wds_RMSetTransferStatistics	1554
9.45.7.70 unpack_wds_RMTransferStatistics_ind	1555
9.45.7.71 unpack_wds_SetAutoconnect	1555
9.45.7.72 unpack_wds_SetDefaultProfile	1555
9.45.7.73 unpack_wds_SetDefaultProfileNum	1556
9.45.7.74 unpack_wds_SetMobileIP	1556
9.45.7.75 unpack_wds_SetMobileIPParameters	1556
9.45.7.76 unpack_wds_SetMobileIPProfile	1557
9.45.7.77 unpack_wds_SetMuxID	1557
9.45.7.78 unpack_wds_SLQSCreateProfile	1557
9.45.7.79 unpack_wds_SLQSDeleteProfile	1558
9.45.7.80 unpack_wds_SLQSDUNCallInfoCallBack_ind	1558
9.45.7.81 unpack_wds_SLQSGet3GPPConfigItem	1558
9.45.7.82 unpack_wds_SLQSGetCurrDataSystemStat	1559

9.45.7.83 unpack_wds_SLQSGetCurrentChannelRate	1559
9.45.7.84 unpack_wds_SLQSGetDataBearerTechnology	1559
9.45.7.85 unpack_wds_SLQSGetDUNCallInfo	1560
9.45.7.86 unpack_wds_SLQSGetProfileSettings	1560
9.45.7.87 unpack_wds_SLQSGetProfileSettingsV2	1560
9.45.7.88 unpack_wds_SLQSGetRuntimeSettings	1561
9.45.7.89 unpack_wds_SLQSModifyProfile	1561
9.45.7.90 unpack_wds_SLQSResetPacketStatics	1561
9.45.7.91 unpack_wds_SLQSSet3GPPConfigItem	1562
9.45.7.92 unpack_wds_SLQSSetIPFamilyPreference	1562
9.45.7.93 unpack_wds_SLQSSetPacketSrvStatusCallback	1562
9.45.7.94 unpack_wds_SLQSSetWdsEventCallback	1563
9.45.7.95 unpack_wds_SLQSSetWdsEventCallback_ind	1563
9.45.7.96 unpack_wds_SLQSSetDHCPv4ClientConfig	1563
9.45.7.97 unpack_wds_SLQSSetLoopback	1564
9.45.7.98 unpack_wds_SLQSSetDHCPv4ClientConfig	1564
9.45.7.99 unpack_wds_SLQSSetLoopback	1564
9.45.7.100unpack_wds_SLQSStartDataSession	1565
9.45.7.101unpack_wds_SLQSStopDataSession	1565
9.45.7.102unpack_wds_SLQSWdsGoActive	1565
9.45.7.103unpack_wds_SLQSWdsGoDormant	1566
9.45.7.104unpack_wds_SLQSWdsSetEventReport	1566
9.45.7.105unpack_wds_SLQSWdsSwiPDPRuntimeSettings	1566

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)	31
Streaming Download Protocol (lite-fw)	32

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tables	33
----------------------------------	----

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

_litew_FirmwareFileInfo	35
_litew_FirmwareInfo	36
_litew_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_BdsSV	149
loc_BdsSVInfo	149
loc_CellDb	150
loc_ClkInfo	150
loc_GnssData	152
loc_gpsTime	153
loc_gyroAcceptReady	154
loc_gyroTempAcceptReady	155
loc_IPv4Config	156
loc_IPv4Info	156
loc_IPv6Config	157
loc_IPv6Info	157
loc_LocApplicationInfo	158
loc_precisionDilution	159
loc_satelliteInfo	160
loc_sensorDataUsage	162
loc_SV	163
loc_SVInfo	164
loc_svUsedforFix	164
loc_urlAddr	165
loc_URLAddrInfo	165
lteSSInfo	166
messageModeTlv	167
nas_acqOrderPref	167
nas_AcqOrderPrefTlv	168
nas_ActPilotPNElement	169
nas_AddCDMASysInfo	169
nas_AddSysInfo	170
nas_BandPrefInfoTlv	170
nas_BandPrefTlv	171
nas_CallBarringSysInfo	173
nas_callBarStatus	173
nas_CDMAChannel	174
nas_CDMAECIOThresh	175

nas_CDMAInfo	176
nas_CDMARSSIThresh	177
nas_CDMASysInfo	177
nas_CDMASysInfoExt	181
nas_cellParams	181
nas_ciotAcqOrderPref	182
nas_CiotAcqOrderPrefTlv	183
nas_CiotLteOpModePrefTlv	184
nas_CommInfo	184
nas_CSGID	186
nas_CsgId	187
nas_currentPLMN	187
nas_dataSrvCapabilities	188
nas_DataStatusDetail	189
nas_detailSvcInfo	191
nas_DeviceConfigDetail	192
nas_dirNum	193
nas_DRCPParams	193
nas_ecioListElement	194
nas_ECIOThresh	195
nas_EdrxCiotLteMode	195
nas_EdrxCycleLength	196
nas_EdrxEnableType	196
nas_EdrxPagingTimeWindow	197
nas_EdrxRatType	197
nas_EmerModeTlv	198
nas_errorRateListElement	199
nas_ForbiddenNetworks3GPP	200
nas_GERANInfo	201
nas_geranInstInfo	202
nas_gsmCellInfo	203
nas_GSMRSSIThresh	204
nas_GSMSrvStatusInfo	205
nas_GSMSysInfo	206
nas_GWAcqOrderPrefTlv	209
nas_HDRECIOThresh	209
nas_HDRIOThresh	210
nas_HDRPersonality_Ind_Data	210
nas_HDRRSSIThresh	211
nas_HDRSINRThresh	212
nas_HDRSINRThreshold	212
nas_HDRSysInfo	213
nas_homeSIDNID	215
nas_infoInterFreq	215
nas_IOThresh	217
nas_lteBandPrefExt	217
nas_LTEBandPrefTlv	218
nas_LteCiotOpModeTlv	220
nas_lteEARFCN	220
nas_lteGsmCellInfo	221
nas_LTEInfo	222
nas_LTEInfoInterfreq	225
nas_LTEInfoIntrafreq	225
nas_LTEInfoNeighboringGSM	227
nas_LTEInfoNeighboringWCDMA	228
nas_LteM1BandPrefTlv	229
nas_LteNb1BandPrefTlv	231
nas_LTEOperationMode	233

nas_LteOpMode	233
nas_LteOpModeTlv	234
nas_LtePCI	235
nas_LteRsrpInformation	235
nas_LTEsSRPThresh	236
nas_LTEsSRQThresh	236
nas_LTEsSSThresh	237
nas_LTESigRptCfg	238
nas_LTESigRptConfig	238
nas_LteSnrInformation	239
nas_LTESNRThresh	240
nas_LTESNRThreshold	240
nas_LTESysInfo	241
nas_LteWcdmaCellInfo	243
nas_minBasedIMSI	244
nas_MNRInfo	245
nas_ModePrefTlv	246
nas_namName	246
nas_netSelectionPref	247
nas_NetSelPrefTlv	248
nas_NetworkStat1x	248
nas_NetworkStatEVDO	250
nas_nmrCellInfo	252
nas_nr5gBandPref	253
nas_NR5GCellStatusTlv	254
nas_NR5GSerStatTlv	255
nas_NR5GSystemInfoTlv	256
nas_operatorNameString	259
nas_OperatorPLMNData	259
nas_operatorPLMNList	260
nas_PhyCaAggPcellInfo	261
nas_PhyCaAggScellDIBw	262
nas_PhyCaAggScellIndex	262
nas_PhyCaAggScellIndType	263
nas_PhyCaAggScellInfo	264
nas_PilotSetData	266
nas_PilotSetParams	267
nas_PlmnID	267
nas_PLMNNetworkName	268
nas_PLMNNetworkNameData	268
nas_PRLPrefTlv	270
nas_protocolSubtypeElement	271
nas_qaQmi3Gpp2TimeZone	272
nas_QmiNas3GppNetworkInfo	273
nas_QmiNas3GppNetworkRAT	274
nas_QmisNasPcsDigit	275
nas_QmisNasSigsNasPCICellInfo	276
nas_QmisNasSigsNasPCIInfo	277
nas_RankIndicatorTlv	278
nas_RatDisabledMaskTlv	278
nas_RejectReasonTlv	279
nas_RFBandInfoElements	279
nas_RfBandInfoExtFormat	280
nas_RfBandInfoExtFormatElements	281
nas_RfBandInfoList	282
nas_RfBandwidthInfo	282
nas_RfBandwidthInfoElements	283
nas_RfDedicatedBandInfo	284

nas_RfDedicatedBandInfoElements	284
nas_RFInfoTlv	285
nas_roamIndList	286
nas_RoamPrefTlv	287
nas_RSRPThresh	287
nas_rsrqInformation	288
nas_RSRQThresh	288
nas_RSSIThresh	289
nas_rxInfo	290
nas_RxSigInfo	291
nas_rxSignalStrengthListElement	292
nas_SccRxInfo	293
nas_serviceProviderName	294
nas_servSystem	295
nas_sidNid	296
nas_SignalStrengthTlv	297
nas_SLQSSignalStrengthsIndReq	297
nas_SLQSSignalStrengthsInformation	299
nas_SLQSSignalStrengthsTlv	300
nas_SrvDomainPrefTlv	301
nas_SrvStatusInfo	301
nas_sysInfoCommon	302
nas_TDSCDMAECIOThresh	305
nas_TDSCDMARSCPTthresh	305
nas_TDSCDMARSSIThresh	306
nas_TDSCDMASINRCONFThresh	306
nas_TDSCDMASINRThresh	307
nas_timeInfo	307
nas_trueIMSI	309
nas_txInfo	310
nas_UMTSInfo	310
nas_UMTSinstInfo	312
nas_umtsLTENbrCell	313
nas_UniversalTime	314
nas_wcdmaCellInfo	315
nas_WCDMAECIOThresh	316
nas_WCDMAInfoLTENeighborCell	316
nas_WCDMARSSIThresh	317
nas_WCDMASysInfo	318
nas_wcdmaUARFCN	321
NASAcqOrderPrefTlv	322
NASBandPreferenceTlv	322
NASCiotAcqOrderPrefTlv	323
NASCiotLteOpModePrefTlv	323
NASEmergencyModeTlv	323
NasGetLTECphyCalInfo	324
NASGWAOrderPrefTlv	325
NASLTEBandPreferenceTlv	325
NASLteM1BandPrefTlv	326
NASLteNasReleaseInfoTlv	326
NASLteNB1BandPrefTlv	327
NASModePreferenceTlv	327
NASNetSelPreferenceTlv	328
NASNr5gBandPrefTlv	328
NASOTAMessageTlv	329
NASPhyCaAggPcellInfo	330
NASPhyCaAggScellArray	331
NASPhyCaAggScellIDBw	332

NASPhyCaAggScellIndex	332
NASPhyCaAggScellIndType	333
NASPhyCaAggScellInfo	334
NASPRLPreferenceTlv	335
NASQmiCbkNasSwiOTAMessageInd	335
NASQmiCbkNasSystemSelPrefInd	336
NASRatDisabledMaskTlv	338
NASRoamPreferenceTlv	339
NASServDomainPrefTlv	339
NASServingSystemInfo	339
NASTimeInfoTlv	341
newMTMessageTlv	342
pack_audio_SLQSGetAudioPathConfig_t	342
pack_audio_SLQSGetAudioProfile_t	343
pack_audio_SLQSGetAudioVolTLBConfig_t	344
pack_audio_SLQSSetAudioPathConfig_t	344
pack_audio_SLQSSetAudioProfile_t	346
pack_audio_SLQSSetAudioVolTLBConfig_t	348
pack_cat_CATSendEnvelopeCommand_t	348
pack_cat_CATSendTerminalResponse_t	349
pack_cat_SetCATEventCallback_t	350
pack_dms_ActivateAutomatic_t	351
pack_dms_GetCustFeaturesV2_t	351
pack_dms_ResetToFactoryDefaults_t	352
pack_dms_SetActivationStatusCallback_t	352
pack_dms_SetCrashAction_t	352
pack_dms_SetCustFeature_t	353
pack_dms_SetCustFeaturesV2_t	355
pack_dms_SetEventReport_t	356
pack_dms_SetIndicationRegister_t	356
pack_dms_SetPower_t	357
pack_dms_SetUSBComp_t	358
pack_dms_SLQSDmsSwiIndicationRegister_t	359
pack_dms_SLQSSetPowerSaveModeConfig_t	359
pack_dms_SLQSSwiGetCrashInfo_t	360
pack_dms_SLQSSwiSetDyingGaspCfg_t	360
pack_dms_SLQSSwiSetHostDevInfo_t	361
pack_dms_SLQSSwiSetOSInfo_t	362
pack_dms_SwiSetEventReport_t	362
pack_dms_SwiUimSelect_t	363
pack_dms_UIMChangePIN_t	364
pack_dms_UIMGetControlKeyStatus_t	364
pack_dms_UIMGetICCID_t	365
pack_dms_UIMSetControlKeyProtection_t	365
pack_dms_UIMSetPINProtection_t	366
pack_dms_UIMUnblockControlKey_t	367
pack_dms_UIMUnblockPIN_t	367
pack_dms_UIMVerifyPIN_t	368
pack_fms_GetImagesPreference_t	368
pack_fms_GetStoredImages_t	369
pack_fms_SetImagesPreference_t	369
pack_ims_SLQSImsConfigIndicationRegister_t	370
pack_ims_SLQSSetIMSSMSConfig_t	371
pack_ims_SLQSSetIMSUserConfig_t	372
pack_ims_SLQSSetIMSVoIPConfig_t	373
pack_ims_SLQSSetRegMgrConfig_t	375
pack_ims_SLQSSetSIPConfig_t	376
pack_imsa_SLQSRegisterIMSAIndication_t	377

pack_loc_Delete_Assist_Data_t	378
pack_loc_EventRegister_t	379
pack_loc_SetExtPowerState_t	381
pack_loc_SetOperationMode_t	381
pack_loc_SLQSLOCGetBestAvailPos_t	382
pack_loc_SLQSLOCGetServer_t	383
pack_loc_SLQSLOCInjectPosition_t	383
pack_loc_SLQSLOCInjectSensorData_t	388
pack_loc_SLQSLOCInjectUTCTime_t	390
pack_loc_SLQSLOCSetCradleMountConfig_t	390
pack_loc_SLQSLOCSetServer_t	391
pack_loc_Start_t	392
pack_loc_Stop_t	394
pack_nas_InitiateDomainAttach_t	394
pack_nas_PerformNetworkScanPCI_t	395
pack_nas_SetACCOLC_t	395
pack_nas_SetCDMANetworkParameters_t	395
pack_nas_SetNetworkPreference_t	397
pack_nas_SLQSConfigSigInfo_t	398
pack_nas_SLQSGetPLMNName_t	399
pack_nas_SLQSIInitiateNetworkRegistration_t	400
pack_nas_SLQSNasConfigSigInfo2_t	401
pack_nas_SLQSNasGet3GPP2Subscription_t	406
pack_nas_SLQSNASGeteDRXParamsExt_t	406
pack_nas_SLQSNasGetTxRxInfo_t	407
pack_nas_SLQSNasIndicationRegisterExt_t	408
pack_nas_SLQSNASSeteDRXParams_t	411
pack_nas_SLQSNasSwiIndicationRegister_t	412
pack_nas_SLQSNASSwiSetChannelLock_t	414
pack_nas_SLQSSetBandPreference_t	414
pack_nas_SLQSSetSignalStrengthsCallback_t	416
pack_nas_SLQSSetSysSelectionPref_t	416
pack_nas_SLQSSetSysSelectionPrefExt_t	422
pack_nas_SLQSSwiPSDetach_t	432
pack_pds_PDSInjectTimeReference_t	432
pack_pds_ResetPDSDData_t	433
pack_pds_SetEventReportCallback_t	434
pack_pds_SetPDSDefaults_t	434
pack_pds_SetPDSSState_t	435
pack_pds_SetPortAutomaticTracking_t	435
pack_pds_SetServiceAutomaticTracking_t	436
pack_pds_SetXTRAAutomaticDownload_t	436
pack_pds_SetXTRANetwork_t	437
pack_pds_SLQSGetAGPSConfig_t	437
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t	438
pack_pds_SLQSPDSInjectPositionData_t	439
pack_pds_SLQSSetAGPSConfig_t	441
pack_pds_SLQSSetPositionMethodState_t	441
pack_pds_StartPDSTrackingSessionExt_t	442
pack_qmi_t	443
pack_qos_BindDataPort_t	444
pack_qos_SLQSQosSwiReadApnExtraParams_t	445
pack_qos_SLQSQosSwiReadDataStats_t	445
pack_qos_SLQSSetQosEventCallback_t	445
pack_rms_SetSMSWake_t	446
pack_sar_SLQSSetRfSarState_t	446
pack_sms_SaveSMS_t	447
pack_sms_SendSMS_t	448

Generated on Thu Nov 21 2019 08:29:34 for LinuxQMISDK-Lite by Doxygen

pack_uim_SLQSUIMRefreshOK_t	487
pack_uim_SLQSUIMRefreshRegister_t	488
pack_uim_SLQSUIMSwitchSlot_t	488
pack_uim_UnblockPin_t	489
pack_uim_VerifyPin_t	490
pack_voice_AnswerUSSD_t	491
pack_voice_OriginateUSSD_t	492
pack_voice_SLQSOriginateUSSD_t	492
pack_voice_SLQSVoiceALSSelectLine_t	493
pack_voice_SLQSVoiceALSSetLineSwitching_t	493
pack_voice_SLQSVoiceAnswerCall_t	494
pack_voice_SLQSVoiceBindSubscription_t	494
pack_voice_SLQSVoiceBurstDTMF_t	494
pack_voice_SLQSVoiceDialCall_t	495
pack_voice_SLQSVoiceEndCall_t	497
pack_voice_SLQSVoiceGetCallBarring_t	497
pack_voice_SLQSVoiceGetCallForwardingStatus_t	498
pack_voice_SLQSVoiceGetCallInfo_t	499
pack_voice_SLQSVoiceGetCallWaiting_t	499
pack_voice_SLQSVoiceGetConfig_t	500
pack_voice_SLQSVoiceIndicationRegister_t	502
pack_voice_SLQSVoiceManageCalls_t	502
pack_voice_SLQSVoiceOrigUSSDNoWait_t	503
pack_voice_SLQSVoiceSendFlash_t	504
pack_voice_SLQSVoiceSetCallBarringPassword_t	504
pack_voice_SLQSVoiceSetConfig_t	505
pack_voice_SLQSVoiceSetPreferredPrivacy_t	507
pack_voice_SLQSVoiceSetSUPSService_t	507
pack_voice_SLQSVoiceStartContDTMF_t	509
pack_voice_SLQSVoiceStopContDTMF_t	510
pack_wds_DHCPv4ClientLeaseChange_t	510
pack_wds_GetDefaultProfile_t	511
pack_wds_GetDefaultProfileNum_t	511
pack_wds_GetDefaultProfileV2_t	512
pack_wds_GetDormancyState_t	512
pack_wds_GetLastMobileIPError_t	512
pack_wds_GetMobileIP_t	512
pack_wds_GetMobileIPProfile_t	513
pack_wds_GetPacketStatistics_t	513
pack_wds_GetPacketStatus_t	514
pack_wds_GetSessionDuration_t	514
pack_wds_RMSetTransferStatistics_t	514
pack_wds_SetAutoconnect_t	515
pack_wds_SetDefaultProfile_t	515
pack_wds_SetDefaultProfileNum_t	517
pack_wds_SetMobileIP_t	518
pack_wds_SetMobileIPParameters_t	518
pack_wds_SetMobileIPProfile_t	519
pack_wds_SLQSCreateProfile_t	521
pack_wds_SLQSDeleteProfile_t	522
pack_wds_SLQSGetCurrDataSystemStat_t	523
pack_wds_SLQSGetDataBearerTechnology_t	523
pack_wds_SLQSGetDUNCallInfo_t	523
pack_wds_SLQSGetProfileSettings_t	525
pack_wds_SLQSGetRuntimeSettings_t	525
pack_wds_SLQSModifyProfile_t	526
pack_wds_SLQSSet3GPPConfigItem_t	527
pack_wds_SLQSSetIPFamilyPreference_t	529

pack_wds_SLQSSetWdsEventCallback_t	529
pack_wds_SLQSSGetDHCPv4ClientConfig_t	530
pack_wds_SLQSSSetDHCPv4ClientConfig_t	530
pack_wds_SLQSSSetLoopback_t	531
pack_wds_SLQSSStartDataSession_t	531
pack_wds_SLQSSStopDataSession_t	533
pack_wds_SLQSWdsSetEventReport_t	533
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	535
PackCreateProfileOut	535
packgetDyingGaspCfg	536
packgetDyingGaspStatistics	536
PackSwiAvmsSetSettingsAPNInfo	537
PackSwiAvmsSetSettingsConnectionRetryTimers	538
PackSwiAvmsSetSettingsPeriodInfo	539
PackSwiAVMSSettingsAPNInfo	539
PackSwiAVMSSettingsConnectionRetryTimers	540
PackSwiAVMSSettingsPeriodsInfo	541
qmiSmsMessageList	541
qmiWSDDataBearerTechnology	541
qmTlvResult	542
qos_BindDataPortMuxID_t	542
qos_BindDataPortPeripheralEndPointID_t	543
qos_BindDataPortSIODDataPort_t	543
RFBandInfoElements	544
rmTrasnferStaticsReq	544
sensorData_t	545
slot_t	547
slotInf	548
slots_t	549
sms_BroadcastConfig	550
sms_CDMABroadcastConfig	550
sms_getIndicationReg	551
sms_getMsgWaitingInfo	552
sms_getTransLayerInfo	552
sms_getTransNWRegInfo	553
sms_maxStorageSizeReq	554
sms_maxStorageSizeResp	554
sms_messageWaitingInfoContent	555
sms_msgProtocolResp	556
sms_qaQmi3GPP2BroadcastCfgInfo	556
sms_qaQmi3GPPBroadcastCfgInfo	557
sms_routeEntry	557
sms_sendAsynCsmsParams	558
sms_setIndicationReg	561
sms_setRoutesReq	561
sms_transLayerInfo	562
sMSCAddressInfo	563
sMSCAddressTlv	563
sMSEtwsMessageInfo	564
sMSEtwsMessageTlv	564
sMSEtwsPlmnInfo	565
sMSMessageModelInfo	565
sMSMTMessageInfo	566
sMSOnIMSInfo	566
sMSOnIMSTlv	567
sMSTransferRouteMTMessageInfo	567
swi_uint256_t	568
swiaudio_PCMparams	568

swidms_ehrpdMTUSizeTlv	569
swidms_hrpdmTUSizeTlv	569
swidms_ifaceCfgTlv	570
swidms_mtuSize3gppTlv	571
swidms_supportedIntBitmaskTlv	572
swidms_SwiDmsGetHWWatchdog	572
swidms_usbMTUSizeTlv	573
tdscdmaSigInfoExt	574
tempData_t	574
tmd_mitigationDevList	575
transferRouteMessageTlv	576
uim_appStatus	576
uim_authenticateResult	579
uim_authenticationData	580
uim_cardResult	581
uim_cardStatus	582
uim_changeUIMPIN	583
uim_depersonalizationInformation	584
uim_encryptedPIN1	585
uim_fileAttributes	586
uim_fileInfo	589
uim_GetSlotsInfoTlv	590
uim_GetSlotsStatusTlv	591
uim_hotSwapStatus	591
uim_personalizationStatus	592
uim_physlotInfo	593
uim_physlotsInfo	594
uim_readResult	594
uim_readTransparentInfo	595
uim_refreshevent	595
uim_registerRefresh	597
uim_remainingRetries	598
uim_sessionInformation	598
uim_setPINProtection	599
uim_simBusyStatus	600
uim_slotInfo	601
uim_UIMSessionInformation	602
uim_unblockUIMPIN	603
uim_validCardStatus	604
uim_verifyUIMPIN	605
unpack_audio_SLQSGetAudioPathConfig_t	605
unpack_audio_SLQSGetAudioProfile_t	608
unpack_audio_SLQSGetAudioVoTLBConfig_t	609
unpack_audio_SLQSSetAudioVoTLBConfig_t	609
unpack_cat_SetCatEventCallback_ind_t	610
unpack_cat_SetCATEventCallback_t	610
unpack_dms_GetActivationState_t	611
unpack_dms_GetBandCapability_t	612
unpack_dms_GetCrashAction_t	614
unpack_dms_GetCustFeature_t	615
unpack_dms_GetCustFeaturesV2_t	617
unpack_dms_GetDeviceCap_t	618
unpack_dms_GetDeviceCapabilities_t	619
unpack_dms_GetDeviceCapabilitiesV2_t	621
unpack_dms_GetDeviceHardwareRev_t	623
unpack_dms_GetDeviceMfr_t	624
unpack_dms_GetDeviceSerialNumbers_t	625
unpack_dms_GetFirmwareInfo_t	627

unpack_dms_GetFirmwareRevision_t	629
unpack_dms_GetFirmwareRevisions_t	630
unpack_dms_GetFSN_t	631
unpack_dms_GetHardwareRevision_t	632
unpack_dms_GetIMSI_t	632
unpack_dms_GetManufacturer_t	633
unpack_dms_GetModelID_t	633
unpack_dms_GetNetworkTime_t	634
unpack_dms_GetNetworkTimeV2_t	635
unpack_dms_GetOfflineReason_t	636
unpack_dms_GetPower_t	637
unpack_dms_GetPRLVersion_t	638
unpack_dms_GetSerialNumbers_t	639
unpack_dms_GetUSBComp_t	640
unpack_dms_GetVoiceNumber_t	642
unpack_dms_PSMCfgChange_ind_t	644
unpack_dms_ResetToFactoryDefaults_t	644
unpack_dms_SetActivationStatusCallback_t	645
unpack_dms_SetCrashAction_t	645
unpack_dms_SetCustFeature_t	646
unpack_dms_SetCustFeaturesV2_t	646
unpack_dms_SetEventReport_ind_t	647
unpack_dms_SetEventReport_t	648
unpack_dms_SetFirmwarePreference_t	648
unpack_dms_SetIndicationRegister_t	649
unpack_dms_SetPower_t	649
unpack_dms_SetUSBComp_t	650
unpack_dms_SLQSDmsSwiGetPCInfo_t	650
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t	653
unpack_dms_SLQSDmsSwiGetResetInfo_t	654
unpack_dms_SLQSDmsSwiGetUimSelection_t	655
unpack_dms_SLQSDmsSwiIndicationRegister_t	656
unpack_dms_SLQSGetBandCapability_t	657
unpack_dms_SLQSGetBandCapabilityExt_t	661
unpack_dms_SLQSGetERIFile_t	665
unpack_dms_SLQSGetPowerSaveModeConfig_t	666
unpack_dms_SLQSSetPowerSaveModeConfig_t	667
unpack_dms_SLQSSwiClearDyingGaspStatistics_t	668
unpack_dms_SLQSSwiGetCrashInfo_t	668
unpack_dms_SLQSSwiGetDyingGaspCfg_t	669
unpack_dms_SLQSSwiGetDyingGaspStatistics_t	670
unpack_dms_SLQSSwiGetFirmwareCurr_t	670
unpack_dms_SLQSSwiGetFwUpdateStatus_t	671
unpack_dms_SLQSSwiGetHostDevInfo_t	673
unpack_dms_SLQSSwiGetOSInfo_t	674
unpack_dms_SLQSSwiGetSerialNoExt_t	675
unpack_dms_SLQSSwiSetDyingGaspCfg_t	676
unpack_dms_SLQSSwiSetHostDevInfo_t	676
unpack_dms_SLQSSwiSetOSInfo_t	677
unpack_dms_SLQSUIMGetState_t	677
unpack_dms_SwiEventReportCallBack_ind_t	678
unpack_dms_SwiSetEventReport_t	679
unpack_dms_SwiUimSelect_t	679
unpack_dms_UIMGetControlKeyStatus_t	680
unpack_dms_UIMGetICCID_t	681
unpack_dms_UIMGetPINStatus_t	682
unpack_dms_UIMSetControlKeyProtection_t	683
unpack_dms_UIMSetPINProtection_t	684

unpack_dms_UIMUnblockControlKey_t	685
unpack_fms_GetImagesPreference_t	685
unpack_fms_GetStoredImages_t	686
unpack_fms_SetImagesPreference_t	687
unpack_ims_SLQSGetIMSSMSConfig_t	687
unpack_ims_SLQSGetIMSUserConfig_t	688
unpack_ims_SLQSGetIMSVoIPConfig_t	689
unpack_ims_SLQSGetRegMgrConfig_t	692
unpack_ims_SLQSGetSIPConfig_t	693
unpack_ims_SLQSRegMgrCfgCallBack_ind_t	694
unpack_ims_SLQSSetIMSSMSConfig_t	695
unpack_ims_SLQSSetIMSUserConfig_t	696
unpack_ims_SLQSSetIMSVoIPConfig_t	696
unpack_ims_SLQSSetRegMgrConfig_t	697
unpack_ims_SLQSSetSIPConfig_t	697
unpack_ims_SLQSSIPCfgCallBack_ind_t	698
unpack_ims_SLQSSMSCfgCallBack_ind_t	699
unpack_ims_SLQSUserCfgCallBack_ind_t	700
unpack_ims_SLQSVoIPCfgCallBack_ind_t	701
unpack_imsa_SLQSGetIMSARegStatus_t	703
unpack_imsa_SLQSGetIMSAServiceStatus_t	704
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t	706
unpack_imsa_SLQSImsaRatStatusCallBack_ind_t	707
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t	708
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t	709
unpack_loc_BestAvailPos_Ind_t	710
unpack_loc_CradleMountCallback_Ind_t	717
unpack_loc_Delete_Assist_Data_t	717
unpack_loc_DeleteAssistData_Ind_t	718
unpack_loc_EngineState_Ind_t	719
unpack_loc_EventNMEA_Ind_t	720
unpack_loc_EventRegister_t	721
unpack_loc_EventTimeSyncCallback_Ind_t	721
unpack_loc_GetOpMode_Ind_t	721
unpack_loc_GetServer_Ind_t	723
unpack_loc_GnssSvInfo_Ind_t	724
unpack_loc_InjectPositionCallback_Ind_t	725
unpack_loc_InjectSensorDataCallback_Ind_t	726
unpack_loc_InjectTimeSyncDataCallback_Ind_t	728
unpack_loc_InjectUTCTimeCallback_Ind_t	728
unpack_loc_PositionRpt_Ind_t	729
unpack_loc_SensorStreamingCallback_Ind_t	736
unpack_loc_SetExtPowerConfig_Ind_t	737
unpack_loc_SetExtPowerState_t	737
unpack_loc_SetOperationMode_Ind_t	738
unpack_loc_SetOperationMode_t	739
unpack_loc_SetServer_Ind_t	739
unpack_loc_SLQSLOCGetBestAvailPos_t	740
unpack_loc_SLQSLOCGetOpMode_t	741
unpack_loc_Start_t	741
unpack_loc_Stop_t	742
unpack_nas_GetACCOLC_t	742
unpack_nas_GetANAAAAAuthenticationStatus_t	743
unpack_nas_GetCDMANetworkParameters_t	743
unpack_nas_GetHomeNetwork3GPP2_t	746
unpack_nas_GetHomeNetwork_t	748
unpack_nas_GetNetworkPreference_t	749
unpack_nas_GetRFInfo_t	750

unpack_nas_GetServingNetwork_t	751
unpack_nas_GetServingNetworkCapabilities_t	754
unpack_nas_GetSignalStrengths_t	755
unpack_nas_PerformNetworkScan_t	756
unpack_nas_SetDataCapabilitiesCallback_ind_t	757
unpack_nas_SetEventReportInd_t	758
unpack_nas_SetNasLTECphyCalIndCallback_ind_t	759
unpack_nas_SetNetworkPreference_t	760
unpack_nas_SetRoamingIndicatorCallback_ind_t	761
unpack_nas_SetServingSystemCallback_ind_t	761
unpack_nas_SLQSGetErrorRate_t	762
unpack_nas_SlqsGetLTECphyCAInfo_t	763
unpack_nas_SLQSGetNetworkTime_t	764
unpack_nas_SLQSGetOperatorNameData_t	764
unpack_nas_SLQSGetPLMNName_t	766
unpack_nas_SLQSGetServingSystem_t	769
unpack_nas_SLQSGetSignalStrength_t	773
unpack_nas_SLQSGetSysInfo_t	776
unpack_nas_SLQSGetSysSelectionPref_t	780
unpack_nas_SLQSGetSysSelectionPrefExt_t	784
unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t	787
unpack_nas_SLQSNasGet3GPP2Subscription_t	788
unpack_nas_SLQSNasGetCellLocationInfo_t	789
unpack_nas_SLQSNASGeteDRXParams_t	791
unpack_nas_SLQSNASGeteDRXParamsExt_t	792
unpack_nas_SLQSNASGetForbiddenNetworks_t	793
unpack_nas_SLQSNasGetHDRColorCode_t	794
unpack_nas_SLQSNasGetRFInfo_t	794
unpack_nas_SLQSNasGetSigInfo_t	795
unpack_nas_SLQSNasGetTxRxInfo_t	796
unpack_nas_SLQSNasNetworkRejectCallback_Ind_t	797
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t	800
unpack_nas_SLQSNasSigInfoCallback_ind_t	801
unpack_nas_SLQSNASSwiGetChannelLock_t	803
unpack_nas_SLQSNasSwiModemStatus_t	803
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t	804
unpack_nas_SLQSNasTimerCallback_ind_t	805
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t	805
unpack_nas_SLQSSwiGetHDRPersonality_t	806
unpack_nas_SLQSSwiGetHDRProtSubtype_t	807
unpack_nas_SLQSSwiGetHRPDStats_t	808
unpack_nas_SLQSSwiGetLteCQI_t	809
unpack_nas_SLQSSwiGetLteSccRxInfo_t	810
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t	810
unpack_nas_SLQSSwiNetworkDebug_t	811
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t	812
unpack_nas_SLQSSysInfoCallback_ind_t	812
unpack_omaDmConfigTlv_t	816
unpack_omaDmFotaTlv_t	817
unpack_omaDmNotificationsTlv_t	819
unpack_pds_GetPDSDDefaults_t	820
unpack_pds_GetPDSSState_t	821
unpack_pds_GetPortAutomaticTracking_t	822
unpack_pds_GetServiceAutomaticTracking_t	823
unpack_pds_GetXTRAAutomaticDownload_t	823
unpack_pds_GetXTRANetwork_t	824
unpack_pds_GetXTRAValidity_t	824
unpack_pds_SetEventReport_Ind_t	825

unpack_pds_SetPdsState_ind_t	827
unpack_pds_SLQSGetAGPSConfig_t	828
unpack_pds_SLQSGetGPSSStateInfo_t	829
unpack_qmi_t	834
unpack_qos_BindDataPort_t	834
unpack_qos_dataRate_t	835
unpack_qos_IPv4Addr_t	835
unpack_qos_IPv6Addr_t	835
unpack_qos_IPv6TrafCls_t	836
unpack_qos_pktErrRate_t	836
unpack_qos_Port_t	837
unpack_qos_QosFlowInfo_t	837
unpack_qos_QosFlowInfoState_t	839
unpack_qos_SLQSQosGetNetworkStatus_t	839
unpack_qos_SLQSQosSwiReadApnExtraParams_t	840
unpack_qos_SLQSQosSwiReadDataStats_t	842
unpack_qos_SLQSSetQosEventCallback_ind_t	843
unpack_qos_SLQSSetQosEventCallback_t	844
unpack_qos_SLQSSetQosNWStatusCallback_ind_t	844
unpack_qos_SLQSSetQosPriEventCallback_ind_t	845
unpack_qos_SLQSSetQosStatusCallback_ind_t	845
unpack_qos_swiQosFilter_t	847
unpack_qos_swiQosFlow_t	850
unpack_qos_tokenBucket_t	854
unpack_qos_Tos_t	855
unpack_QosFlowStat_t	855
unpack_result_t	856
unpack_rms_GetSMSWake_t	857
unpack_rms_SetSMSWake_t	857
unpack_RMTransferStatistics_ind_t	858
unpack_sar_SLQSGetRfSarState_t	859
unpack_sms_GetSMSCAddress_t	859
unpack_sms_SaveSMS_t	860
unpack_sms_SendSMS_t	861
unpack_sms_SetNewSMSCallback_ind_t	862
unpack_sms_SetNewSMSCallback_t	863
unpack_sms_SetSMSCAddress_t	863
unpack_sms_SLQSDeleteSMS_t	864
unpack_sms_SLQSGetIndicationRegister_t	864
unpack_sms_SLQSGetMessageWaiting_t	865
unpack_sms_SLQSGetSMS_t	865
unpack_sms_SLQSGetSmsBroadcastConfig_t	866
unpack_sms_SLQSGetSMSList_t	867
unpack_sms_SLQSGetTransLayerInfo_t	868
unpack_sms_SLQSGetTransNWRegInfo_t	868
unpack_sms_SLQSMModifySMSStatus_t	869
unpack_sms_SLQSNWRegInfoCallback_ind_t	869
unpack_sms_SLQSSendAsyncSMS_t	870
unpack_sms_SLQSSetIndicationRegister_t	870
unpack_sms_SLQSSetSmsBroadcastActivation_t	871
unpack_sms_SLQSSetSmsBroadcastConfig_t	871
unpack_sms_SLQSSetSmsStorage_t	872
unpack_sms_SLQSSmsGetMaxStorageSize_t	872
unpack_sms_SLQSSmsGetMessageProtocol_t	873
unpack_sms_SLQSSmsSetRoutes_t	873
unpack_sms_SLQSSwiGetSMSStorage_t	874
unpack_sms_SLQSTransLayerInfoCallback_ind_t	874
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t	875

unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	877
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t	878
unpack_swiaudio_SLQSGetM2MAudioProfile_t	878
unpack_swiaudio_SLQSGetM2MAudioVolume_t	880
unpack_swiaudio_SLQSGetM2MAVMute_t	880
unpack_swiaudio_SLQSGetM2MSpkrGain_t	881
unpack_swiaavms_SLQSAVMSEventReportInd_t	882
unpack_swiaavms_SLQSAVMSGetSettings_t	883
unpack_swiaavms_SLQSAVMSGetSettings_v2_t	886
unpack_swiaavms_SLQSAVMSSendSelection_t	888
unpack_swiaavms_SLQSAVMSSessionGetInfo_t	889
unpack_swiaavms_SLQSAvmsSetEventReport_t	890
unpack_swiaavms_SLQSAVMSSetSettings_t	890
unpack_swiaavms_SLQSAVMSSetSettings_v2_t	891
unpack_swiaavms_SLQSAVMSStartSession_t	891
unpack_swiaavms_SLQSAVMSStopSession_t	892
unpack_swidms_SLQSSwiDmsGetHWWatchdog_t	892
unpack_swidms_SLQSSwiDmsGetMTU_t	893
unpack_swidms_SLQSSwiDmsGetSecureInfo_t	893
unpack_swidms_SLQSSwiDmsGetUsbComp_t	894
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t	895
unpack_swidms_SLQSSwiDmsSetHWWatchdog_t	896
unpack_swidms_SLQSSwiDmsSetMTU_t	896
unpack_swidms_SLQSSwiDmsSetUsbComp_t	897
unpack_swiloc_SwiLocGetAutoStart_t	897
unpack_swioma_SLQSOMADMAAlertCallback_ind_t	899
unpack_swioma_SLQSOMADMGetSessionInfo_t	900
unpack_swioma_SLQSOMADMGetSessionInfoExt_t	903
unpack_swioma_SLQSOMADMGetSettings_t	907
unpack_swioma_SLQSOMADMStartSession_t	908
unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t	909
unpack_tmd_SLQSTmdGetMitigationDevList_t	909
unpack_tmd_SLQSTmdGetMitigationLvl_t	910
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t	911
unpack_tmd_SLQSTmdRegNotMitigationLvl_t	911
unpack_uim_ChangePin_t	912
unpack_uim_GetCardStatus_t	913
unpack_uim_GetCardStatusV2_t	914
unpack_uim_ReadTransparent_t	915
unpack_uim_SetPinProtection_t	916
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	916
unpack_uim_SLQSUIMAuthenticate_t	917
unpack_uim_SLQSUIMDepersonalization_t	918
unpack_uim_SLQSUIMEventRegister_t	918
unpack_uim_SLQSUIMGetConfiguration_t	919
unpack_uim_SLQSUIMGetFileAttributes_t	920
unpack_uim_SLQSUIMGetSlotsStatus_t	921
unpack_uim_SLQSUIMGetSlotsStatusV2_t	921
unpack_uim_SLQSUIMRefreshCallback_Ind_t	922
unpack_uim_SLQSUIMRefreshGetLastEvent_t	923
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t	923
unpack_uim_UnblockPin_t	924
unpack_uim_UnblockPinV2_t	925
unpack_uim_VerifyPin_t	926
unpack_voice_allCallStatusCallback_ind_t	927
unpack_voice_DTMFEEventCallback_ind_t	929
unpack_voice_OTASPStatusCallback_ind_t	930
unpack_voice_SLQSOriginateUSSD_t	931

unpack_voice_SLQSVoiceAnswerCall_t	933
unpack_voice_SLQSVoiceBurstDTMF_t	934
unpack_voice_SLQSVoiceDialCall_t	934
unpack_voice_SLQSVoiceEndCall_t	935
unpack_voice_SLQSVoiceGetAllCallInfo_t	936
unpack_voice_SLQSVoiceGetCallBarring_t	940
unpack_voice_SLQSVoiceGetCallForwardingStatus_t	941
unpack_voice_SLQSVoiceGetCallInfo_t	943
unpack_voice_SLQSVoiceGetCallWaiting_t	946
unpack_voice_SLQSVoiceGetCLIP_t	947
unpack_voice_SLQSVoiceGetCLIR_t	949
unpack_voice_SLQSVoiceGetCNAP_t	950
unpack_voice_SLQSVoiceGetCOLP_t	952
unpack_voice_SLQSVoiceGetCOLR_t	953
unpack_voice_SLQSVoiceGetConfig_t	954
unpack_voice_SLQSVoiceManageCalls_t	956
unpack_voice_SLQSVoiceSendFlash_t	957
unpack_voice_SLQSVoiceSetCallBarringPassword_t	958
unpack_voice_SLQSVoiceSetConfig_t	959
unpack_voice_SLQSVoiceSetSUPSService_t	961
unpack_voice_SLQSVoiceStartContDTMF_t	962
unpack_voice_SLQSVoiceStopContDTMF_t	962
unpack_voice_SLQSVoiceSUPSCallback_ind_t	963
unpack_voice_SUPSNotificationCallback_ind_t	966
unpack_voice_USSDNotificationCallback_ind_t	968
unpack_voice_VoiceInfoRecCallback_ind_t	968
unpack_voice_voicePrivacyChangeCallback_ind_t	971
unpack_wds_DHCPv4ClientLease_ind_t	972
unpack_wds_GetAutoconnect_t	973
unpack_wds_GetByteTotals_t	973
unpack_wds_GetConnectionRate_t	974
unpack_wds_GetDataBearerTechnology_t	975
unpack_wds_GetDefaultProfile_t	976
unpack_wds_GetDefaultProfileNum_t	979
unpack_wds_GetDefaultProfileV2_t	979
unpack_wds_GetDormancyState_t	982
unpack_wds_GetLastMobileIPError_t	983
unpack_wds_GetMobileIP_t	984
unpack_wds_GetMobileIPProfile_t	984
unpack_wds_GetPacketStatistics_t	986
unpack_wds_GetPacketStatus_t	988
unpack_wds_GetSessionDuration_t	990
unpack_wds_GetSessionDurationV2_t	991
unpack_wds_GetSessionState_t	992
unpack_wds_RMSetTransferStatistics_t	992
unpack_wds_SetMobileIPProfile_t	993
unpack_wds_SLQSCreateProfile_t	993
unpack_wds_SLQSDeleteProfile_t	994
unpack_wds_SLQSDUNCAllInfoCallBack_ind_t	994
unpack_wds_SLQSGet3GPPConfigItem_t	996
unpack_wds_SLQSGetCurrDataSystemStat_t	998
unpack_wds_SLQSGetCurrentChannelRate_t	999
unpack_wds_SLQSGetDataBearerTechnology_t	1000
unpack_wds_SLQSGetDUNCAllInfo_t	1001
unpack_wds_SLQSGetProfileSettings_t	1004
unpack_wds_SLQSGetProfileSettingsV2_t	1004
unpack_wds_SLQSGetRuntimeSettings_t	1005
unpack_wds_SLQSModifyProfile_t	1008

unpack_wds_SLQSSetIPFamilyPreference_t	1009
unpack_wds_SLQSSetPacketSrvStatusCallback_t	1009
unpack_wds_SLQSSetWdsEventCallback_ind_t	1011
unpack_wds_SLQSSetDHCPv4ClientConfig_t	1017
unpack_wds_SLQSSetLoopback_t	1018
unpack_wds_SLQSSetDataSession_t	1019
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	1020
UnPackGetProfileSettingOut	1022
UnPackGetProfileSettingOutV2	1023
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo	1023
UnpackSwiAvmsEventReportConfig	1025
UnpackSwiAvmsEventReportConnectionRequest	1026
UnpackSwiAvmsEventReportDataSessionStatus	1027
UnpackSwiAvmsEventReportHTTPStatus	1027
UnpackSwiAvmsEventReportNotification	1029
UnpackSwiAvmsEventReportPackageID	1030
UnpackSwiAvmsEventReportRegStatus	1030
UnpackSwiAvmsEventReportSessionType	1031
UnpackSwiAvmsEventReportWAMSPParamChange	1031
unpackWdsProfileParam	1032
unpackWdsProfileParamV2	1032
voice_airTimer	1033
voice_allCallsAlphaIDInfo	1033
voice_allCallsDiagInfo	1034
voice_allCallsUUSInfo	1034
voice_alphaIDInfo	1035
voice_arrAlertingPattern	1035
voice_arrAlertingType	1036
voice_arrAlphaID	1037
voice_arrCalledPartyNum	1037
voice_arrCallEndReason	1038
voice_arrCallInfo	1039
voice_arrConnectPartyNum	1039
voice_arrDiagInfo	1040
voice_arrRedirPartyNum	1040
voice_arrRemotePartyName	1041
voice_arrRemotePartyNum	1041
voice_arrSvcOption	1042
voice_arrUUSInfo	1043
voice_burstDTMFInfo	1043
voice_calledPartyInfo	1044
voice_calledPartySubAdd	1045
voice_callerIDInfo	1046
voice_callFwdTypeAndPlan	1047
voice_callFWExtInfo	1048
voice_callFWInfo	1050
voice_callInfo	1051
voice_callingPartyInfo	1052
voice_ccSUPSType	1054
voice_CLIPResp	1055
voice_CLIRResp	1055
voice_CNAPResp	1056
voice_COLPResp	1057
voice_COLRResp	1057
voice_connectNumInfo	1058
voice_CUGInfo	1060
voice_curAMRConfig	1060
voice_diagInfo	1061

voice_DTMFInfo	1062
voice_DTMFLengths	1062
voice_ECTNum	1063
voice_extDispRecInfo	1064
voice_getAllCallInformation	1065
voice_getAllCallRmtPtyName	1065
voice_getAllCallRmtPtyNum	1066
voice_getCallFWExtInfo	1066
voice_getCallFWInfo	1067
voice_lineCtrlInfo	1067
voice_newPwdData	1068
voice_NSSAudioCtrl	1069
voice_peerNumberInfo	1069
voice_prefVoiceSO	1071
voice_redirNumInfo	1073
voice_remotePartyName	1075
voice_remotePartyNum	1075
voice_roamTimer	1076
voice_signalInfo	1077
voice_SUPSInfo	1078
voice_USSDNotificationNetworkInfo	1078
voice_USSInfo	1079
voice_UUSInfo	1079
wds_channelRate	1081
wds_ChannelRateTlv	1081
wds_ConnStatusTlv	1082
wds_currNetworkInfo	1082
wds_DataBearTechTlv	1083
wds_DataULongLongTlv	1084
wds_DataULongTlv	1085
wds_DHCPLeaseOptTlv	1085
wds_DHCPLeaseStateTlv	1086
wds_DHCPOpt	1086
wds_DHCPProfileIdTlv	1087
wds_DHCPv4HWConfig	1087
wds_DHCPv4Option	1088
wds_DHCPv4OptionList	1089
wds_DHCPv4ProfileId	1089
wds_Domain	1090
wds_DomainNameList	1090
wds_DormStatTlv	1091
wds_GPRSQoS	1091
wds_IPv4AdTlv	1092
wds_IPV6AddressInfo	1093
wds_IPV6GWAddressInfo	1093
wds_LastMdmCallEndRsnTlv	1094
wds_PCSCFFQDNAddress	1094
wds_PCSCFFQDNAddressList	1095
wds_PCSCFIPv4ServerAddressList	1095
wds_ProfileIdentifier	1096
wds_profileInfo	1097
wds_RXBytesOKTlv	1097
wds_transferStatInd	1098
wds_TrStatInd	1098
wds_TXBytesOKTlv	1099
wds_UMTSMInQoS	1100
wdsDhcpv4HwConfig	1102
wdsDhcpv4Option	1103

wdsDhcpv4OptionList	1103
wdsDhcpv4ProfileId	1104

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

apdoxypages.c	Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages	1105
audio.h		1105
cat.h		1111
common.h		1114
dms.h		1120
fms.h		1182
ims.h		1186
imsa.h		1198
lite-fw.h		1203
loc.h		1217
nas.h		1240
pds.h		1303
qaGobiApiTableBandClasses.h	Network Access Service API Band Classes table	1325
qaGobiApiTableCallControlReturnReasons.h	Call Control Return Reasons table	1328
qaGobiApiTableCallEndReasons.h	Wireless Data Service Call End Reasons	1329
qaGobiApiTableCarrierCodes.h	Carrier Codes table	1345
qaGobiApiTableCodingScheme.h	Data Coding Scheme	1347
qaGobiApiTableGpsCapabilityCodes.h	Position Determination Service API GPS Capability Codes	1350
qaGobiApiTablePowerModes.h	Device Management Service API Power Modes table	1350
qaGobiApiTableRadioInterfaces.h	Network Access Service API Radio Interfaces table	1351
qaGobiApiTableRegionCodes.h	Region Codes table	1352
qaGobiApiTableServiceOptions.h	Voice Service Options	1352
qaGobiApiTableSupServiceInfoClasses.h	Voice Supplementary Service Information Classes	1355
qaGobiApiTableSwiAudio.h	Swi Audio related tables	1355

qaGobiApiTableSwiOMADMSessionStatus.h	
Session Status table	1356
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h	
Update Complete Status table	1357
qaGobiApiTableVoiceCallEndReasons.h	
Voice Service Call and supplementary services end reasons	1358
qmerrno.h	1365
qos.h	1372
rms.h	1382
sar.h	1384
sms.h	1386
swiaudio.h	1408
swiavms.h	1418
SwiDataTypes.h	
SWI data types	1432
swidms.h	1433
swiloc.h	1441
swioma.h	1443
swiomaext.h	1453
switype_256bit.h	1459
SWIWWANCMAP.h	1461
tmd.h	1461
uim.h	1465
voice.h	1485
wds.h	1517

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

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

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

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"> Part Number of the provided firmware image
---------------------	--

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

<i>coverage</i>	<ul style="list-style-type: none"> • Specifies the region of uncertainty. • Valid values <ul style="list-style-type: none"> – 0 - Not specified – 1 - Altitude uncertainty is valid at the injected horizontal position coordinates only – 2 - Altitude uncertainty applies to the position of the device regardless of horizontal position
-----------------	---

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"> • Indicates the type of the application. <ul style="list-style-type: none"> – 0 - Unknown – 1 - SIM card – 2 - USIM application – 3 - RUIM card – 4 - CSIM application – 5 - ISIM application • Other values are reserved for the future and are to be handled as "Unknown".
----------------	--

<i>appState</i>	<ul style="list-style-type: none"> Indicates the state of the application. <ul style="list-style-type: none"> 0 - Unknown 1 - Detected 2 - PIN1 or UPIN is required 3 - PUK1 or PUK for UPIN is required 4 - Personalization state must be checked 5 - PIN1 is blocked 6 - Illegal 7 - Ready
<i>persoState</i>	<ul style="list-style-type: none"> Indicates the state of the personalization for the application. <ul style="list-style-type: none"> 0 - Unknown 1 - Personalization operation is in progress 2 - Ready 3 - Personalization code is required 4 - PUK for personalization code is required 5 - Permanently blocked
<i>persoFeature</i>	<ul style="list-style-type: none"> Indicates the personalization feature. This applies only when a personalization code is required to deactivate or unblock personalization. <ul style="list-style-type: none"> 0 - GW network personalization 1 - GW network subset personalization 2 - GW service provider personalization 3 - GW corporate personalization 4 - GW UIM personalization 5 - 1X network type 1 personalization 6 - 1X network type 2 personalization 7 - 1X HRPD personalization 8 - 1X service provider personalization 9 - 1X corporate personalization 10 - 1X RUIM personalization 11 - Unknown
<i>persoRetries</i>	<ul style="list-style-type: none"> Indicates the number of retries remaining to disable the personalization.
<i>persoUnblock-Retries</i>	<ul style="list-style-type: none"> Indicates the number of retries remaining to unblock the personalization.
<i>aidLength</i>	<ul style="list-style-type: none"> Number of sets of the following elements. i.e. aidVal If zero(0) then no aidVal information exists.
<i>aidVal[<small>MAX_DESCRIPTION_LENGTH</small>]</i>	<ul style="list-style-type: none"> Application identifier value.

<i>univPin</i>	<ul style="list-style-type: none"> Indicates whether UPIN replaces PIN1. <ul style="list-style-type: none"> 0 - PIN1 is used 1 - UPIN replaces PIN1
<i>pin1State</i>	<ul style="list-style-type: none"> Indicates the state of PIN1. <ul style="list-style-type: none"> 0 - Unknown 1 - Enabled and not verified 2 - Enabled and verified 3 - Disabled 4 - Blocked 5 - Permanently blocked
<i>pin1Retries</i>	<ul style="list-style-type: none"> Indicates the number of retries remaining to verify PIN1.
<i>puk1Retries</i>	<ul style="list-style-type: none"> Indicates the number of retries remaining to unblock PIN1.
<i>pin2State</i>	<ul style="list-style-type: none"> Indicates the state of PIN2. <ul style="list-style-type: none"> 0 - Unknown 1 - Enabled and not verified 2 - Enabled and verified 3 - Disabled 4 - Blocked 5 - Permanently blocked
<i>pin2Retries</i>	<ul style="list-style-type: none"> Indicates the number of retries remaining to verify PIN2.
<i>puk2Retries</i>	<ul style="list-style-type: none"> Indicates the number of retries remaining to unblock PIN2.

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"> • RX pre-compressor static gain
<i>pRXAIG</i>	<ul style="list-style-type: none"> • RX pre-compressor gain selection flag
<i>pRXExpThres</i>	<ul style="list-style-type: none"> • RX expansion threshold
<i>pRXExpSlope</i>	<ul style="list-style-type: none"> • RX expansion slope
<i>pRXComprThres</i>	<ul style="list-style-type: none"> • RX compression threshold
<i>pRXComprSlope</i>	<ul style="list-style-type: none"> • RX compression slope

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 * [pAVRXAVCHeadroom](#)

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"> • AVC variation from nominal sensitivity
<i>pAVRXAVC-Headroom</i>	<ul style="list-style-type: none"> • AVC headroom

8.7.2 Field Documentation

- 8.7.2.1 uint16_t* audio_RXAVCList::pAVRXAVCHeadroom
- 8.7.2.2 uint16_t* audio_RXAVCList::pAVRXAVCSens

8.8 audio_RXPCMIIRFiltr 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_RXPCMIIRFLTR.

Parameters

<i>pFlag</i>	<ul style="list-style-type: none"> • Flag <ul style="list-style-type: none"> – 0x0000 - IIR filter disable – 0xffff - IIR filter enable
<i>pStageCnt</i>	<ul style="list-style-type: none"> • Stage Count <ul style="list-style-type: none"> – 0-4
<i>pStage0Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 0 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage1Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 1 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage2Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 2 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage3Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 3 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage4Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 4 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2

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"> • TX pre-compressor static gain
<i>pTXAIG</i>	<ul style="list-style-type: none"> • TX pre-compressor gain selection flag
<i>pTXExpThres</i>	<ul style="list-style-type: none"> • TX expansion threshold
<i>pTXExpSlope</i>	<ul style="list-style-type: none"> • TX expansion slope
<i>pTXComprThres</i>	<ul style="list-style-type: none"> • TX compression threshold
<i>pTXComprSlope</i>	<ul style="list-style-type: none"> • TX compression slope

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"> • Flag <ul style="list-style-type: none"> – 0x0000 - IIR filter disable – 0xffff - IIR filter enable
<i>pStageCnt</i>	<ul style="list-style-type: none"> • Stage Count <ul style="list-style-type: none"> – 0-4
<i>pStage0Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 0 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2

<i>pStage1Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 1 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage2Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 2 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage3Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 3 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2
<i>pStage4Val</i>	<ul style="list-style-type: none"> • A 20 BYTE sized parameter indicating Stage 4 value <ul style="list-style-type: none"> – A1 – A2 – B0 – B1 – B2

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"> • Unique numeric carrier ID indicating the carrier that the following images belong to
<i>m_nFolderId</i>	<ul style="list-style-type: none"> • Unique numeric folder ID indicating the folder where the images should reside on the host storage.
<i>m_nStorage</i>	<ul style="list-style-type: none"> • Information of storage type • Values <ul style="list-style-type: none"> – 0 - Device – 1 - Host
<i>m_FwImageId</i>	<ul style="list-style-type: none"> • Firmware image ID
<i>m_FwBuildId</i>	<ul style="list-style-type: none"> • Firmware build ID
<i>m_PriImageId</i>	<ul style="list-style-type: none"> • PRI image ID
<i>m_PriBuildId</i>	<ul style="list-style-type: none"> • PRI build ID

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"> • 0x01 - End proactive session command type received from the card • 0x02 - End proactive session internal to ME
----------------------------	--

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"> • 0x00000001 - User Activity Notify • 0x00000002 - Idle Screen Available • 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
-----------------------	---

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"> • 0x01 - Refresh start • 0x02 - Refresh success • 0x03 - Refresh failed

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"> • RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal.
<i>ecio</i>	<ul style="list-style-type: none"> • ECIO value representing negative 0.5 dBm increments, i.e., 2 means -1 dBm (14 means -7 dBm, 63 means -31.5 dBm).

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"> • Current link status <ul style="list-style-type: none"> – 0x01 - DISCONNECTED – 0x02 - CONNECTED – 0xff - Invalid data.
<i>MDMCall-Duration</i>	<ul style="list-style-type: none"> • Call duration in milliseconds. • If the modem connection status is connected, this represent the duration of the current DUN call. • 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"> – 0xffffffffffffff - Invalid data.

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"> • Device Crash Status • 0 - no crash • 1 - crash has occurred
<i>crashInfo[OUT]</i>	<ul style="list-style-type: none"> • See crashInformation

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

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

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"> • Values: <ul style="list-style-type: none"> – 0 - 3GPP – 1 - 3GPP2
--------------------	--

<i>RATMask</i>	RAT Mask
<i>SOMask</i>	SO Mask

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"> • Tlv Present
<i>activationStatus</i>	<ul style="list-style-type: none"> • Service Activation Code <ul style="list-style-type: none"> – 0 - Service not activated – 1 - Service activated – 2 - Activation connecting – 3 - Activation connected – 4 - OTASP security authenticated – 5 - OTASP NAM downloaded – 6 - OTASP MDN downloaded – 7 - OTASP IMSI downloaded – 8 - OTASP PRL downloaded – 9 - OTASP SPC downloaded – 10 - OTASP settings committed

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"> • Maximum Tx channel rate in bits per second
<i>MaxRXChannelRate</i>	<ul style="list-style-type: none"> • Maximum Rx channel rate in bits per second
<i>DataServiceCapability</i>	<ul style="list-style-type: none"> • DMS_DATA_CAP_NONE - 0x00 • DMS_DATA_CAP_CS_ONLY-circuit switched only - 0x01 • DMS_DATA_CAP_PS_ONLY-packet switched only - 0x02 • DMS_DATA_CAP_SIMUL_CS_AND_PS - 0x03 • DMS_DATA_CAP_NONSIMUL_CS_AND_PS - 0x04
<i>SimCapability</i>	<ul style="list-style-type: none"> • SIM_NOT_SUPPORTED - 0x01 • SIM_SUPPORTED - 0x02
<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"> • DMS_RADIO_IF_1X- CDMA2000 1X - 0x01 • DMS_RADIO_IF_1X_EVDO- CDMA2000 HRPD - 0x02 • DMS_RADIO_IF_GSM- GSM - 0x04 • DMS_RADIO_IF_UMTS - 0x05 • DMS_RADIO_IF_LTE - 0x08 • DMS_RADIO_IF_TD - 0x09

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

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

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

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

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

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

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

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

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

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

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

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

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

8.37.2 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"> • Boolean indicating the presence of the TLV in the QMI response
<i>ActiveTimerInd</i>	<ul style="list-style-type: none"> • Active Timer • PSM active timer value (in seconds).

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

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

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

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

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"> • Boolean indicating the presence of the TLV in the QMI response
<i>EnableStateInd</i>	<ul style="list-style-type: none"> • PSM Enable State • Values <ul style="list-style-type: none"> – 0 - PSM is not enabled – 1 - PSM is enabled

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

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"> • Boolean indicating the presence of the TLV in the QMI response
<i>PeriodicUpdate-TimerInd</i>	<ul style="list-style-type: none"> • Periodic Update Timer • PSM periodic update timer value (in seconds).

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

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

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"> • value is 1 if this TLV is present in indication, otherwise 0
<i>TempStat</i>	<ul style="list-style-type: none"> • Temperature state <ul style="list-style-type: none"> – 0 - unknown – 1 - normal – 2 - high (warning) – 3 - high (critical) – 4 - low (critical)
<i>Temperature</i>	<ul style="list-style-type: none"> • temperature in degree celsius

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

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

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

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"> • length of cust_id field
<i>cust_id</i>	<ul style="list-style-type: none"> • Customization ID (Maximum 64 bytes) • NULL terminated ASCII string.
<i>value_length</i>	<ul style="list-style-type: none"> • length of cust_value field
<i>cust_value</i>	<ul style="list-style-type: none"> • Customization Setting Value (Maximum 8 bytes)

<i>cust_attr</i>	<ul style="list-style-type: none"> Customization Setting attribute through QMI <ul style="list-style-type: none"> bit 0: Values: <ul style="list-style-type: none"> 0 - read only 1 - read/write
------------------	--

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"> list type requested
<i>num_instances</i>	<ul style="list-style-type: none"> number of instances of customization setting
<i>custSetting</i>	<ul style="list-style-type: none"> See DMScustSettingInfo for more information

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"> • See DMSgetCustomInput for more information
<i>pCustSetting-Info[OUT]</i>	<ul style="list-style-type: none"> • See DMScustSettingInfo for more information
<i>pCustSetting-List[OUT]</i>	<ul style="list-style-type: none"> • See DMScustSettingList for more information

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"> • Customization ID (Maximum 64 bytes) • NULL terminated ASCII string.
<i>list_type</i>	<ul style="list-style-type: none"> • list type requested

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"> • Instantaneous channel Tx rate in bits per second • 0xffffffff - Invalid data.
<i>CurrChanRxRate</i>	<ul style="list-style-type: none"> • Instantaneous channel Rx rate in bits per second • 0xffffffff - Invalid data.
<i>MaxChanTxRate</i>	<ul style="list-style-type: none"> • maximum Tx rate that can be assigned to the device by the serving system in bits per second • 0xffffffff - Invalid data.
<i>MaxChanRxRate</i>	<ul style="list-style-type: none"> • maximum Rx rate that can be assigned to the device by the serving system in bits per second • 0xffffffff - Invalid data.

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

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

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"> • Type of image 0 - Modem 1 - PRI
<i>imageId</i>	<ul style="list-style-type: none"> • Unique image identifier
<i>buildIdLength</i>	<ul style="list-style-type: none"> • Length of the build ID string (may be zero)
<i>pBuildId</i>	<ul style="list-style-type: none"> • Build ID ANSI string with length provided by the previous field

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"> Index in storage where the image is located(a value of 0xFF indicates that the storage for this type of image is not relevant)
<i>failureCount</i>	<ul style="list-style-type: none"> Number of consecutive write attempts to this storage index that have failed(a value of 0xFF indicates unspecified)
<i>imageID</i>	<ul style="list-style-type: none"> Image unique identifier(max 16 chars.)
<i>buildIDLength</i>	<ul style="list-style-type: none"> Length of the build ID string. If there is no build ID, this field will be 0 and no data will follow.
<i>buildID</i>	<ul style="list-style-type: none"> String containing image build information(Max 100 characters)

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"> Type of image <ul style="list-style-type: none"> 0 - Modem 1 - PRI
------------------	---

<i>maxImages</i>	<ul style="list-style-type: none"> Maximum number of images of this type that may be stored concurrently on the device
<i>executingImage</i>	<ul style="list-style-type: none"> Index (into the next array) of image that is currently executing
<i>imageIDSize</i>	<ul style="list-style-type: none"> The number of elements in the image ID list
<i>imageIDElement</i>	<ul style="list-style-type: none"> Array of ImageIDElement Structure (Max 50 elements) See FMSImageIDElement

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"> The number of elements in the image list
<i>imageIDEntries</i>	<ul style="list-style-type: none"> Array of ImageIDEntries Structure (Max 2 entries)

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"> • The number of elements in the image list
<i>pListEntries</i>	<ul style="list-style-type: none"> • Array of Image entries with size provided by previous field • See FMSImageElement

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"> • RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal.
<i>ecio</i>	<ul style="list-style-type: none"> • ECIO value representing negative 0.5 dBm increments, i.e., 2 means -1 dBm (14 means -7 dBm, 63 means -31.5 dBm).

<i>sinr</i>	<ul style="list-style-type: none"> • SINR level. • SINR is only applicable for 1xEV-DO. • Valid levels are 0 to 8, where the maximum value for: <ul style="list-style-type: none"> – 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 – 0xFF - Not Available
<i>io</i>	<ul style="list-style-type: none"> • Received IO in dBm. • IO is only applicable for 1xEV-DO.

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"> • Image Type • Values: <ul style="list-style-type: none"> – 0 - FW – 1 - configuration
------------------	--

<i>uniqueID</i>	<ul style="list-style-type: none"> Image Unique Identifier (ASCII characters)
<i>buildIDLen</i>	<ul style="list-style-type: none"> Length of build ID string to follow If set to zero, build ID string will be blank
<i>buildID</i>	<ul style="list-style-type: none"> String containing image information(ASCII characters) Maximum length of this string is 255 chars

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"> value is 1 if this TLV is present in indication, otherwise 0
<i>amrMode</i>	Bitmask indicating AMR modes. Values: <ul style="list-style-type: none"> 0x1 - 4.75 kbps 0x2 - 5.15 kbps 0x4 - 5.9 kbps 0x8 - 6.17 kbps 0x10 - 7.4 kbps 0x20 - 7.95 kbps 0x40 - 10.2 kbps 0x80 - 12.2 kbps

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">• value is 1 if this TLV is present in indication, otherwise 0
<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"> • value is 1 if this TLV is present in indication, otherwise 0
<i>amrWBMode</i>	Bitmask indicating AMR WB modes. Values: <ul style="list-style-type: none"> • 0x1 - 6.60 kbps • 0x2 - 8.85 kbps • 0x4 - 12.65 kbps • 0x8 - 14.25 kbps • 0x10 - 15.85 kbps • 0x20 - 18.25 kbps • 0x40 - 19.85 kbps • 0x80 - 23.05 kbps • 0x100 - 23.85 kbps

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"> • value is 1 if this TLV is present in indication, otherwise 0
<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">value is 1 if this TLV is present in indication, otherwise 0
<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">value is 1 if this TLV is present in indication, otherwise 0
<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"> value is 1 if this TLV is present in indication, otherwise 0
<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"> value is 1 if this TLV is present in indication, otherwise 0
<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"> value is 1 if this TLV is present in indication, otherwise 0
-------------------	--

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

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

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

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

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

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

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

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">• value is 1 if this TLV is present in indication, otherwise 0
<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">• value is 1 if this TLV is present in indication, otherwise 0
<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"> • value is 1 if this TLV is present in indication, otherwise 0
<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"> • value is 1 if this TLV is present in indication, otherwise 0
<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">• value is 1 if this TLV is present in indication, otherwise 0
<i>tmrTf</i>	-Non-invite transaction timeout timer, in msec

8.91.2 Field Documentation

8.91.2.1 `uint8_t imsa_TmrTfInfo::TlvPresent`

8.91.2.2 `uint32_t imsa_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">• value is 1 if this TLV is present in indication, otherwise 0
<i>ImsFailErrCode</i>	<ul style="list-style-type: none">• 0 -IMSA_PDP_STATUS_OTHER_FAILURE - Generic failure reason• 1 -IMSA_PDP_STATUS_OPTION_UNSUBSCRIBED - Option is unsubscribed• 2 -IMSA_PDP_STATUS_UNKNOWN_PDP - PDP was unknown

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

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

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

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

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

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

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

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">value is 1 if this TLV is present in indication, otherwise 0
<i>UtRatVal</i>	<ul style="list-style-type: none">Values:0-IMSA_WLAN - IMS service is registered on WLAN1-IMSA_WWAN - IMS service is registered on WWAN2-IMSA_IWLAN - IMS service is registered on interworking WLAN

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

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

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

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

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

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"> • 0xff - Invalid IPv6 address information.
<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"> • Precedence class
<i>delayClass</i>	<ul style="list-style-type: none"> • Delay class
<i>reliabilityClass</i>	<ul style="list-style-type: none"> • Reliability class
<i>peak-Throughput-Class</i>	<ul style="list-style-type: none"> • Peak throughput class
<i>mean-Throughput-Class</i>	<ul style="list-style-type: none"> • Mean throughput class

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"> • One or more uint8_ts describing the profile • NULL pointer - Invalid data. • Bit to check in ParamPresenceMask - 16
<i>pProfilename-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 16
<i>pPDPTYPE</i>	<ul style="list-style-type: none"> • 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"> – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPV6 – 0x03 - PDP-IPV4V6 • Bit to check in ParamPresenceMask - 17
<i>pPdpHdrComp-Type</i>	<ul style="list-style-type: none"> • PDP header compression type <ul style="list-style-type: none"> – 0 - PDP header compression is OFF – 1 - Manufacturer preferred compression – 2 - PDP header compression based on RFC 1144 – 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095 • Bit to check in ParamPresenceMask - 18

<i>pPdpDataComp-Type</i>	<ul style="list-style-type: none"> • PDP data compression type <ul style="list-style-type: none"> – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 3 - V.44 data compression • Bit to check in ParamPresenceMask - 19
<i>pAPNName</i>	<ul style="list-style-type: none"> • Access point name • Bit to check in ParamPresenceMask - 20
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 20
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> • Primary DNS IPv4 Address Preference • Bit to check in ParamPresenceMask - 21
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> • Secondary DNS IPv4 Address Preference • Bit to check in ParamPresenceMask - 22
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> • UMTS Requested QoS • Bit to check in ParamPresenceMask - 23
<i>pUMTSMInQoS</i>	<ul style="list-style-type: none"> • UMTS Minimum QoS • Bit to check in ParamPresenceMask - 24
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> • GPRS Requested QoS • Bit to check in ParamPresenceMask - 25
<i>pGPRS-MinimumQoS</i>	<ul style="list-style-type: none"> • GPRS Minimum QoS • Bit to check in ParamPresenceMask - 26
<i>pUsername</i>	<ul style="list-style-type: none"> • User name • Bit to check in ParamPresenceMask - 27
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 27

<i>pPassword</i>	<ul style="list-style-type: none"> • Password • Bit to check in ParamPresenceMask - 28
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 28
<i>pAuthentication-Pref</i>	<ul style="list-style-type: none"> • Authentication Preference <ul style="list-style-type: none"> – Bit map that indicates the authentication algorithm preference <ul style="list-style-type: none"> * Bit 0 - PAP preference <ul style="list-style-type: none"> • 0 - PAP is never performed • 1 - PAP may be performed * Bit 1 - CHAP preference <ul style="list-style-type: none"> • 0 - CHAP is never performed • 1 - CHAP may be performed * 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. • Bit to check in ParamPresenceMask - 29
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> • IPv4 Address Preference • Bit to check in ParamPresenceMask - 30
<i>pPcscfAddr-UsingPCO</i>	<ul style="list-style-type: none"> • P-CSCF Address using PCO Flag <ul style="list-style-type: none"> – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0 • Bit to check in ParamPresenceMask - 31
<i>pPdpAccess-ConFlag</i>	<ul style="list-style-type: none"> • PDP access control flag <ul style="list-style-type: none"> – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission • Bit to check in ParamPresenceMask - 32
<i>pPcscfAddr-UsingDhcp</i>	<ul style="list-style-type: none"> • P-CSCF address using DHCP <ul style="list-style-type: none"> – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0 • Bit to check in ParamPresenceMask - 33

<i>plmCnFlag</i>	<ul style="list-style-type: none"> IM CN flag <ul style="list-style-type: none"> 1 - (TRUE) implies request IM CN flag for this profile 0 - (FALSE) implies do not request IM CN flag for this profile Bit to check in ParamPresenceMask - 34
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> Traffic Flow Template Bit to check in ParamPresenceMask - 35
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> Traffic Flow Template Bit to check in ParamPresenceMask - 36
<i>pPdpContext</i>	<ul style="list-style-type: none"> PDP context number Bit to check in ParamPresenceMask - 37
<i>pSecondaryFlag</i>	<ul style="list-style-type: none"> PDP context secondary flag <ul style="list-style-type: none"> 1 - (TRUE) implies this is secondary profile 0 - (FALSE) implies this is not secondary profile Bit to check in ParamPresenceMask - 38
<i>pPrimaryID</i>	<ul style="list-style-type: none"> PDP context primary ID function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device Bit to check in ParamPresenceMask - 39
<i>pIPv6AddPref</i>	<ul style="list-style-type: none"> 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 Bit to check in ParamPresenceMask - 40
<i>pUMTSReqQoS-SigInd</i>	<ul style="list-style-type: none"> UMTS requested QoS with Signalling Indication flag Bit to check in ParamPresenceMask - 41
<i>pUMTSMinQoS-SigInd</i>	<ul style="list-style-type: none"> UMTS minimum QoS with Signalling Indication flag Bit to check in ParamPresenceMask - 42
<i>pPrimaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> Primary DNS IPv6 address preference <ul style="list-style-type: none"> 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 Bit to check in ParamPresenceMask - 43

<i>pSecondaryDN-SIPv6addpref</i>	<ul style="list-style-type: none"> • Secondary DNS IPv6 address preference • Bit to check in ParamPresenceMask - 44
<i>paddrAllocation-Pref</i>	<ul style="list-style-type: none"> • DHCP/NAS preference <ul style="list-style-type: none"> – This enumerated value may be used to indicate the address allocation preference <ul style="list-style-type: none"> * 0 - NAS signaling is used for address allocation * 1 - DHCP is used for address allocation • Bit to check in ParamPresenceMask - 45
<i>pQosClassID</i>	<ul style="list-style-type: none"> • 3GPP LTE QoS parameters • Bit to check in ParamPresenceMask - 46
<i>pAPNDisabled-Flag</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: <ul style="list-style-type: none"> – 0 - FALSE(default) – 1 - True • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). • Bit to check in ParamPresenceMask - 47
<i>pPDNInactiv-Timeout</i>	<ul style="list-style-type: none"> • Optional 4 uint8_ts indicating the duration of inactivity timer in seconds • 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 • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). • Bit to check in ParamPresenceMask - 48
<i>pAPNClass</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). • Bit to check in ParamPresenceMask - 49
<i>pSupport-EmergencyCalls</i>	<ul style="list-style-type: none"> • Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled • Can be queried to get current status • Values: <ul style="list-style-type: none"> – 0 - FALSE(default) – 1 - TRUE • Bit to check in ParamPresenceMask - 54

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::plmCnFlag`
- 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"> • Negotiate DNS Server Preference <ul style="list-style-type: none"> – 1 - (TRUE) implies request DNS addresses from the PDSN – 0 - (FALSE) implies do not request DNS addresses from the PDSN – Default value is 1 (TRUE) • Bit to check in ParamPresenceMask - 144
<i>pPppSessClose-TimerDO</i>	<ul style="list-style-type: none"> • PPP Session Close Timer for DO <ul style="list-style-type: none"> – Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down • Bit to check in ParamPresenceMask - 145
<i>pPppSessClose-Timer1x</i>	<ul style="list-style-type: none"> • PPP Session Close Timer for 1X <ul style="list-style-type: none"> – Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down • Bit to check in ParamPresenceMask - 146
<i>pAllowLinger</i>	<ul style="list-style-type: none"> • Allow/disallow lingering of interface <ul style="list-style-type: none"> – 1 -(TRUE) implies allow lingering – 0 -(FALSE) implies do not allow lingering • Bit to check in ParamPresenceMask - 147
<i>pLcpAckTimeout</i>	<ul style="list-style-type: none"> • LCP ACK Timeout <ul style="list-style-type: none"> – Value of LCP ACK Timeout in milliseconds • Bit to check in ParamPresenceMask - 148
<i>pIpcpAck-Timeout</i>	<ul style="list-style-type: none"> • IPCP ACK Timeout <ul style="list-style-type: none"> – Value of IPCP ACK Timeout in milliseconds • Bit to check in ParamPresenceMask - 149
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> • AUTH Timeout <ul style="list-style-type: none"> – Value of Authentication Timeout in milliseconds • Bit to check in ParamPresenceMask - 150

<i>pLcpCreqRetry-Count</i>	<ul style="list-style-type: none"> • LCP Configuration Request Retry Count • Bit to check in ParamPresenceMask - 151
<i>pIpcpCreqRetry-Count</i>	<ul style="list-style-type: none"> • IPCP Configuration Request Retry Count • Bit to check in ParamPresenceMask - 152
<i>pAuthRetry-Count</i>	<ul style="list-style-type: none"> • Authentication Retry Count value • Bit to check in ParamPresenceMask - 153
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> • Authentication Protocol <ul style="list-style-type: none"> – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP • Bit to check in ParamPresenceMask - 154
<i>pUserId</i>	<ul style="list-style-type: none"> • User ID to be used during data network authentication • maximum length allowed is 127 uint8_ts; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value. • Bit to check in ParamPresenceMask - 155
<i>pUserIdSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 155
<i>pAuthPassword</i>	<ul style="list-style-type: none"> • Password to be used during data network authentication; • maximum length allowed is 127 uint8_ts • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value. • Bit to check in ParamPresenceMask - 156
<i>pAuthPassword-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 156
<i>pDataRate</i>	<ul style="list-style-type: none"> • Data Rate Requested <ul style="list-style-type: none"> – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2 • Bit to check in ParamPresenceMask - 157

<i>pAppType</i>	<ul style="list-style-type: none"> • Application Type: <ul style="list-style-type: none"> – 0x00000001 - Default Application Type – 0x00000020 - LBS Application Type – 0x00000040 - Tethered Application Type – This parameter is not used while creating/modifying a profile • Bit to check in ParamPresenceMask - 158
<i>pDataMode</i>	<ul style="list-style-type: none"> • Data Mode to use: <ul style="list-style-type: none"> – 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) – 1 - CDMA Only (1X only) – 2 - HDR Only (1xEV-DO only) – Default is 0 • Bit to check in ParamPresenceMask - 159
<i>pAppPriority</i>	<ul style="list-style-type: none"> • Application Priority <ul style="list-style-type: none"> – Numerical 1 uint8_t value defining the application priority; higher value implies higher priority – This parameter is not used while creating/modifying a profile • Bit to check in ParamPresenceMask - 160
<i>pApnString</i>	<ul style="list-style-type: none"> • String representing the Access Point Name • maximum length allowed is 100 uint8_ts • QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long. • Bit to check in ParamPresenceMask - 161
<i>pApnStringSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 161
<i>pPdnType</i>	<ul style="list-style-type: none"> • Packed Data Network Type Requested: <ul style="list-style-type: none"> – 0 - IPv4 PDN Type – 1 - IPv6 PDN Type – 2 - IPv4 or IPv6 PDN Type – 3 - Unspecified PDN Type (implying no preference) • Bit to check in ParamPresenceMask - 162
<i>plsPcscf-AddressNedded</i>	<ul style="list-style-type: none"> • This boolean value is used to control if PCSCF address is requested from PDSN <ul style="list-style-type: none"> – 1 -(TRUE) implies request for PCSCF value from the PDSN – 0 -(FALSE) implies do not request for PCSCF value from the PDSN • Bit to check in ParamPresenceMask - 163

<i>pPrimaryV4Dns-Address</i>	<ul style="list-style-type: none"> IPv4 Primary DNS address <ul style="list-style-type: none"> The Primary IPv4 DNS address that can be statically assigned to the UE Bit to check in ParamPresenceMask - 164
<i>pSecondaryV4-DnsAddress</i>	<ul style="list-style-type: none"> IPv4 Secondary DNS address <ul style="list-style-type: none"> The Secondary IPv4 DNS address that can be statically assigned to the UE Bit to check in ParamPresenceMask - 165
<i>pPriV6Dns-Address</i>	<ul style="list-style-type: none"> Primary IPv6 DNS address <ul style="list-style-type: none"> The Primary IPv6 DNS address that can be statically assigned to the UE Bit to check in ParamPresenceMask - 166
<i>pSecV6Dns-Address</i>	<ul style="list-style-type: none"> Secondary IPv6 DNS address <ul style="list-style-type: none"> The Secondary IPv6 DNS address that can be statically assigned to the UE Bit to check in ParamPresenceMask - 167
<i>pRATType</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating RAT Type Values: <ul style="list-style-type: none"> 1 - HRPD 2 - EHRPD 3 - HRPD_EHRPD This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). Bit to check in ParamPresenceMask - 168
<i>pAPNEnabled3-GPP2</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating if the APN is disabled/enabled If disabled, the profile can not be used for making data calls Values: <ul style="list-style-type: none"> 0 - Disabled 1 - Enabled(default value) This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). Bit to check in ParamPresenceMask - 169
<i>pPDNInactiv-Timeout3GPP2</i>	<ul style="list-style-type: none"> Optional 4 uint8_ts indicating the duration of inactivity timer in seconds 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 Default value of zero indicates infinite value This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). Bit to check in ParamPresenceMask - 170

<i>pAPNClass3GPP2</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings(). • Bit to check in ParamPresenceMask - 171
-----------------------	---

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"> • One or more uint8_ts describing the profile • NULL pointer - Invalid data. • Bit to check in ParamPresenceMask - 16
<i>pProfileName-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 16
<i>pPDPTYPE</i>	<ul style="list-style-type: none"> • 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"> – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPv6 – 0x03 - PDP-IPv4V6 • Bit to check in ParamPresenceMask - 17

<i>pPdpHdrComp-Type</i>	<ul style="list-style-type: none"> • PDP header compression type <ul style="list-style-type: none"> – 0 - PDP header compression is OFF – 1 - Manufacturer preferred compression – 2 - PDP header compression based on RFC 1144 – 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095 • Bit to check in ParamPresenceMask - 18
<i>pPdpDataComp-Type</i>	<ul style="list-style-type: none"> • PDP data compression type <ul style="list-style-type: none"> – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 3 - V.44 data compression • Bit to check in ParamPresenceMask - 19
<i>pAPNName</i>	<ul style="list-style-type: none"> • Access point name • Bit to check in ParamPresenceMask - 20
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 20
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> • Primary DNS IPv4 Address Preference • Bit to check in ParamPresenceMask - 21
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> • Secondary DNS IPv4 Address Preference • Bit to check in ParamPresenceMask - 22
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> • UMTS Requested QoS • Bit to check in ParamPresenceMask - 23
<i>pUMTSMInQoS</i>	<ul style="list-style-type: none"> • UMTS Minimum QoS • Bit to check in ParamPresenceMask - 24
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> • GPRS Requested QoS • Bit to check in ParamPresenceMask - 25
<i>pGPRS-MinimumQoS</i>	<ul style="list-style-type: none"> • GPRS Minimum QoS • Bit to check in ParamPresenceMask - 26

<i>pUsername</i>	<ul style="list-style-type: none"> • User name • Bit to check in ParamPresenceMask - 27
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 27
<i>pPassword</i>	<ul style="list-style-type: none"> • Password • Bit to check in ParamPresenceMask - 28
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 uint8_ts. • Bit to check in ParamPresenceMask - 28
<i>pAuthentication-Pref</i>	<ul style="list-style-type: none"> • Authentication Preference <ul style="list-style-type: none"> – Bit map that indicates the authentication algorithm preference <ul style="list-style-type: none"> * Bit 0 - PAP preference <ul style="list-style-type: none"> • 0 - PAP is never performed • 1 - PAP may be performed * Bit 1 - CHAP preference <ul style="list-style-type: none"> • 0 - CHAP is never performed • 1 - CHAP may be performed * 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. • Bit to check in ParamPresenceMask - 29
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> • IPv4 Address Preference • Bit to check in ParamPresenceMask - 30
<i>pPcscfAddr-UsingPCO</i>	<ul style="list-style-type: none"> • P-CSCF Address using PCO Flag <ul style="list-style-type: none"> – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0 • Bit to check in ParamPresenceMask - 31
<i>pPdpAccess-ConFlag</i>	<ul style="list-style-type: none"> • PDP access control flag <ul style="list-style-type: none"> – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission • Bit to check in ParamPresenceMask - 32

<i>pPcscfAddr-UsingDhcp</i>	<ul style="list-style-type: none"> • P-CSCF address using DHCP <ul style="list-style-type: none"> – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0 • Bit to check in ParamPresenceMask - 33
<i>plmCnFlag</i>	<ul style="list-style-type: none"> • IM CN flag <ul style="list-style-type: none"> – 1 - (TRUE) implies request IM CN flag for this profile – 0 - (FALSE) implies do not request IM CN flag for this profile • Bit to check in ParamPresenceMask - 34
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> • Traffic Flow Template • Bit to check in ParamPresenceMask - 35
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> • Traffic Flow Template • Bit to check in ParamPresenceMask - 36
<i>pPdpContext</i>	<ul style="list-style-type: none"> • PDP context number • Bit to check in ParamPresenceMask - 37
<i>pSecondaryFlag</i>	<ul style="list-style-type: none"> • PDP context secondary flag <ul style="list-style-type: none"> – 1 - (TRUE) implies this is secondary profile – 0 - (FALSE) implies this is not secondary profile • Bit to check in ParamPresenceMask - 38
<i>pPrimaryID</i>	<ul style="list-style-type: none"> • PDP context primary ID • default value 0xFF if parameter not returned by the device • Bit to check in ParamPresenceMask - 39
<i>pIPv6AddPref</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 40
<i>pUMTSReqQoS-SigInd</i>	<ul style="list-style-type: none"> • UMTS requested QoS with Signalling Indication flag • Bit to check in ParamPresenceMask - 41
<i>pUMTSMInQoS-SigInd</i>	<ul style="list-style-type: none"> • UMTS minimum QoS with Signalling Indication flag • Bit to check in ParamPresenceMask - 42

<i>pPrimaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> Primary DNS IPv6 address preference <ul style="list-style-type: none"> 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 Bit to check in ParamPresenceMask - 43
<i>pSecondaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> Secondary DNS IPv6 address preference Bit to check in ParamPresenceMask - 44
<i>paddrAllocation-Pref</i>	<ul style="list-style-type: none"> DHCP/NAS preference <ul style="list-style-type: none"> This enumerated value may be used to indicate the address allocation preference <ul style="list-style-type: none"> * 0 - NAS signaling is used for address allocation * 1 - DHCP is used for address allocation Bit to check in ParamPresenceMask - 45
<i>pQosClassID</i>	<ul style="list-style-type: none"> 3GPP LTE QoS parameters Bit to check in ParamPresenceMask - 46
<i>pAPNDisabled-Flag</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating if the APN is disabled/enabled If set, the profile can not be used for making data calls Any data call is failed locally Values: <ul style="list-style-type: none"> 0 - FALSE(default) 1 - True Bit to check in ParamPresenceMask - 47
<i>pPDNInactiv-Timeout</i>	<ul style="list-style-type: none"> Optional 4 uint8_ts indicating the duration of inactivity timer in seconds 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 Default value of zero indicates infinite value Bit to check in ParamPresenceMask - 48
<i>pAPNClass</i>	<ul style="list-style-type: none"> Optional 1 uint8_t numeric identifier representing the APN in profile Can be set and queried but is not used by the modem Bit to check in ParamPresenceMask - 49
<i>pAPNBearer</i>	<ul style="list-style-type: none"> Optional 8 Byte numeric APN bearer mask <ul style="list-style-type: none"> 0x0000000000000001 – GSM 0x0000000000000002 – WCDMA 0x0000000000000004 – LTE 0x8000000000000000 – Any Bit to check in ParamPresenceMask - 53

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

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

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

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"> • One or more bytes describing the profile
<i>pProfilename-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 bytes.
<i>pPDPTYPE</i>	<ul style="list-style-type: none"> • 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"> – 0x00 - PDP-IP (IPv4) – 0x01 - PDP-PPP – 0x02 - PDP-IPV6 – 0x03 - PDP-IPV4V6

<i>pPdpHdrComp-Type</i>	<ul style="list-style-type: none"> • PDP header compression type <ul style="list-style-type: none"> – 0 - PDP header compression is OFF – 1 - Manufacturer preferred compression – 2 - PDP header compression based on RFC 1144 – 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095
<i>pPdpDataComp-Type</i>	<ul style="list-style-type: none"> • PDP data compression type <ul style="list-style-type: none"> – 0 - PDP data compression is OFF – 1 - Manufacturer preferred compression – 2 - V.42BIS data compression – 3 - V.44 data compression
<i>pAPNName</i>	<ul style="list-style-type: none"> • Access point name
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 bytes.
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> • Primary DNS IPv4 Address Preference
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> • Secondary DNS IPv4 Address Preference
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> • UMTS Requested QoS
<i>pUMTSMInQoS</i>	<ul style="list-style-type: none"> • UMTS Minimum QoS
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> • GPRS Minimum QoS
<i>pUsername</i>	<ul style="list-style-type: none"> • User name
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes.
<i>pPassword</i>	<ul style="list-style-type: none"> • Password
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes.

<i>pAuthentication-Pref</i>	<ul style="list-style-type: none"> • Authentication Preference <ul style="list-style-type: none"> – Bit map that indicates the authentication algorithm preference <ul style="list-style-type: none"> * Bit 0 - PAP preference <ul style="list-style-type: none"> • 0 - PAP is never performed • 1 - PAP may be performed * Bit 1 - CHAP preference <ul style="list-style-type: none"> • 0 - CHAP is never performed • 1 - CHAP may be performed * 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.
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> • IPv4 Address Preference
<i>pPcscfAddr-UsingPCO</i>	<ul style="list-style-type: none"> • P-CSCF Address using PCO Flag <ul style="list-style-type: none"> – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0
<i>pPdpAccess-ConFlag</i>	<ul style="list-style-type: none"> • PDP access control flag <ul style="list-style-type: none"> – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission
<i>pPcscfAddr-UsingDhcp</i>	<ul style="list-style-type: none"> • P-CSCF address using DHCP <ul style="list-style-type: none"> – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0
<i>plmCnFlag</i>	<ul style="list-style-type: none"> • IM CN flag <ul style="list-style-type: none"> – 1 - (TRUE) implies request IM CN flag for this profile – 0 - (FALSE) implies do not request IM CN flag for this profile
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> • Traffic Flow Template
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> • Traffic Flow Template
<i>pPdpContext</i>	<ul style="list-style-type: none"> • PDP context number
<i>pSecondaryFlag</i>	<ul style="list-style-type: none"> • PDP context secondary flag <ul style="list-style-type: none"> – 1 - (TRUE) implies this is secondary profile – 0 - (FALSE) implies this is not secondary profile

<i>pPrimaryID</i>	<ul style="list-style-type: none"> • PDP context primary ID • function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device
<i>pIPv6AddPref</i>	<ul style="list-style-type: none"> • 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
<i>pUMTSReqQoS-SigInd</i>	<ul style="list-style-type: none"> • UMTS requested QoS with Signalling Indication flag
<i>pUMTSMinQoS-SigInd</i>	<ul style="list-style-type: none"> • UMTS minimum QoS with Signalling Indication flag
<i>pPrimaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> • Primary DNS IPv6 address preference <ul style="list-style-type: none"> – 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
<i>pSecondaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> • Secondary DNS IPv6 address preference
<i>paddrAllocation-Pref</i>	<ul style="list-style-type: none"> • DHCP/NAS preference <ul style="list-style-type: none"> – This enumerated value may be used to indicate the address allocation preference <ul style="list-style-type: none"> * 0 - NAS signaling is used for address allocation * 1 - DHCP is used for address allocation
<i>pQoSClassID</i>	<ul style="list-style-type: none"> • 3GPP LTE QoS parameters
<i>pAPNDisabled-Flag</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If set, the profile can not be used for making data calls • Any data call is failed locally • Values: <ul style="list-style-type: none"> – 0 - FALSE(default) – 1 - True • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
<i>pPDNInactivity-Timeout</i>	<ul style="list-style-type: none"> • Optional 4 Bytes indicating the duration of inactivity timer in seconds • 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 • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().

<i>pAPNClass</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
<i>pSupport-EmergencyCalls</i>	<ul style="list-style-type: none"> • Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled • If set, the profile can be used for making emergency calls • Values: <ul style="list-style-type: none"> – 0 - FALSE(default) – 1 - TRUE

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"> • Negotiate DNS Server Preference <ul style="list-style-type: none"> – 1 - (TRUE)implies request DNS addresses from the PDSN – 0 - (FALSE)implies do not request DNS addresses from the PDSN – Default value is 1 (TRUE)
<i>pPppSessClose-TimerDO</i>	<ul style="list-style-type: none"> • PPP Session Close Timer for DO <ul style="list-style-type: none"> – Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down
<i>pPppSessClose-Timer1x</i>	<ul style="list-style-type: none"> • PPP Session Close Timer for 1X <ul style="list-style-type: none"> – Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down
<i>pAllowLinger</i>	<ul style="list-style-type: none"> • Allow/disallow lingering of interface <ul style="list-style-type: none"> – 1 -(TRUE) implies allow lingering – 0 -(FALSE) implies do not allow lingering
<i>pLcpAckTimeout</i>	<ul style="list-style-type: none"> • LCP ACK Timeout <ul style="list-style-type: none"> – Value of LCP ACK Timeout in milliseconds

<i>pIpcpAck-Timeout</i>	<ul style="list-style-type: none"> • IPCP ACK Timeout <ul style="list-style-type: none"> – Value of IPCP ACK Timeout in milliseconds
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> • AUTH Timeout <ul style="list-style-type: none"> – Value of Authentication Timeout in milliseconds
<i>pLcpCreqRetry-Count</i>	<ul style="list-style-type: none"> • LCP Configuration Request Retry Count
<i>pIpcpCreqRetry-Count</i>	<ul style="list-style-type: none"> • IPCP Configuration Request Retry Count
<i>pAuthRetry-Count</i>	<ul style="list-style-type: none"> • Authentication Retry Count value
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> • Authentication Protocol <ul style="list-style-type: none"> – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP
<i>pUserId</i>	<ul style="list-style-type: none"> • User ID to be used during data network authentication • maximum length allowed is 127 bytes; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
<i>pUserIdSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 bytes.
<i>pAuthPassword</i>	<ul style="list-style-type: none"> • Password to be used during data network authentication; • maximum length allowed is 127 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
<i>pAuthPassword-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 bytes.
<i>pDataRate</i>	<ul style="list-style-type: none"> • Data Rate Requested <ul style="list-style-type: none"> – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2

<i>pAppType</i>	<ul style="list-style-type: none"> Application Type: <ul style="list-style-type: none"> 0x00000001 - Default Application Type 0x00000020 - LBS Application Type 0x00000040 - Tethered Application Type This parameter is not used while creating/modifying a profile
<i>pDataMode</i>	<ul style="list-style-type: none"> Data Mode to use: <ul style="list-style-type: none"> 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) 1 - CDMA Only (1X only) 2 - HDR Only (1xEV-DO only) Default is 0
<i>pAppPriority</i>	<ul style="list-style-type: none"> Application Priority <ul style="list-style-type: none"> Numerical 1 uint8_t value defining the application priority; higher value implies higher priority This parameter is not used while creating/modifying a profile
<i>pApnString</i>	<ul style="list-style-type: none"> String representing the Access Point Name maximum length allowed is 100 bytes QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.
<i>pApnStringSize;</i>	<ul style="list-style-type: none"> This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 bytes.
<i>pPdnType</i>	<ul style="list-style-type: none"> Packed Data Network Type Requested: <ul style="list-style-type: none"> 0 - IPv4 PDN Type 1 - IPv6 PDN Type 2 - IPv4 or IPv6 PDN Type 3 - Unspecified PDN Type (implying no preference)
<i>plsPcscf-AddressNedded</i>	<ul style="list-style-type: none"> This boolean value is used to control if PCSCF address is requested from PDSN <ul style="list-style-type: none"> 1 -(TRUE) implies request for PCSCF value from the PDSN 0 -(FALSE) implies do not request for PCSCF value from the PDSN
<i>pPrimaryV4Dns-Address</i>	<ul style="list-style-type: none"> IPv4 Primary DNS address <ul style="list-style-type: none"> The Primary IPv4 DNS address that can be statically assigned to the UE
<i>pSecondaryV4-DnsAddress</i>	<ul style="list-style-type: none"> IPv4 Secondary DNS address <ul style="list-style-type: none"> The Secondary IPv4 DNS address that can be statically assigned to the UE

<i>pPriV6Dns-Address</i>	<ul style="list-style-type: none"> Primary IPv6 DNS address <ul style="list-style-type: none"> The Primary IPv6 DNS address that can be statically assigned to the UE
<i>pSecV6Dns-Address</i>	<ul style="list-style-type: none"> Secondary IPv6 DNS address <ul style="list-style-type: none"> The Secondary IPv6 DNS address that can be statically assigned to the UE
<i>pRATType</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating RAT Type Values: <ul style="list-style-type: none"> 1 - HRPD 2 - EHRPD 3 - HRPD_EHRPD This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pAPNEnabled3GPP2</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating if the APN is disabled/enabled If disabled, the profile can not be used for making data calls Values: <ul style="list-style-type: none"> 0 - Disabled 1 - Enabled(default value) This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pPDNInactiv-Timeout3GPP2</i>	<ul style="list-style-type: none"> Optional 4 Bytes indicating the duration of inactivity timer in seconds 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 Default value of zero indicates infinite value This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pAPNClass3GPP2</i>	<ul style="list-style-type: none"> Optional 1 uint8_t numeric identifier representing the APN in profile Can be set and queried but is not used by the modem This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

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

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"> • 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
<i>gDIBitRate</i>	<ul style="list-style-type: none"> • Guaranteed DL bit rate

<i>maxDlBitRate</i>	<ul style="list-style-type: none"> Maximum DL bit rate
<i>gUlBitRate</i>	<ul style="list-style-type: none"> Guaranteed UL bit rate
<i>maxUlBitRate</i>	<ul style="list-style-type: none"> Maximum UL bit rate

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"> Filter identifier
-----------------	---

<i>eValid</i>	<ul style="list-style-type: none"> • Evaluation precedence index
<i>ipVersion</i>	<ul style="list-style-type: none"> • IP version number <ul style="list-style-type: none"> – 4 - IPv4 – 6 - IPv6
<i>pSourceIP</i>	<ul style="list-style-type: none"> • Source IP address <ul style="list-style-type: none"> – IPv4 - Fill the first 4 uint8_ts – IPv6 - Fill all the 16 uint8_ts
<i>sourceIPMask</i>	<ul style="list-style-type: none"> • Mask value for the source address
<i>nextHeader</i>	<ul style="list-style-type: none"> • Next header/protocol value
<i>destPortRange-Start</i>	<ul style="list-style-type: none"> • Start value of the destination port range
<i>destPortRange-End</i>	<ul style="list-style-type: none"> • End value of the destination port range
<i>srcPortRange-Start</i>	<ul style="list-style-type: none"> • Start value of the source port range
<i>srcPortRange-End</i>	<ul style="list-style-type: none"> • End value of the source port range
<i>IPSECSPi</i>	<ul style="list-style-type: none"> • IPSEC security parameter index
<i>tosMask</i>	<ul style="list-style-type: none"> • TOS mask (Traffic class for IPv6)
<i>flowLabel</i>	<ul style="list-style-type: none"> • Flow label

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"> • 0x00 - Subscribed • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 0x04 - Background • 0xff - Invalid UMTS Quality Of Service Information.
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> • Maximum uplink bit rate in bits/sec

<i>maxDownlink-Bitrate</i>	<ul style="list-style-type: none"> Maximum downlink bit rate in bits/sec
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> Guaranteed uplink bit rate in bits/sec
<i>grntDownlink-Bitrate</i>	<ul style="list-style-type: none"> Guaranteed downlink bit rate in bits/sec
<i>qosDelivery-Order</i>	<ul style="list-style-type: none"> - Qos delivery order 0x00 - Subscribe 0x01 - delivery order on 0x02 - delivery order off
<i>maxSDUSize</i>	<ul style="list-style-type: none"> Maximum SDU size
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> - SDU error ratio Target value for fraction of SDUs lost or detected as erroneous. 0x00 - Subscribe 0x01 - $1 \cdot 10^{-2}$ 0x02 - $7 \cdot 10^{-3}$ 0x03 - $1 \cdot 10^{-3}$ 0x04 - $1 \cdot 10^{-4}$ 0x05 - $1 \cdot 10^{-5}$ 0x06 - $1 \cdot 10^{-6}$ 0x07 - $1 \cdot 10^{-1}$
<i>resBerRatio</i>	<ul style="list-style-type: none"> - Residual bit error ratio Target value for undetected bit error ratio in in the delivered SDUs. 0x00 - Subscribe 0x01 - $5 \cdot 10^{-2}$ 0x02 - $1 \cdot 10^{-2}$ 0x03 - $5 \cdot 10^{-3}$ 0x04 - $4 \cdot 10^{-3}$ 0x05 - $1 \cdot 10^{-3}$ 0x06 - $1 \cdot 10^{-4}$ 0x07 - $1 \cdot 10^{-5}$ 0x08 - $1 \cdot 10^{-6}$ 0x09 - $1 \cdot 10^{-8}$

<i>deliveryErrSDU</i>	- Delivery of erroneous SDUs <ul style="list-style-type: none"> • Indicates whether SDUs detected as erroneous shall be delivered or not. • 0x00 - Subscribe • 0x01 - $5 \cdot 10^{(-2)}$ • 0x02 - $1 \cdot 10^{(-2)}$ • 0x03 - $5 \cdot 10^{(-3)}$ • 0x04 - $4 \cdot 10^{(-3)}$ • 0x05 - $1 \cdot 10^{(-3)}$ • 0x06 - $1 \cdot 10^{(-4)}$ • 0x07 - $1 \cdot 10^{(-5)}$ • 0x08 - $1 \cdot 10^{(-6)}$ • 0x09 - $1 \cdot 10^{(-8)}$
<i>transferDelay</i>	- Transfer delay (ms) <ul style="list-style-type: none"> • Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.
<i>trafficPriority</i>	- Transfer handling priority <ul style="list-style-type: none"> • Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.

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"> • Contains the UMTS Quality Of Service Information • See LibPackUMTSQoS
<i>SigInd</i>	- Signaling Indication flag <ul style="list-style-type: none"> • TRUE - Signaling indication ON • FALSE - Signaling indication OFF

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"> • GNSS location engine is ready to accept data from sensor. • Values <ul style="list-style-type: none"> • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
---------------------	---

<i>samplesPerBatch</i>	<ul style="list-style-type: none"> • number of samples per batch the GNSS location engine is to receive. • $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ • <i>samplesPerBatch</i> must be a nonzero positive value.
<i>batchPerSec</i>	<ul style="list-style-type: none"> • LTE NAS version minor • Number of sensor-data batches the GNSS location engine is to receive per second. • <i>BatchesPerSecond</i> must be a nonzero positive value.

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

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_BdsSV Struct Reference

Data Fields

- uint16_t [id](#)
- uint8_t [mask](#)

8.121.1 Detailed Description

This structure contains the BDS SV Info

Parameters

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

8.121.2 Field Documentation

8.121.2.1 uint16_t loc_BdsSV::id

8.121.2.2 uint8_t loc_BdsSV::mask

8.122 loc_BdsSVInfo Struct Reference

Data Fields

- uint8_t [len](#)
- [loc_BdsSV](#) * [pSV](#)

8.122.1 Detailed Description

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

Parameters

<i>len</i>	<ul style="list-style-type: none"> Number of sets of the following elements: <ul style="list-style-type: none"> gnssSvId deleteSvInfoMask
<i>pSV</i>	<ul style="list-style-type: none"> Pointer to struct loc_BdsSV. See loc_BdsSV for more information

8.122.2 Field Documentation

8.122.2.1 `uint8_t loc_BdsSVInfo::len`8.122.2.2 `loc_BdsSV* loc_BdsSVInfo::pSV`8.123 `loc_CellDb` Struct Reference

Data Fields

- `uint32_t` [mask](#)

8.123.1 Detailed Description

This structure contains the cell database

Parameters

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

8.123.2 Field Documentation

8.123.2.1 `uint32_t loc_CellDb::mask`8.124 `loc_ClkInfo` Struct Reference

Data Fields

- uint32_t [mask](#)

8.124.1 Detailed Description

This structure contains the clock Info

Parameters

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

8.124.2 Field Documentation

8.124.2.1 `uint32_t loc_ClkInfo::mask`

8.125 `loc_GnssData` Struct Reference

Data Fields

- `uint64_t` [mask](#)

8.125.1 Detailed Description

This structure contains the GNSS data

Parameters

<i>mask</i>	<ul style="list-style-type: none"> • Mask for the GNSS data that is to be deleted • Valid values: <ul style="list-style-type: none"> – QMI_LOC_MASK_DELETE_GPS_SVDIR (0x00000001) - Mask to delete GPS SVDIR – QMI_LOC_MASK_DELETE_GPS_SVSTEER (0x00000002) - Mask to delete GPS SVSTEER – QMI_LOC_MASK_DELETE_GPS_TIME (0x00000004) - Mask to delete GPS time – QMI_LOC_MASK_DELETE_GPS_ALM_CORR (0x00000008) - Mask to delete almanac correlation – QMI_LOC_MASK_DELETE_GLO_SVDIR (0x00000010) - Mask to delete GLONASS SVDIR – QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete GLONASS SVSTEER – QMI_LOC_MASK_DELETE_GLO_TIME (0x00000040) - Mask to delete GLONASS time – QMI_LOC_MASK_DELETE_GLO_ALM_CORR (0x00000080) - Mask to delete GLONASS almanac correlation – QMI_LOC_MASK_DELETE_SBAS_SVDIR (0x00000100) - Mask to delete SBAS SVDIR – QMI_LOC_MASK_DELETE_SBAS_SVSTEER (0x00000200) - Mask to delete SBAS SVSTEER – QMI_LOC_MASK_DELETE_POSITION (0x00000400) - Mask to delete position estimate – QMI_LOC_MASK_DELETE_TIME (0x00000800) - Mask to delete time estimate – QMI_LOC_MASK_DELETE_IONO (0x00001000) - Mask to delete IONO – QMI_LOC_MASK_DELETE_UTC (0x00002000) - Mask to delete UTC estimate – QMI_LOC_MASK_DELETE_HEALTH (0x00004000) - Mask to delete SV health record – QMI_LOC_MASK_DELETE_SADATA (0x00008000) - Mask to delete SADATA – QMI_LOC_MASK_DELETE_RTI (0x00010000) - Mask to delete RTI – QMI_LOC_MASK_DELETE_SV_NO_EXIST (0x00020000) - Mask to delete SV_NO_EXIST – QMI_LOC_MASK_DELETE_FREQ_BIAS_EST (0x00040000) - Mask to delete frequency bias estimate – QMI_LOC_MASK_DELETE_BDS_SVDIR (0x00080000) - Mask to delete BDS SVDIR – QMI_LOC_MASK_DELETE_BDS_SVSTEER (0x00100000) - Mask to delete BDS SVSTEER – QMI_LOC_MASK_DELETE_BDS_TIME (0x00200000) - Mask to delete BDS time – QMI_LOC_MASK_DELETE_BDS_ALM_CORR (0x00400000) - Mask to delete BDS almanac correlation – QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GPS (0x00800000) - Mask to delete GNSS SV blacklist GPS – QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GLO (0x01000000) - Mask to delete GNSS SV blacklist GLO – QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_BDS (0x02000000) - Mask to delete GNSS SV blacklist BDS
-------------	---

8.125.2 Field Documentation

8.125.2.1 uint64_t loc_GnssData::mask

8.126 loc_gpsTime Struct Reference

Data Fields

- uint16_t [gpsWeek](#)
- uint32_t [gpsTimeOfWeekMs](#)

8.126.1 Detailed Description

This structure contains GPS Time info.

Parameters

<i>gpsWeek</i>	<ul style="list-style-type: none"> • Current GPS week as calculated from midnight, Jan. 6, 1980. • Units - Weeks
<i>gpsTimeOf- WeekMs</i>	<ul style="list-style-type: none"> • Amount of time into the current GPS week. • Units - Milliseconds

8.126.2 Field Documentation

8.126.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

8.126.2.2 uint16_t loc_gpsTime::gpsWeek

8.127 loc_gyroAcceptReady Struct Reference

Data Fields

- uint8_t [injectEnable](#)
- uint16_t [samplesPerBatch](#)
- uint16_t [batchPerSec](#)

8.127.1 Detailed Description

This structure contains Gyroscope Accept Ready Info

Parameters

<i>injectEnable</i>	<ul style="list-style-type: none"> • GNSS location engine is ready to accept data from sensor. • Values <ul style="list-style-type: none"> • 0x01 - Ready to accept sensor data • 0x00 - Not ready to accept sensor data
<i>samplesPer- Batch</i>	<ul style="list-style-type: none"> • number of samples per batch the GNSS location engine is to receive. • $\text{samplingFrequency} = \text{samplesPerBatch} * \text{batchesPerSecond}$ • samplesPerBatch must be a nonzero positive value.

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

8.127.2 Field Documentation

8.127.2.1 uint16_t loc_gyroAcceptReady::batchPerSec

8.127.2.2 uint8_t loc_gyroAcceptReady::injectEnable

8.127.2.3 uint16_t loc_gyroAcceptReady::samplesPerBatch

8.128 loc_gyroTempAcceptReady Struct Reference

Data Fields

- uint8_t [injectEnable](#)
- uint16_t [samplesPerBatch](#)
- uint16_t [batchPerSec](#)

8.128.1 Detailed Description

This structure contains Gyroscope Temperature Accept Ready Info

Parameters

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

8.128.2 Field Documentation

8.128.2.1 uint16_t loc_gyroTempAcceptReady::batchPerSec

8.128.2.2 uint8_t loc_gyroTempAcceptReady::injectEnable

8.128.2.3 `uint16_t loc_gyroTempAcceptReady::samplesPerBatch`

8.129 `loc_IPv4Config` Struct Reference

Data Fields

- `uint32_t` [IPv4Addr](#)
- `uint16_t` [IPv4Port](#)

8.129.1 Detailed Description

This structure specifies information regarding the IPv4 address and port.

Parameters

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

8.129.2 Field Documentation

8.129.2.1 `uint32_t loc_IPv4Config::IPv4Addr`

8.129.2.2 `uint16_t loc_IPv4Config::IPv4Port`

8.130 `loc_IPv4Info` Struct Reference

Data Fields

- `uint32_t` [address](#)
- `uint16_t` [port](#)
- `uint8_t` [TlvPresent](#)

8.130.1 Detailed Description

This structure contains LOC Server IPV4 info field.

Parameters

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

8.130.2 Field Documentation

8.130.2.1 `uint32_t loc_IPv4Info::address`

8.130.2.2 `uint16_t loc_IPv4Info::port`

8.130.2.3 `uint8_t loc_IPv4Info::TlvPresent`

8.131 loc_IPv6Config Struct Reference

Data Fields

- `uint16_t IPv6Addr` [8]
- `uint32_t IPv6Port`

8.131.1 Detailed Description

This structure specifies information regarding the IPv6 address and port.

Parameters

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

8.131.2 Field Documentation

8.131.2.1 `uint16_t loc_IPv6Config::IPv6Addr[8]`

8.131.2.2 `uint32_t loc_IPv6Config::IPv6Port`

8.132 loc_IPv6Info Struct Reference

Data Fields

- `uint16_t address` [8]
- `uint32_t port`
- `uint8_t TlvPresent`

8.132.1 Detailed Description

This structure contains LOC Get Server IPV6 info field.

Parameters

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

8.132.2 Field Documentation

8.132.2.1 uint16_t loc_IPv6Info::address[8]

8.132.2.2 uint32_t loc_IPv6Info::port

8.132.2.3 uint8_t loc_IPv6Info::TlvPresent

8.133 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](#)

8.133.1 Detailed Description

This structure contains the Application Information

Parameters

<i>appProviderLength</i>	<ul style="list-style-type: none"> Length of the Application Provider
<i>pAppProvider</i>	<ul style="list-style-type: none"> Application Provider Depends upon the Length of application Provider
<i>appNameLength</i>	<ul style="list-style-type: none"> Length of Application Name
<i>pAppName</i>	<ul style="list-style-type: none"> Application Name Depends upon the Length of application Name

<i>appVersionValid</i>	<ul style="list-style-type: none"> • Specifies whether the application version string contains a valid value • 0x00 (FALSE) Application version string is invalid • 0x01 (TRUE) Application version string is valid
<i>appVersionLength</i>	<ul style="list-style-type: none"> • Length of Application Version
<i>pAppVersion</i>	<ul style="list-style-type: none"> • Application Version • Depends upon the Length of application Version

8.133.2 Field Documentation

8.133.2.1 `uint8_t loc_LocApplicationInfo::appNameLength`

8.133.2.2 `uint8_t loc_LocApplicationInfo::appProviderLength`

8.133.2.3 `uint8_t loc_LocApplicationInfo::appVersionLength`

8.133.2.4 `uint8_t loc_LocApplicationInfo::appVersionValid`

8.133.2.5 `uint8_t* loc_LocApplicationInfo::pAppName`

8.133.2.6 `uint8_t* loc_LocApplicationInfo::pAppProvider`

8.133.2.7 `uint8_t* loc_LocApplicationInfo::pAppVersion`

8.134 loc_precisionDilution Struct Reference

Data Fields

- `uint32_t PDOP`
- `uint32_t HDOP`
- `uint32_t VDOP`

8.134.1 Detailed Description

This structure contains Dilution of precision associated with this position.

Parameters

<i>PDOP</i>	<ul style="list-style-type: none"> • Position dilution of precision. • Range - 1 (highest accuracy) to 50 (lowest accuracy) • PDOP = square root of (Square of HDOP + Square of VDOP²)
<i>HDOP</i>	<ul style="list-style-type: none"> • Horizontal dilution of precision. • Range - 1 (highest accuracy) to 50 (lowest accuracy)

<i>VDOP</i>	<ul style="list-style-type: none"> • Vertical dilution of precision. • Range- 1 (highest accuracy) to 50 (lowest accuracy)
-------------	--

8.134.2 Field Documentation

8.134.2.1 `uint32_t loc_precisionDilution::HDOP`

8.134.2.2 `uint32_t loc_precisionDilution::PDOP`

8.134.2.3 `uint32_t loc_precisionDilution::VDOP`

8.135 `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`

8.135.1 Detailed Description

Contain fields in struct `loc_satelliteInfo`

Parameters

<i>svListLen</i>	<ul style="list-style-type: none"> • number of sets of the following elements: <ul style="list-style-type: none"> – <code>validMask</code> – <code>system</code> – <code>gnssSvid</code> – <code>healthStatus</code> – <code>svStatus</code> – <code>svInfoMask</code> – <code>elevation</code> – <code>azimuth</code> – <code>snr</code>
------------------	--

<i>validMask</i>	<ul style="list-style-type: none"> • Bitmask indicating which of the fields in this TLV are valid. Valid bitmasks: <ul style="list-style-type: none"> – 0x00000001 - VALID_SYSTEM – 0x00000002 - VALID_GNSS_SVID – 0x00000004 - VALID_HEALTH_STATUS – 0x00000008 - VALID_PROCESS_STATUS – 0x00000010 - VALID_SVINFO_MASK – 0x00000020 - VALID_ELEVATION – 0x00000040 - VALID_AZIMUTH – 0x00000080 - VALID_SNR
<i>system</i>	<ul style="list-style-type: none"> • Indicates to which constellation this SV belongs. Valid values: <ul style="list-style-type: none"> – eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite – eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite – eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite – eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite – eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite – eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite
<i>gnssSvid</i>	<ul style="list-style-type: none"> • GNSS SV ID. The GPS and GLONASS SVs can be disambiguated using the system field. Range: <ul style="list-style-type: none"> – FOR GPS: 1 to 32 – FOR GLONASS: 1 to 32 – FOR SBAS: 120 to 151 – for BDS: 201 to 237
<i>healthStatus</i>	<ul style="list-style-type: none"> • health status. Range: 0 - 1 <ul style="list-style-type: none"> – 0 - unhealthy – 1 - healthy
<i>svStatus</i>	<ul style="list-style-type: none"> • SV process status. Valid values: <ul style="list-style-type: none"> – eQMI_LOC_SV_STATUS_IDLE (1) - SV is not being actively processed – eQMI_LOC_SV_STATUS_SEARCH (2) - The system is searching for this SV – eQMI_LOC_SV_STATUS_TRACK (3) - SV is being tracked
<i>svInfoMask</i>	<ul style="list-style-type: none"> • Indicates whether almanac and ephemeris information is available. Valid bitmasks: <ul style="list-style-type: none"> – 0x01 - SVINFO_HAS_EPHEMERIS – 0x02 - SVINFO_HAS_ALMANAC
<i>elevation</i>	<ul style="list-style-type: none"> • SV elevation angle. <ul style="list-style-type: none"> – Units: Degrees – Range: 0 to 90

<i>azimuth</i>	<ul style="list-style-type: none"> SV azimuth angle. <ul style="list-style-type: none"> Units: Degrees Range: 0 to 360
<i>snr</i>	<ul style="list-style-type: none"> SV signal-to-noise ratio <ul style="list-style-type: none"> Units: dB-Hz

8.135.2 Field Documentation

8.135.2.1 float loc_satelliteInfo::azimuth

8.135.2.2 float loc_satelliteInfo::elevation

8.135.2.3 uint16_t loc_satelliteInfo::gnssSvId

8.135.2.4 uint8_t loc_satelliteInfo::healthStatus

8.135.2.5 float loc_satelliteInfo::snr

8.135.2.6 uint8_t loc_satelliteInfo::svInfoMask

8.135.2.7 uint8_t loc_satelliteInfo::svListLen

8.135.2.8 uint32_t loc_satelliteInfo::svStatus

8.135.2.9 uint32_t loc_satelliteInfo::system

8.135.2.10 uint32_t loc_satelliteInfo::validMask

8.136 loc_sensorDataUsage Struct Reference

Data Fields

- uint32_t [usageMask](#)
- uint32_t [aidingIndicatorMask](#)

8.136.1 Detailed Description

This structure contains Sensor Data Usage info.

Parameters

<i>usageMask</i>	<ul style="list-style-type: none"> Specifies which sensors were used in calculating the position in the position report. <ul style="list-style-type: none"> Value <ul style="list-style-type: none"> * 0x00000001 - Accelerometer used * 0x00000002 - Gyroscope used
------------------	--

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

8.136.2 Field Documentation

8.136.2.1 `uint32_t loc_sensorDataUsage::aidingIndicatorMask`

8.136.2.2 `uint32_t loc_sensorDataUsage::usageMask`

8.137 loc_SV Struct Reference

Data Fields

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

8.137.1 Detailed Description

This structure contains the Delete LOC SV Info

Parameters

<i>id</i>	<ul style="list-style-type: none"> LOC SV ID of the satellite whose data is to be deleted Range: <ul style="list-style-type: none"> For GPS: 1 to 32 For SBAS: 33 to 64 For GLONASS: 65 to 96
<i>system</i>	<ul style="list-style-type: none"> Indicates to which constellation this loc_SV belongs Valid values: <ul style="list-style-type: none"> eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite
<i>mask</i>	<ul style="list-style-type: none"> Indicates if the ephemeris or almanac for a satellite is to be deleted Valid values: <ul style="list-style-type: none"> 0x01 - DELETE_EPHEMERIS 0x02 - DELETE_ALMANAC

8.137.2 Field Documentation

8.137.2.1 `uint16_t loc_SV::id`

8.137.2.2 `uint8_t loc_SV::mask`

8.137.2.3 `uint32_t loc_SV::system`

8.138 `loc_SVInfo` Struct Reference

Data Fields

- `uint8_t len`
- `loc_SV * pSV`

8.138.1 Detailed Description

This structure contains the elements of Delete LOC SV Info

Parameters

<i>len</i>	<ul style="list-style-type: none"> • Number of sets of the following elements in struct <code>loc_SV</code>: <ul style="list-style-type: none"> – <code>gnssSvId</code> – <code>system</code> – <code>deleteSvInfoMask</code>
<i>pSV</i>	<ul style="list-style-type: none"> • Pointer to struct <code>loc_SV</code>. See <code>loc_SV</code> for more information

8.138.2 Field Documentation

8.138.2.1 `uint8_t loc_SVInfo::len`

8.138.2.2 `loc_SV * loc_SVInfo::pSV`

8.139 `loc_svUsedforFix` Struct Reference

Data Fields

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

8.139.1 Detailed Description

This structure contains SVs Used to Calculate the Fix.

Parameters

<i>gnssSvUsedList_len</i>	<ul style="list-style-type: none"> • Number of sets of <code>gnssSvUsedList</code>
---------------------------	---

<i>pGnssSvUsed-List</i>	<ul style="list-style-type: none"> • Entry in the list contains the SV ID of a satellite used for calculating this position report. • Following information is associated with each SV ID: <ul style="list-style-type: none"> – GPS - 1 to 32 – SBAS - 33 to 64 – GLONASS - 65 to 96 – QZSS - 193 to 197 – BDS - 201 to 237
-------------------------	---

8.139.2 Field Documentation

8.139.2.1 `uint16_t loc_svUsedforFix::gnssSvUsedList[255]`

8.139.2.2 `uint8_t loc_svUsedforFix::gnssSvUsedList_len`

8.140 loc_urlAddr Struct Reference

Data Fields

- char [address](#) [256]
- uint8_t [TlvPresent](#)

8.140.1 Detailed Description

This structure contains LOC Get Server URL address field.

Parameters

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

8.140.2 Field Documentation

8.140.2.1 `char loc_urlAddr::address[256]`

8.140.2.2 `uint8_t loc_urlAddr::TlvPresent`

8.141 loc_URLAddrInfo Struct Reference

Data Fields

- char [urlAddr](#) [256]

8.141.1 Detailed Description

This structure specifies information regarding the URL.

Parameters

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

8.141.2 Field Documentation

8.141.2.1 `char loc_URLAddrInfo::urlAddr[256]`

8.142 lteSSInfo Struct Reference

Data Fields

- `int8_t rssi`
- `int8_t rsrq`
- `int16_t rsrp`
- `int16_t snr`

8.142.1 Detailed Description

This structure contains the parameters for LTE Signal Strength Information

Parameters

<i>rssi</i>	<ul style="list-style-type: none"> • RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal. <ul style="list-style-type: none"> – For CDMA and UMTS, this indicates forward link pilot Ec – For GSM, this indicates received signal strength
<i>rsrq</i>	<ul style="list-style-type: none"> • RSRQ value in dB (signed integer value) as measured by L1. • Range: -3 to -20 (-3 means -3 dB, -20 means -20 dB).
<i>rsrp</i>	<ul style="list-style-type: none"> • Current RSRP in dBm as measured by L1. • Range: -44 to -140 (-44 means -44 dBm, -140 means -140 dBm).
<i>snr</i>	<ul style="list-style-type: none"> • 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,

8.142.2 Field Documentation

8.142.2.1 int16_t lteSSInfo::rsrp

8.142.2.2 int8_t lteSSInfo::rsrq

8.142.2.3 int8_t lteSSInfo::rssi

8.142.2.4 int16_t lteSSInfo::snr

8.143 messageModeTlv Struct Reference

Data Fields

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

8.143.1 Detailed Description

This structure contains message mode TLV information.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• Boolean indicating the presence of the TLV in the QMI response
<i>MessageMode-Info</i>	<ul style="list-style-type: none">• Message Mode• See sMSMessageModelInfo for more information

8.143.2 Field Documentation

8.143.2.1 [sMSMessageModelInfo](#) [messageModeTlv::MessageModelInfo](#)

8.143.2.2 uint8_t [messageModeTlv::TlvPresent](#)

8.144 nas_acqOrderPref Struct Reference

Data Fields

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

8.144.1 Detailed Description

Contain the Acquisition Order Preference.

Parameters

<i>acqOrdeLen</i>	<ul style="list-style-type: none">• Number of sets of the following elements.
-------------------	---

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

8.144.2 Field Documentation

8.144.2.1 `uint8_t nas_acqOrderPref::acqOrdeLen`

8.144.2.2 `uint8_t* nas_acqOrderPref::pAcqOrder`

8.145 nas_AcqOrderPrefTlv Struct Reference

Data Fields

- `uint8_t acqOrdeLen`
- `uint8_t * pAcqOrder`
- `uint8_t TlvPresent`

8.145.1 Detailed Description

Contain the Acquisition Order Preference.

Parameters

<i>acqOrdeLen</i>	<ul style="list-style-type: none"> Number of sets of the following elements.
<i>pAcqOrder</i>	<ul style="list-style-type: none"> Acquisition order preference to be set. Values: <ul style="list-style-type: none"> 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO) 0x04 - NAS_RADIO_IF_GSM - GSM 0x05 - NAS_RADIO_IF_UMTS - UMTS 0x08 - NAS_RADIO_IF_LTE - LTE 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA 0x0C - NAS_RADIO_IF_NR5G - NR5G
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present

8.145.2 Field Documentation

8.145.2.1 `uint8_t nas_AcqOrderPrefTlv::acqOrdeLen`

8.145.2.2 `uint8_t*` `nas_AcqOrderPrefTlv::pAcqOrder`

8.145.2.3 `uint8_t` `nas_AcqOrderPrefTlv::TlvPresent`

8.146 nas_ActPilotPNElement Struct Reference

Data Fields

- `uint16_t` [ActSetPilotPN](#)
- `uint8_t` [ActSetPilotPNStrength](#)

8.146.1 Detailed Description

This structure contains pilot PN Element.

Parameters

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

8.146.2 Field Documentation

8.146.2.1 `uint16_t` `nas_ActPilotPNElement::ActSetPilotPN`

8.146.2.2 `uint8_t` `nas_ActPilotPNElement::ActSetPilotPNStrength`

8.147 nas_AddCDMASysInfo Struct Reference

Data Fields

- `uint16_t` [geoSysIdx](#)
- `uint16_t` [regPrd](#)

8.147.1 Detailed Description

Structure for storing the Additional CDMA System Information.

Parameters

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

8.147.2 Field Documentation

8.147.2.1 `uint16_t nas_AddCDMASysInfo::geoSysIdx`

8.147.2.2 `uint16_t nas_AddCDMASysInfo::regPrd`

8.148 nas_AddSysInfo Struct Reference

Data Fields

- `uint16_t` [geoSysIdx](#)
- `uint32_t` [cellBroadcastCap](#)

8.148.1 Detailed Description

Structure for storing the Additional GSM and WCDMA System Information.

Parameters

<i>geoSysIdx</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
<i>cellBroadcast-Cap</i>	<ul style="list-style-type: none"> • Cell broadcast capability of the serving system. • When the CDMA registration period is not valid, 0xFFFF is used. <ul style="list-style-type: none"> – NAS_CELL_BROADCAST_CAP_UNKNOWN - Cell broadcast support is unknown – NAS_CELL_BROADCAST_CAP_OFF - Cell broadcast is not supported – NAS_CELL_BROADCAST_CAP_ON - Cell broadcast is supported

8.148.2 Field Documentation

8.148.2.1 `uint32_t nas_AddSysInfo::cellBroadcastCap`

8.148.2.2 `uint16_t nas_AddSysInfo::geoSysIdx`

8.149 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](#)

8.149.1 Detailed Description

Contain the Band Preference.

Parameters

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

8.149.2 Field Documentation

8.149.2.1 uint64_t nas_BandPrefInfoTlv::bits_129_192

8.149.2.2 uint64_t nas_BandPrefInfoTlv::bits_193_256

8.149.2.3 uint64_t nas_BandPrefInfoTlv::bits_1_64

8.149.2.4 uint64_t nas_BandPrefInfoTlv::bits_65_128

8.149.2.5 uint8_t nas_BandPrefInfoTlv::TlvPresent

8.150 nas_BandPrefTlv Struct Reference

Data Fields

- uint64_t [BandPref](#)
- uint8_t [TlvPresent](#)

8.150.1 Detailed Description

Contain the band preference for system selection preferences.

Parameters

<i>BandPref</i>	<ul style="list-style-type: none"> • Bitmask representing the radio technologies that are disabled. • Bit values: • Bit mask representing the band preference • Bit values: <ul style="list-style-type: none"> – Bit 0 - Band Class 0, A-System – Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band – Bit 2 - Band Class 1, all blocks – Bit 3 - Band Class 2 place holder – Bit 4 - Band Class 3, A-System – Bit 5 - Band Class 4, all blocks – Bit 6 - Band Class 5, all blocks – Bit 7 - GSM_DCS_1800 band – Bit 8 - GSM Extended GSM (E-GSM) 900 band – Bit 9 - GSM Primary GSM (P-GSM) 900 band – Bit 10 - Band Class 6 – Bit 11 - Band Class 7 – Bit 12 - Band Class 8 – Bit 13 - Band Class 9 – Bit 14 - Band Class 10 – Bit 15 - Band Class 11 – Bit 16 - GSM 450 band – Bit 17 - GSM 480 band – Bit 18 - GSM 750 band – Bit 19 - GSM 850 band – Bit 20 - GSM Railways GSM 900 Band – Bit 21 - GSM PCS 1900 band – Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band – Bit 23 - WCDMA U.S. PCS 1900 band – Bit 24 - WCDMA Europe and China DCS 1800 band – Bit 25 - WCDMA U.S. 1700 band – Bit 26 - WCDMA U.S. 850 band – Bit 27 - WCDMA Japan 800 band – Bit 28 - Band Class 12 – Bit 29 - Band Class 14 – Bit 30 - Reserved – Bit 31 - Band Class 15 – Bit 32 to 47 - Reserved – Bit 48 - WCDMA Europe 2600 band – Bit 49 - WCDMA Europe and Japan 900 band – Bit 50 - WCDMA Japan 1700 band – Bit 51 to 55 - Reserved – Bit 56 - Band Class 16 – Bit 57 - Band Class 17 – Bit 58 - Band Class 18 – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved
-----------------	---

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

8.150.2 Field Documentation

8.150.2.1 uint64_t nas_BandPrefTlv::BandPref

8.150.2.2 uint8_t nas_BandPrefTlv::TlvPresent

8.151 nas_CallBarringSysInfo Struct Reference

Data Fields

- uint32_t [csBarStatus](#)
- uint32_t [psBarStatus](#)

8.151.1 Detailed Description

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

Parameters

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

8.151.2 Field Documentation

8.151.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

8.151.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

8.152 nas_callBarStatus Struct Reference

Data Fields

- uint32_t [csBarStatus](#)
- uint32_t [psBarStatus](#)

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

8.152.2 Field Documentation

8.152.2.1 uint32_t nas_callBarStatus::csBarStatus

8.152.2.2 uint32_t nas_callBarStatus::psBarStatus

8.153 nas_CDMAChannel Struct Reference

Data Fields

- uint16_t [priChA](#)
- uint16_t [priChB](#)
- uint16_t [secChA](#)
- uint16_t [secChB](#)

8.153.1 Detailed Description

This structure contains CDMA channel.

Parameters

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

8.153.2 Field Documentation

8.153.2.1 uint16_t nas_CDMAChannel::priChA

8.153.2.2 uint16_t nas_CDMAChannel::priChB

8.153.2.3 uint16_t nas_CDMAChannel::secChA

8.153.2.4 uint16_t nas_CDMAChannel::secChB

8.154 nas_CDMAECIOThresh Struct Reference

Data Fields

- uint8_t [CDMAECIOThreshListLen](#)
- int16_t * [pCDMAECIOThreshList](#)

8.154.1 Detailed Description

This structure contains CDMA ECIO threshold related parameters.

Parameters

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

8.154.2 Field Documentation

8.154.2.1 uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen

8.154.2.2 `int16_t*` `nas_CDMAECIOThresh::pCDMAECIOThreshList`

8.155 `nas_CDMAInfo` Struct Reference

Data Fields

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

8.155.1 Detailed Description

This structure contains information about the CDMA Network.

Parameters

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

8.155.2 Field Documentation

8.155.2.1 `uint16_t` `nas_CDMAInfo::baselId`

8.155.2.2 `uint32_t` `nas_CDMAInfo::baseLat`

8.155.2.3 `uint32_t` `nas_CDMAInfo::baseLong`

8.155.2.4 uint16_t nas_CDMAInfo::nid

8.155.2.5 uint16_t nas_CDMAInfo::refpn

8.155.2.6 uint16_t nas_CDMAInfo::sid

8.156 nas_CDMARSSIThresh Struct Reference

Data Fields

- uint8_t [CDMARSSIThreshListLen](#)
- int16_t * [pCDMARSSIThreshList](#)

8.156.1 Detailed Description

This structure contains CDMA RSSI threshold related parameters.

Parameters

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

8.156.2 Field Documentation

8.156.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

8.156.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

8.157 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]

8.157.1 Detailed Description

Structure for storing the CDMA System Information.

Parameters

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

<i>ccsSupportedValid</i>	<ul style="list-style-type: none"> Indicates whether the supported concurrent service is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>ccsSupported</i>	<ul style="list-style-type: none"> Whether concurrent service is supported. Only applicable for CDMA. <ul style="list-style-type: none"> 0x00 - Not supported 0x01 - Supported 0xFF - Not Available
<i>cdmaSysIdValid</i>	<ul style="list-style-type: none"> Indicates whether the CDMA system ID is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>systemID</i>	<ul style="list-style-type: none"> System ID. <ul style="list-style-type: none"> 0xFFFF - Not Available
<i>networkID</i>	<ul style="list-style-type: none"> Network ID. <ul style="list-style-type: none"> 0xFFFF - Not Available
<i>bsInfoValid</i>	<ul style="list-style-type: none"> Indicates whether the base station information is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>baseLat</i>	<ul style="list-style-type: none"> Base station latitude in units of 0.25 sec. Expressed as a two's complement signed number with positive numbers signifying North latitudes. <ul style="list-style-type: none"> 0xFFFFFFFF - Not Available
<i>baseLong</i>	<ul style="list-style-type: none"> Base station longitude in units of 0.25 sec. Expressed as a two's complement signed number with positive numbers signifying East latitudes. <ul style="list-style-type: none"> 0xFFFFFFFF - Not Available
<i>packetZoneValid</i>	<ul style="list-style-type: none"> Indicates whether the packet zone is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available

<i>packetZone</i>	<ul style="list-style-type: none"> • Packet zone (8-bit). <ul style="list-style-type: none"> – 0xFFFF indicates no packet zone. • Only applicable for CDMA.
<i>networkIdValid</i>	<ul style="list-style-type: none"> • Indicates whether the network ID is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>MCC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Country Code. • MCC digits in ASCII characters • MCC wildcard value is returned as {'3', 0xFF, 0xFF}.
<i>MNC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • MNC wildcard value is returned as {'7', 0xFF, 0xFF}.

8.157.2 Field Documentation

8.157.2.1 `uint16_t nas_CDMASysInfo::baseId`

8.157.2.2 `uint32_t nas_CDMASysInfo::baseLat`

8.157.2.3 `uint32_t nas_CDMASysInfo::baseLong`

8.157.2.4 `uint8_t nas_CDMASysInfo::bsInfoValid`

8.157.2.5 `uint8_t nas_CDMASysInfo::bsPRev`

8.157.2.6 `uint8_t nas_CDMASysInfo::bsPRevValid`

8.157.2.7 `uint8_t nas_CDMASysInfo::ccsSupported`

8.157.2.8 `uint8_t nas_CDMASysInfo::ccsSupportedValid`

8.157.2.9 `uint8_t nas_CDMASysInfo::cdmaSysIdValid`

8.157.2.10 `uint8_t nas_CDMASysInfo::isSysPrIMatch`

8.157.2.11 `uint8_t nas_CDMASysInfo::isSysPrIMatchValid`

8.157.2.12 `uint8_t nas_CDMASysInfo::MCC[3]`

8.157.2.13 `uint8_t nas_CDMASysInfo::MNC[3]`

8.157.2.14 `uint16_t nas_CDMASysInfo::networkID`

- 8.157.2.15 uint8_t nas_CDMA SysInfo::networkIdValid
- 8.157.2.16 uint16_t nas_CDMA SysInfo::packetZone
- 8.157.2.17 uint8_t nas_CDMA SysInfo::packetZoneValid
- 8.157.2.18 uint8_t nas_CDMA SysInfo::pRevInUse
- 8.157.2.19 uint8_t nas_CDMA SysInfo::pRevInUseValid
- 8.157.2.20 nas_sysInfoCommon nas_CDMA SysInfo::sysInfoCDMA
- 8.157.2.21 uint16_t nas_CDMA SysInfo::systemID

8.158 nas_CDMA SysInfoExt Struct Reference

Data Fields

- uint16_t [MCC](#)
- uint8_t [imsi_11_12](#)

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

8.158.2 Field Documentation

- 8.158.2.1 uint8_t nas_CDMA SysInfoExt::imsi_11_12
- 8.158.2.2 uint16_t nas_CDMA SysInfoExt::MCC

8.159 nas_cellParams Struct Reference

Data Fields

- uint16_t [pci](#)
- int16_t [rsrq](#)
- int16_t [rsrp](#)
- int16_t [rssi](#)
- int16_t [srxlev](#)

8.159.1 Detailed Description

This structure contains information about the Cell parameters.

Parameters

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

8.159.2 Field Documentation

8.159.2.1 uint16_t nas_cellParams::pci

8.159.2.2 int16_t nas_cellParams::rsrp

8.159.2.3 int16_t nas_cellParams::rsrq

8.159.2.4 int16_t nas_cellParams::rssI

8.159.2.5 int16_t nas_cellParams::srxlev

8.160 nas_ciotAcqOrderPref Struct Reference

Data Fields

- uint8_t [ciotAcqOrderLen](#)
- uint32_t * [pCiotAcqOrder](#)

8.160.1 Detailed Description

Contain the CIOT Acquisition Order Preference.

Parameters

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

8.160.2 Field Documentation

8.160.2.1 uint8_t nas_ciotAcqOrderPref::ciotAcqOrderLen

8.160.2.2 uint32_t* nas_ciotAcqOrderPref::pCiotAcqOrder

8.161 nas_CiotAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t [ciotAcqOrderLen](#)
- uint32_t * [pCiotAcqOrder](#)
- uint8_t [TlvPresent](#)

8.161.1 Detailed Description

Contain the CIOT Acquisition Order Preference.

Parameters

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

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

8.161.2 Field Documentation

8.161.2.1 uint8_t nas_CiotAcqOrderPrefTlv::ciotAcqOrderLen

8.161.2.2 uint32_t* nas_CiotAcqOrderPrefTlv::pCiotAcqOrder

8.161.2.3 uint8_t nas_CiotAcqOrderPrefTlv::TlvPresent

8.162 nas_CiotLteOpModePrefTlv Struct Reference

Data Fields

- uint64_t [ciotLteOpModePref](#)
- uint8_t [TlvPresent](#)

8.162.1 Detailed Description

Contain the CIOT LTE Operational Mode Preference.

Parameters

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

8.162.2 Field Documentation

8.162.2.1 uint64_t nas_CiotLteOpModePrefTlv::ciotLteOpModePref

8.162.2.2 uint8_t nas_CiotLteOpModePrefTlv::TlvPresent

8.163 nas_CommInfo Struct Reference

Data Fields

- int8_t [temperature](#)
- uint8_t [modemMode](#)
- uint8_t [systemMode](#)
- uint8_t [imsRegState](#)

- `uint8_t psState`

8.163.1 Detailed Description

Structure for storing the common information for the device.

Parameters

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

<i>psState</i>	<ul style="list-style-type: none"> • PS Attach State. <ul style="list-style-type: none"> – 0x00 - Attached – 0x01 - Detached – 0xFF - Unknown
----------------	--

8.163.2 Field Documentation

8.163.2.1 `uint8_t nas_CommlInfo::imsRegState`

8.163.2.2 `uint8_t nas_CommlInfo::modemMode`

8.163.2.3 `uint8_t nas_CommlInfo::psState`

8.163.2.4 `uint8_t nas_CommlInfo::systemMode`

8.163.2.5 `int8_t nas_CommlInfo::temperature`

8.164 nas_CSGID Struct Reference

Data Fields

- `uint16_t mcc`
- `uint16_t mnc`
- `uint8_t mncPcsDigits`
- `uint32_t id`
- `uint8_t rat`

8.164.1 Detailed Description

Contain the CSGID.

Parameters

<i>mcc</i>	<ul style="list-style-type: none"> • MCC value. Range 0 to 999
<i>mnc</i>	<ul style="list-style-type: none"> • MNC value. Range 0 to 999
<i>mncPcsDigits</i>	<ul style="list-style-type: none"> • TRUE - MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 • FALSE - MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
<i>id</i>	<ul style="list-style-type: none"> • Closed subscriber group identifier.

<i>rat</i>	<ul style="list-style-type: none">• Radio interface technology of the CSG network. Values:<ul style="list-style-type: none">– 0x04 - RADIO_IF_GSM - GSM– 0x05 - RADIO_IF_UMTS - UMTS– 0x08 - RADIO_IF_LTE - LTE– 0x09 - RADIO_IF_TDSCDMA - TDS
------------	---

8.164.2 Field Documentation

8.164.2.1 uint32_t nas_CSGID::id

8.164.2.2 uint16_t nas_CSGID::mcc

8.164.2.3 uint16_t nas_CSGID::mnc

8.164.2.4 uint8_t nas_CSGID::mncPcsDigits

8.164.2.5 uint8_t nas_CSGID::rat

8.165 nas_CsgId Struct Reference

Data Fields

- uint32_t [csgId](#)
- uint8_t [TlvPresent](#)

8.165.1 Detailed Description

This structure contains csg ID.

Parameters

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

8.165.2 Field Documentation

8.165.2.1 uint32_t nas_CsgId::csgId

8.165.2.2 uint8_t nas_CsgId::TlvPresent

8.166 nas_currentPLMN Struct Reference

Data Fields

- uint16_t [MCC](#)

- uint16_t [MNC](#)
- uint8_t [netDescrLength](#)
- uint8_t [netDescr](#) [255]

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

8.166.2 Field Documentation

8.166.2.1 uint16_t nas_currentPLMN::MCC

8.166.2.2 uint16_t nas_currentPLMN::MNC

8.166.2.3 uint8_t nas_currentPLMN::netDescr[255]

8.166.2.4 uint8_t nas_currentPLMN::netDescrLength

8.167 nas_dataSrvCapabilities Struct Reference

Data Fields

- uint8_t [dataCapabilitiesLen](#)
- uint8_t [dataCapabilities](#) [32]

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

8.167.2 Field Documentation

8.167.2.1 uint8_t nas_dataSrvCapabilities::dataCapabilities[32]

8.167.2.2 uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen

8.168 nas_DataStatusDetail Struct Reference

Data Fields

- uint32_t [IPAddress](#)
- uint8_t [LastErrCode](#)

8.168.1 Detailed Description

This structure contains data status detail

Parameters

<i>IPAddress</i>	<ul style="list-style-type: none"> • IP Address • 0xABCDEFGH - AB.CD.EF.GH • Example: <ul style="list-style-type: none"> – 0x12345678 - 18.52.86.120 0x12=18 0x34=52 0x56=86 0x78=120 • 0xFFFFFFFF - NA
------------------	---

<i>LastErrCode</i>	<ul style="list-style-type: none"> • MIP Error code <ul style="list-style-type: none"> – 0x00 - MIP_RRP_CODE_SUCCESS – 0x01 - MIP_RRP_CODE_SUCCESS_NO_SIM_BINDINGS – 0x40 - MIP_RRP_CODE_FAILURE_FA_REASON_UNSPECIFIED – 0x41 - MIP_RRP_CODE_FAILURE_FA_ADMIN_PROHIBITED – 0x42 - MIP_RRP_CODE_FAILURE_FA_INSUFFICIENT_RESOURCES – 0x43 - MIP_RRP_CODE_FAILURE_FA_MOBILE_NODE_FAILED_AUTH – 0x44 - MIP_RRP_CODE_FAILURE_FA_HA_FAILED_AUTH – 0x45 - MIP_RRP_CODE_FAILURE_FA_REQUESTED_LIFETIME_TOO_LONG – 0x46 - MIP_RRP_CODE_FAILURE_FA_MALFORMED_REQUEST – 0x47 - MIP_RRP_CODE_FAILURE_FA_MALFORMED_REPLY – 0x48 - MIP_RRP_CODE_FAILURE_FA_ENCAPSULATION_UNAVAILABLE – 0x49 - MIP_RRP_CODE_FAILURE_FA_VJHC_UNAVAILABLE – 0x4A - MIP_RRP_CODE_FAILURE_FA_CANT_REV_TUN – 0x4B - MIP_RRP_CODE_FAILURE_FA_MUST_REV_TUN – 0x4C - MIP_RRP_CODE_FAILURE_FA_BAD_TTL – 0x4D - MIP_RRP_CODE_FAILURE_INVALID_COA – 0x4F - MIP_RRP_CODE_FAILURE_FA_DELIVERY_STYLE_NOT_SUPPORTED – 0x59 - MIP_RRP_CODE_FAILURE_FA_VS_REASON – 0x61 - MIP_RRP_CODE_FAILURE_MISSING_NAI – 0x62 - MIP_RRP_CODE_FAILURE_MISSING_HA_ADDR – 0x63 - MIP_RRP_CODE_FAILURE_MISSING_HOMEADDR – 0x68 - MIP_RRP_CODE_FAILURE_UNKNOWN_CHALLENGE – 0x69 - MIP_RRP_CODE_FAILURE_MISSING_CHALLENGE – 0x6A - MIP_RRP_CODE_FAILURE_STALE_CHALLENGE – 0x6B - MIP_RRP_CODE_FAILURE_MISSING_MN_FA – 0x80 - MIP_RRP_CODE_FAILURE_HA_REASON_UNSPECIFIED – 0x81 - MIP_RRP_CODE_FAILURE_HA_ADMIN_PROHIBITED – 0x82 - MIP_RRP_CODE_FAILURE_HA_INSUFFICIENT_RESOURCES – 0x83 - MIP_RRP_CODE_FAILURE_HA_MOBILE_NODE_FAILED_AUTH – 0x84 - MIP_RRP_CODE_FAILURE_HA_FA_FAILED_AUTH – 0x85 - MIP_RRP_CODE_FAILURE_HA_REG_ID_MISMATCH – 0x86 - MIP_RRP_CODE_FAILURE_HA_MALFORMED_REQUEST – 0x88 - MIP_RRP_CODE_FAILURE_UNKNOWN_HA – 0x89 - MIP_RRP_CODE_FAILURE_HA_CANT_REV_TUN – 0x8A - MIP_RRP_CODE_FAILURE_HA_MUST_REV_TUN – 0x8B - MIP_RRP_CODE_FAILURE_HA_ENCAPSULATION_UNAVAILABLE – 0x8F - MIP_RRP_CODE_FAILURE_REDIRECTED_HA – 0x90 - MIP_RRP_CODE_FAILURE_HA_BAD_AAA_AUTH – 0xFF - NA
--------------------	--

8.168.2 Field Documentation

8.168.2.1 `uint32_t nas_DataStatusDetail::IPAddress`

8.168.2.2 `uint8_t nas_DataStatusDetail::LastErrCode`

8.169 nas_detailSvcInfo Struct Reference

Data Fields

- uint8_t [srvStatus](#)
- uint8_t [srvCapability](#)
- uint8_t [hdrSrvStatus](#)
- uint8_t [hdrHybrid](#)
- uint8_t [isSysForbidden](#)

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

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

8.169.2 Field Documentation

8.169.2.1 `uint8_t nas_detailSvcInfo::hdrHybrid`

8.169.2.2 `uint8_t nas_detailSvcInfo::hdrSrvStatus`

8.169.2.3 `uint8_t nas_detailSvcInfo::isSysForbidden`

8.169.2.4 `uint8_t nas_detailSvcInfo::srvCapability`

8.169.2.5 `uint8_t nas_detailSvcInfo::srvStatus`

8.170 nas_DeviceConfigDetail Struct Reference

Data Fields

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

8.170.1 Detailed Description

This structure contains device configure detail.

Parameters

<i>Technology</i>	<ul style="list-style-type: none"> Current technology in use <ul style="list-style-type: none"> 0x00 - 1x RTT 0x01 - EVDO Rev 0 0x02 - EVDO Rev A 0xFF - NA
<i>QLIC</i>	<ul style="list-style-type: none"> Quasi Linear Interference Cancellation <ul style="list-style-type: none"> 0x00 - Not supported 0x01 - Supported
<i>Chipset</i>	<ul style="list-style-type: none"> Qualcomm platform <ul style="list-style-type: none"> 0x4E - MDM6200 0x4F - MDM6600 0xFF - NA

<i>HWVersion</i>	<ul style="list-style-type: none"> Hardware version <ul style="list-style-type: none"> 0x00 - BSHWREV0 0x01 - BSHWREV1 0x02 - BSHWREV2 0x03 - BSHWREV3 0x04 - BSHWREVMAX 0xFF - BSHWREVUNKNOWN
------------------	--

8.170.2 Field Documentation

8.170.2.1 `uint8_t nas_DeviceConfigDetail::Chipset`

8.170.2.2 `uint8_t nas_DeviceConfigDetail::HWVersion`

8.170.2.3 `uint8_t nas_DeviceConfigDetail::QLIC`

8.170.2.4 `uint8_t nas_DeviceConfigDetail::Technology`

8.171 nas_dirNum Struct Reference

Data Fields

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

8.171.1 Detailed Description

This structure contains the parameters for Directory Number Information

Parameters

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

8.171.2 Field Documentation

8.171.2.1 `uint8_t nas_dirNum::dirNum[255]`

8.171.2.2 `uint8_t nas_dirNum::dirNumLen`

8.172 nas_DRCParams Struct Reference

Data Fields

- uint8_t [DRCValue](#)
- uint8_t [DRCCover](#)

8.172.1 Detailed Description

This structure contains DRC information.

Parameters

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

8.172.2 Field Documentation

8.172.2.1 uint8_t nas_DRCParams::DRCCover

8.172.2.2 uint8_t nas_DRCParams::DRCValue

8.173 nas_ecioListElement Struct Reference

Data Fields

- int16_t [ecio](#)
- uint8_t [radioIf](#)

8.173.1 Detailed Description

This structure contains the ECIO Information

Parameters

<i>ecio</i>	<ul style="list-style-type: none"> • ECIO value in dBm
<i>radioIf</i>	<ul style="list-style-type: none"> • Radio interface technology of the signal being measured <ul style="list-style-type: none"> – 0x00 - RADIO_IF_NO_SVC - None (no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000® 1X – 0x02 - RADIO_IF_CDMA_1xEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS

8.173.2 Field Documentation

8.173.2.1 int16_t nas_ecioListElement::ecio

8.173.2.2 uint8_t nas_ecioListElement::radiolf

8.174 nas_ECIOThresh Struct Reference

Data Fields

- uint8_t [ECIOThresListLen](#)
- int16_t * [pECIOThresList](#)

8.174.1 Detailed Description

This sturcture contain ECIO threshold list.

Parameters

<i>ECIOThresList-Len</i>	<ul style="list-style-type: none"> • Length of the ECIO threshold list parameter to follow
<i>pECIOThresList</i>	<ul style="list-style-type: none"> • 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"> – Each ECIO threshold value is a signed 2 byte value – Each ECIO threshold value increments in negative 0.5 dB, e.g., an ECIO threshold value of 2 means -1dB – Maximum number of threshold values is 16 – At least one value must be specified – Threshold values specified above are used for all RATs

8.174.2 Field Documentation

8.174.2.1 uint8_t nas_ECIOThresh::ECIOThresListLen

8.174.2.2 int16_t* nas_ECIOThresh::pECIOThresList

8.175 nas_EdrxCiotLteMode Struct Reference

Data Fields

- uint32_t [lteOpMode](#)
- uint8_t [TlvPresent](#)

8.175.1 Detailed Description

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

Parameters

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

8.175.2 Field Documentation

8.175.2.1 uint32_t nas_EdrxCiotLteMode::lteOpMode

8.175.2.2 uint8_t nas_EdrxCiotLteMode::TlvPresent

8.176 nas_EdrxCycleLength Struct Reference

Data Fields

- uint8_t [cycleLength](#)
- uint8_t [TlvPresent](#)

8.176.1 Detailed Description

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

Parameters

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

8.176.2 Field Documentation

8.176.2.1 uint8_t nas_EdrxCycleLength::cycleLength

8.176.2.2 uint8_t nas_EdrxCycleLength::TlvPresent

8.177 nas_EdrxEnableType Struct Reference

Data Fields

- uint8_t [edrxEnabled](#)
- uint8_t [TlvPresent](#)

8.177.1 Detailed Description

This structure contains the parameters for eDRX Enabled Indicator Type.

Parameters

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

8.177.2 Field Documentation

8.177.2.1 `uint8_t nas_EdrxEnableType::edrxEnabled`

8.177.2.2 `uint8_t nas_EdrxEnableType::TlvPresent`

8.178 nas_EdrxPagingTimeWindow Struct Reference

Data Fields

- `uint8_t edrxPtw`
- `uint8_t TlvPresent`

8.178.1 Detailed Description

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

Parameters

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

8.178.2 Field Documentation

8.178.2.1 `uint8_t nas_EdrxPagingTimeWindow::edrxPtw`

8.178.2.2 `uint8_t nas_EdrxPagingTimeWindow::TlvPresent`

8.179 nas_EdrxRatType Struct Reference

Data Fields

- `uint8_t edrxRatType`
- `uint8_t TlvPresent`

8.179.1 Detailed Description

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

Parameters

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

8.179.2 Field Documentation

8.179.2.1 `uint8_t nas_EdrxRatType::edrxRatType`

8.179.2.2 `uint8_t nas_EdrxRatType::TlvPresent`

8.180 nas_EmerModeTlv Struct Reference

Data Fields

- `uint8_t EmerMode`
- `uint8_t TlvPresent`

8.180.1 Detailed Description

Contain the emergency Mode parameters for system selection preferences.

Parameters

<i>EmerMode</i>	<ul style="list-style-type: none"> • Emergency Mode • Values: <ul style="list-style-type: none"> – 0 - OFF (normal) – 1 - ON (Emergency)
-----------------	---

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

8.180.2 Field Documentation

8.180.2.1 uint8_t nas_EmerModeTlv::EmerMode

8.180.2.2 uint8_t nas_EmerModeTlv::TlvPresent

8.181 nas_errorRateListElement Struct Reference

Data Fields

- uint16_t [errorRate](#)
- uint8_t [radiolf](#)

8.181.1 Detailed Description

This structure contains the Error Rate Information

Parameters

<i>errorRate</i>	<ul style="list-style-type: none"> • Error rate value corresponds to the RAT that is currently registered. <ul style="list-style-type: none"> – For CDMA, the error rate reported is Frame Error Rate: <ul style="list-style-type: none"> * Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% * A value of 0xFFFF indicates that the error rate is unknown or unavailable – For HDR, the error rate reported is Packet Error Rate: <ul style="list-style-type: none"> * Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3% * A value of 0xFFFF indicates that the error rate is unknown or unavailable – For GSM, the error rate reported is Bit Error Rate: <ul style="list-style-type: none"> * Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 The reported value divided by 100 gives the error rate as an RxQual value, e.g., a value of 300 represents an RxQual value of 3. * A value of 25500 indicates No Data – For WCDMA, the error rate reported is Block Error Rate (BLER): <ul style="list-style-type: none"> * Valid values are 1 to 10000 * The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3%. * A value of 0 indicates No Data
------------------	--

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

8.181.2 Field Documentation

8.181.2.1 uint16_t nas_errorRateListElement::errorRate

8.181.2.2 uint8_t nas_errorRateListElement::radioIf

8.182 nas_ForbiddenNetworks3GPP Struct Reference

Data Fields

- uint16_t [forbiddenNwInstLen](#)
- uint16_t [MCC](#) [255]
- uint16_t [MNC](#) [255]
- uint8_t [TlvPresent](#)

8.182.1 Detailed Description

This structure contains the ForbiddenNetworks3GPP response parameters.

Parameters

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

8.182.2 Field Documentation

8.182.2.1 uint16_t nas_ForbiddenNetworks3GPP::forbiddenNwInstLen

8.182.2.2 uint16_t nas_ForbiddenNetworks3GPP::MCC[255]

8.182.2.3 uint16_t nas_ForbiddenNetworks3GPP::MNC[255]

8.182.2.4 uint8_t nas_ForbiddenNetworks3GPP::TlvPresent

8.183 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]

8.183.1 Detailed Description

This structure contains information about the GERAN Network.

Parameters

<i>cellID</i>	<ul style="list-style-type: none"> • Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
<i>plmn[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrcellID is not present.
<i>lac</i>	<ul style="list-style-type: none"> • Location area code. • This field is ignored when nmrcellID is not present. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>arfcn</i>	<ul style="list-style-type: none"> • Absolute RF channel number. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>bsic</i>	<ul style="list-style-type: none"> • Base station identity code. <ul style="list-style-type: none"> – 0xFF - Not Available

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

8.183.2 Field Documentation

8.183.2.1 uint16_t nas_GERANInfo::arfcn

8.183.2.2 uint8_t nas_GERANInfo::bsic

8.183.2.3 uint32_t nas_GERANInfo::cellID

8.183.2.4 nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]

8.183.2.5 uint16_t nas_GERANInfo::lac

8.183.2.6 uint8_t nas_GERANInfo::nmlInst

8.183.2.7 uint8_t nas_GERANInfo::plmn[3]

8.183.2.8 uint16_t nas_GERANInfo::rxLev

8.183.2.9 uint32_t nas_GERANInfo::timingAdvance

8.184 nas_geranInstInfo Struct Reference

Data Fields

- uint16_t [geranArfcn](#)

- uint8_t [geranBsicNcc](#)
- uint8_t [geranBsicBcc](#)
- int16_t [geranRssi](#)

8.184.1 Detailed Description

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

Parameters

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

8.184.2 Field Documentation

8.184.2.1 uint16_t nas_geranInstInfo::geranArfcn

8.184.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

8.184.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

8.184.2.4 int16_t nas_geranInstInfo::geranRssi

8.185 nas_gsmCellInfo Struct Reference

Data Fields

- uint16_t [arfcn](#)
- uint8_t [band1900](#)
- uint8_t [cellIdValid](#)
- uint8_t [bsicId](#)
- int16_t [rssi](#)
- int16_t [srxlev](#)

8.185.1 Detailed Description

This structure contains information about the GSM Cell.

Parameters

<i>arfcn</i>	<ul style="list-style-type: none"> • GSM frequency being reported. • Range: 0 to 1023.
<i>band1900</i>	<ul style="list-style-type: none"> • Band indicator for the GSM ARFCN • This field is only valid if arfcn is in the overlapping region. • If TRUE and the cell is in the overlapping region, the ARFCN is on the 1900 band. • If FALSE, it is on the 1800 band.
<i>cellIdValid</i>	<ul style="list-style-type: none"> • Flag indicating whether the base station identity code ID is valid.
<i>bsicId</i>	<ul style="list-style-type: none"> • Base station identity code ID, including base station color code and network color code. • The lower 6 bits can be set to any value.
<i>rsSI</i>	<ul style="list-style-type: none"> • Measured RSSI value in 1/10 dB. • Range: -200.0 dB to 0
<i>srxlev</i>	<ul style="list-style-type: none"> • Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.185.2 Field Documentation

8.185.2.1 uint16_t nas_gsmCellInfo::arfcn

8.185.2.2 uint8_t nas_gsmCellInfo::band1900

8.185.2.3 uint8_t nas_gsmCellInfo::bsicId

8.185.2.4 uint8_t nas_gsmCellInfo::cellIdValid

8.185.2.5 int16_t nas_gsmCellInfo::rsSI

8.185.2.6 int16_t nas_gsmCellInfo::srxlev

8.186 nas_GSMRSSIThresh Struct Reference

Data Fields

- uint8_t [GSMRSSIThreshListLen](#)
- int16_t * [pGSMRSSIThreshList](#)

8.186.1 Detailed Description

This structure contains GSM RSSI threshold related parameters.

Parameters

<i>GSMRSSI- ThreshListLen</i>	<ul style="list-style-type: none"> Length of the GSM RSSI threshold list parameter to follow
<i>pGSMRSSI- ThreshList</i>	<ul style="list-style-type: none"> Array of RSSI thresholds (in units of 0.1 dBm) Maximum of 32 values Range for RSSI values: -111 to -48 (in dBm)

8.186.2 Field Documentation

8.186.2.1 uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen

8.186.2.2 int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList

8.187 nas_GSMsrvStatusInfo Struct Reference

Data Fields

- uint8_t [srvStatus](#)
- uint8_t [trueSrvStatus](#)
- uint8_t [isPrefDataPath](#)

8.187.1 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"> Service status of the system. <ul style="list-style-type: none"> 0x00 - No service 0x01 - Limited service 0x02 - Service 0x03 - Limited regional service 0x04 - Power save 0xFF - Not Available
<i>trueSrvStatus</i>	<ul style="list-style-type: none"> True service status of the system. Not applicable to CDMA/HDR. <ul style="list-style-type: none"> 0x00 - No service 0x01 - Limited service 0x02 - Service 0x03 - Limited regional service 0x04 - Power save 0xFF - Not Available

<i>isPrefDataPath</i>	<ul style="list-style-type: none"> Whether the RAT is the preferred data path. <ul style="list-style-type: none"> 0x00 - Not preferred 0x01 - Preferred 0xFF - Not Available
-----------------------	---

8.187.2 Field Documentation

8.187.2.1 `uint8_t nas_GSMSrvStatusInfo::isPrefDataPath`

8.187.2.2 `uint8_t nas_GSMSrvStatusInfo::srvStatus`

8.187.2.3 `uint8_t nas_GSMSrvStatusInfo::trueSrvStatus`

8.188 nas_GSMSysInfo Struct Reference

Data Fields

- [nas_sysInfoCommon sysInfoGSM](#)
- `uint8_t lacValid`
- `uint16_t lac`
- `uint8_t cellIdValid`
- `uint32_t cellId`
- `uint8_t regRejectInfoValid`
- `uint8_t rejectSrvDomain`
- `uint8_t rejCause`
- `uint8_t networkIdValid`
- `uint8_t MCC` [3]
- `uint8_t MNC` [3]
- `uint8_t egprsSuppValid`
- `uint8_t egprsSupp`
- `uint8_t dtmSuppValid`
- `uint8_t dtmSupp`

8.188.1 Detailed Description

Structure for storing the GSM System Information.

Parameters

<i>sysInfoGSM</i>	<ul style="list-style-type: none"> See nas_sysInfoCommon for more information.
<i>lacValid</i>	<ul style="list-style-type: none"> Indicates whether the location area code is valid.. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available

<i>lac</i>	<ul style="list-style-type: none"> • Location area code. • Only applies to 3GPP. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>cellIdValid</i>	<ul style="list-style-type: none"> • Indicates whether the cell ID is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>cellId</i>	<ul style="list-style-type: none"> • Cell ID. <ul style="list-style-type: none"> – 0xFFFFFFFF - Not Available
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> • Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> • Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
<i>rejCause</i>	<ul style="list-style-type: none"> • 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"> – 0xFF - Not Available
<i>networkIdValid</i>	<ul style="list-style-type: none"> • Indicates whether the network ID is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>MCC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Country Code. • MCC digits in ASCII characters
<i>MNC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • 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.

<i>egprsSuppValid</i>	<ul style="list-style-type: none"> Indicates whether the EGPRS support is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>egprsSupp</i>	<ul style="list-style-type: none"> EGPRS support indication. Only applicable for GSM. <ul style="list-style-type: none"> 0x00 - Not available 0x01 - Available 0xFF - Not Available
<i>dtmSuppValid</i>	<ul style="list-style-type: none"> Indicates whether Dual Transfer mode support is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>dtmSupp</i>	<ul style="list-style-type: none"> Dual Transfer mode support indication. Only applicable for GSM. <ul style="list-style-type: none"> 0x00 - Not available 0x01 - Available 0xFF - Not Available

8.188.2 Field Documentation

8.188.2.1 `uint32_t nas_GSMsSysInfo::cellId`

8.188.2.2 `uint8_t nas_GSMsSysInfo::cellIdValid`

8.188.2.3 `uint8_t nas_GSMsSysInfo::dtmSupp`

8.188.2.4 `uint8_t nas_GSMsSysInfo::dtmSuppValid`

8.188.2.5 `uint8_t nas_GSMsSysInfo::egprsSupp`

8.188.2.6 `uint8_t nas_GSMsSysInfo::egprsSuppValid`

8.188.2.7 `uint16_t nas_GSMsSysInfo::lac`

8.188.2.8 `uint8_t nas_GSMsSysInfo::lacValid`

8.188.2.9 `uint8_t nas_GSMsSysInfo::MCC[3]`

8.188.2.10 `uint8_t nas_GSMsSysInfo::MNC[3]`

8.188.2.11 `uint8_t nas_GSMsSysInfo::networkIdValid`

8.188.2.12 `uint8_t nas_GSMsSysInfo::regRejectInfoValid`

8.188.2.13 uint8_t nas_GSMsysInfo::rejCause

8.188.2.14 uint8_t nas_GSMsysInfo::rejectSrvDomain

8.188.2.15 nas_sysInfoCommon nas_GSMsysInfo::sysInfoGSM

8.189 nas_GWAcqOrderPrefTlv Struct Reference

Data Fields

- uint32_t [GWAcqOrderPref](#)
- uint8_t [TlvPresent](#)

8.189.1 Detailed Description

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

Parameters

<i>GWAcqOrder-Pref</i>	<ul style="list-style-type: none">• Parameter indicating GSM/WCDMA Acquisition order Preference• Values:<ul style="list-style-type: none">– 0x00 - Automatic– 0x01 - GSM then WCDMA– 0x02 - WCDMA then GSM
<i>TlvPresent</i>	<ul style="list-style-type: none">• Tlv Present

8.189.2 Field Documentation

8.189.2.1 uint32_t nas_GWAcqOrderPrefTlv::GWAcqOrderPref

8.189.2.2 uint8_t nas_GWAcqOrderPrefTlv::TlvPresent

8.190 nas_HDRECIOTthresh Struct Reference

Data Fields

- uint8_t [HDRECIOTthreshListLen](#)
- int16_t * [pHDRECIOTthreshList](#)

8.190.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

Parameters

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

<i>pHDRECIO- ThreshList</i>	<ul style="list-style-type: none"> • Array of ECIO thresholds (in units of 0.1 dB) • Maximum of 32 values • Range for ECIO values: -31.5 to 0 (in dB).
---------------------------------	---

8.190.2 Field Documentation

8.190.2.1 `uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen`

8.190.2.2 `int16_t* nas_HDRECIOThresh::pHDRECIOThreshList`

8.191 nas_HDRIOThresh Struct Reference

Data Fields

- `uint8_t HDRIOThreshListLen`
- `int16_t * pHDRIOThreshList`

8.191.1 Detailed Description

This structure contains HDR IO threshold related parameters.

Parameters

<i>HDRIOThresh- ListLen</i>	<ul style="list-style-type: none"> • Length of the HDR IO threshold list parameter to follow
<i>pHDRIOThresh- List</i>	<ul style="list-style-type: none"> • Array of IO thresholds (in units of 0.1 dBm) • Maximum of 32 values • Range for IO values: -128 to -13 (in dBm).

8.191.2 Field Documentation

8.191.2.1 `uint8_t nas_HDRIOThresh::HDRIOThreshListLen`

8.191.2.2 `int16_t* nas_HDRIOThresh::pHDRIOThreshList`

8.192 nas_HDRPersonality_Ind_Data Struct Reference

Data Fields

- `uint16_t * pCurrentPersonality`
- `uint8_t * pPersonalityListLength`
- `nas_protocolSubtypeElement * pProtocolSubtypeElement`

8.192.1 Detailed Description

Elements for HDR Personality indication.

Parameters

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

8.192.2 Field Documentation

8.192.2.1 uint16_t* nas_HDRPersonality_Ind_Data::pCurrentPersonality

8.192.2.2 uint8_t* nas_HDRPersonality_Ind_Data::pPersonalityListLength

8.192.2.3 nas_protocolSubtypeElement* nas_HDRPersonality_Ind_Data::pProtocolSubtypeElement

8.193 nas_HDRRSSIThresh Struct Reference

Data Fields

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

8.193.1 Detailed Description

This structure contains HDR RSSI threshold related parameters.

Parameters

<i>HDRRSSI-ThreshListLen</i>	<ul style="list-style-type: none"> Length of the HDR RSSI threshold list parameter to follow
<i>pHDRRSSI-ThreshList</i>	<ul style="list-style-type: none"> Array of RSSI thresholds (in units of 0.1 dBm) Maximum of 32 values. Range for RSSI values: -118 to -13 (in dBm).

8.193.2 Field Documentation

8.193.2.1 uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen

8.193.2.2 `int16_t* nas_HDRRSSIthresh::pHDRRSSIthreshList`

8.194 nas_HDRSINRThresh Struct Reference

Data Fields

- `uint8_t HDRSINRThreshListLen`
- `uint8_t* pHDRSINRThreshList`

8.194.1 Detailed Description

This structure contains HDR SINR threshold list.

Parameters

<i>HDRSINRThreshListLen</i>	<ul style="list-style-type: none"> • Length of the HDR SINR threshold list parameter to follow
<i>pHDRSINRThreshList</i>	<ul style="list-style-type: none"> • 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"> – SINR is reported only for HDR – Each SINR threshold value is an unsigned 1 byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.194.2 Field Documentation

8.194.2.1 `uint8_t nas_HDRSINRThresh::HDRSINRThreshListLen`

8.194.2.2 `uint8_t* nas_HDRSINRThresh::pHDRSINRThreshList`

8.195 nas_HDRSINRThreshold Struct Reference

Data Fields

- `uint8_t HDRSINRThreshListLen`
- `uint16_t* pHDRSINRThreshList`

8.195.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

Parameters

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

<i>pHDRSINR- ThreshList</i>	<ul style="list-style-type: none"> • Array of SINR level thresholds (in units of 1) • maximum of 32 values. • Valid levels are 0 to 8 <ul style="list-style-type: none"> – 0x00 - SINR_LEVEL_0 is -9 dB – 0x01 - SINR_LEVEL_1 is -6 dB – 0x02 - SINR_LEVEL_2 is -4.5 dB – 0x03 - SINR_LEVEL_3 is -3 dB – 0x04 - SINR_LEVEL_4 is -2 dB – 0x05 - SINR_LEVEL_5 is +1 dB – 0x06 - SINR_LEVEL_6 is +3 dB – 0x07 - SINR_LEVEL_7 is +6 dB – 0x08 - SINR_LEVEL_8 is +9 dB
---------------------------------	--

8.195.2 Field Documentation

8.195.2.1 `uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen`

8.195.2.2 `uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList`

8.196 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]

8.196.1 Detailed Description

Structure for storing the HDR System Information.

Parameters

<i>sysInfoHDR</i>	<ul style="list-style-type: none"> • See nas_sysInfoCommon for more information.
<i>isSysPrIMatch- Valid</i>	<ul style="list-style-type: none"> • Indicates whether the system PRL match is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available

<i>isSysPrIMatch</i>	<ul style="list-style-type: none"> Indicates whether the system is in a PRL. Only applies to CDMA/HDR. <ul style="list-style-type: none"> 0x00 - System is not in a PRL 0x01 - System is in a PRL 0xFF - Not Available If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL. If the system is in a PRL, roam_status is set to the value based on the standard specification.
<i>hdrPersonality-Valid</i>	<ul style="list-style-type: none"> Indicates whether the HDR personality is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>hdrPersonality</i>	<ul style="list-style-type: none"> HDR personality information. Only applicable for HDR. <ul style="list-style-type: none"> 0x00 - None 0x02 - HRPD 0x03 - eHRPD 0xFF - Not Available
<i>hdrActiveProt-Valid</i>	<ul style="list-style-type: none"> Indicates whether the HDR active protocol revision information is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>hdrActiveProt</i>	<ul style="list-style-type: none"> HDR active protocol revision information . Only applicable for HDR. <ul style="list-style-type: none"> 0x00 - None 0x02 - HDR Rel 0 0x03 - HDR Rel A 0x04 - HDR Rel B 0xFF - Not Available
<i>is856SysIdValid</i>	<ul style="list-style-type: none"> Indicates whether the IS-856 system ID is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>is856SysId[SLQ-S_SYSTEM_ID-SIZE]</i>	<ul style="list-style-type: none"> IS-856 system ID. Only applicable for HDR.

8.196.2 Field Documentation

- 8.196.2.1 `uint8_t nas_HDRSysInfo::hdrActiveProt`
- 8.196.2.2 `uint8_t nas_HDRSysInfo::hdrActiveProtValid`
- 8.196.2.3 `uint8_t nas_HDRSysInfo::hdrPersonality`
- 8.196.2.4 `uint8_t nas_HDRSysInfo::hdrPersonalityValid`
- 8.196.2.5 `uint8_t nas_HDRSysInfo::is856SysId[16]`
- 8.196.2.6 `uint8_t nas_HDRSysInfo::is856SysIdValid`
- 8.196.2.7 `uint8_t nas_HDRSysInfo::isSysPrIMatch`
- 8.196.2.8 `uint8_t nas_HDRSysInfo::isSysPrIMatchValid`
- 8.196.2.9 `nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR`

8.197 nas_homeSIDNID Struct Reference

Data Fields

- `uint8_t numInstances`
- `nas_sidNid SidNid` [255]

8.197.1 Detailed Description

This structure contains Home SID/NID

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • Number of sets of the following elements: <ul style="list-style-type: none"> – sid – nid • If zero(0), then no information follows.
<i>SidNid</i>	<ul style="list-style-type: none"> • See nas_sidNid for more information

8.197.2 Field Documentation

- 8.197.2.1 `uint8_t nas_homeSIDNID::numInstances`
- 8.197.2.2 `nas_sidNid nas_homeSIDNID::SidNid[255]`

8.198 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]

8.198.1 Detailed Description

This structure contains information about the inter-frequency.

Parameters

<i>earfcn</i>	<ul style="list-style-type: none"> • E-UTRA absolute radio frequency channel number of the serving cell. • Range: 0 to 65535.
<i>threshXLow</i>	<ul style="list-style-type: none"> • Cell Srxlev low threshold. • Range: 0 to 31. • 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.
<i>threshXHigh</i>	<ul style="list-style-type: none"> • Cell Srxlev high threshold. • Range: 0 to 31. • 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.
<i>cell_resel_priority</i>	<ul style="list-style-type: none"> • Cell re-selection priority • Range: 0 to 7. • This field is only valid when <code>ue_in_idle</code> is TRUE.
<i>cells_len</i>	<ul style="list-style-type: none"> • Provides the number of set of cell params.
<i>cellInterFreqParams[NAS_MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_cellParams for more information.

8.198.2 Field Documentation

8.198.2.1 uint8_t `nas_infoInterFreq::cell_resel_priority`

8.198.2.2 `nas_cellParams` `nas_infoInterFreq::cellInterFreqParams`[255]

8.198.2.3 uint8_t `nas_infoInterFreq::cells_len`

8.198.2.4 uint16_t `nas_infoInterFreq::earfcn`

8.198.2.5 `uint8_t nas_infoInterFreq::threshXHigh`

8.198.2.6 `uint8_t nas_infoInterFreq::threshXLow`

8.199 nas_IOTresh Struct Reference

Data Fields

- `uint8_t IOThesListLen`
- `int32_t * pIOThesList`

8.199.1 Detailed Description

This structure contains IO threshold list.

Parameters

<i>IOThresListLen</i>	<ul style="list-style-type: none"> • Length of the IO threshold list parameter to follow
<i>pIOThesList</i>	<ul style="list-style-type: none"> • Sequence of thresholds delimiting IO event reporting bands • 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"> – IO is applicable only for HDR – Each IO threshold value is a signed 4 byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.199.2 Field Documentation

8.199.2.1 `uint8_t nas_IOTresh::IOThresListLen`

8.199.2.2 `int32_t* nas_IOTresh::pIOThesList`

8.200 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`

8.200.1 Detailed Description

Contain the LTE Band Preference Extended parameters

Parameters

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

8.200.2 Field Documentation

8.200.2.1 uint64_t nas_lteBandPrefExt::bits_129_192

8.200.2.2 uint64_t nas_lteBandPrefExt::bits_193_256

8.200.2.3 uint64_t nas_lteBandPrefExt::bits_1_64

8.200.2.4 uint64_t nas_lteBandPrefExt::bits_65_128

8.201 nas_LTEBandPrefTlv Struct Reference

Data Fields

- uint64_t [LTEBandPref](#)
- uint8_t [TlvPresent](#)

8.201.1 Detailed Description

Contain the LTE Band Preference for system selection preferences.

Parameters

<i>LTEBandPref</i>	<ul style="list-style-type: none"> • Bit mask representing the LTE band preference • Bit Values <ul style="list-style-type: none"> – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – All other bits are reserved
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.201.2 Field Documentation

8.201.2.1 uint64_t nas_LTEBandPrefTlv::LTEBandPref

8.201.2.2 uint8_t nas_LTEBandPrefTlv::TlvPresent

8.202 nas_LteCiotOpModeTlv Struct Reference

Data Fields

- uint32_t [campedCiotLteOpMode](#)
- uint8_t [TlvPresent](#)

8.202.1 Detailed Description

This structure contains Camped CIOT LTE Operational Mode parameters.

Parameters

<i>campedCiotLteOpMode</i>	<ul style="list-style-type: none"> • Indicates the camped CIOT LTE mode of operation. • Values: <ul style="list-style-type: none"> – NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service – NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband – NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1 – NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present or not

8.202.2 Field Documentation

8.202.2.1 uint32_t nas_LteCiotOpModeTlv::campedCiotLteOpMode

8.202.2.2 uint8_t nas_LteCiotOpModeTlv::TlvPresent

8.203 nas_LteEARFCN Struct Reference

Data Fields

- uint8_t [status](#)
- uint32_t [earfcn0](#)
- uint32_t [earfcn1](#)

8.203.1 Detailed Description

This structure contains LTE EARFCN information.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • 0 - Disable • 1 - Enable
---------------	---

<i>earfcn0</i>	<ul style="list-style-type: none"> Primary DL EARFCN to which the UE is locked
<i>earfcn1</i>	<ul style="list-style-type: none"> Secondary DL EARFCN to which the UE is locked Note : Make earfcn1 value equal to earfcn0 if only one EARFCN is desired.

8.203.2 Field Documentation

8.203.2.1 uint32_t nas_lteEARFCN::earfcn0

8.203.2.2 uint32_t nas_lteEARFCN::earfcn1

8.203.2.3 uint8_t nas_lteEARFCN::status

8.204 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]

8.204.1 Detailed Description

This structure contains information about the LTE GSM Cell.

Parameters

<i>cellReselPriority</i>	<ul style="list-style-type: none"> Priority of this frequency group. Range: 0 to 7. This field is only valid when ue_in_idle is TRUE.
<i>threshGsmHigh</i>	<ul style="list-style-type: none"> Reselection threshold for high priority layers. Range: 0 to 31. This field is only valid when ue_in_idle is TRUE.
<i>threshGsmLow</i>	<ul style="list-style-type: none"> Reselection threshold for low priority layers. Range: 0 to 31. This field is only valid when ue_in_idle is TRUE.

<i>nccPermitted</i>	<ul style="list-style-type: none"> • Bitmask specifying whether a neighbor with a specific network color code is to be reported. • Range: 0 to 255. • 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. • This field is only valid when <code>ue_in_idle</code> is TRUE.
<i>cells_len</i>	<ul style="list-style-type: none"> • Provides the number of set of gsm cells.
<i>GsmCellInfo[NA-S_MAX_DESCRIPTION_LEN-GTH]</i>	<ul style="list-style-type: none"> • See nas_gsmCellInfo for more information.

8.204.2 Field Documentation

8.204.2.1 `uint8_t nas_lteGsmCellInfo::cellReselPriority`

8.204.2.2 `uint8_t nas_lteGsmCellInfo::cells_len`

8.204.2.3 `nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]`

8.204.2.4 `uint8_t nas_lteGsmCellInfo::nccPermitted`

8.204.2.5 `uint8_t nas_lteGsmCellInfo::threshGsmHigh`

8.204.2.6 `uint8_t nas_lteGsmCellInfo::threshGsmLow`

8.205 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`

8.205.1 Detailed Description

Structure for storing the LTE information for the device.

Parameters

<i>band</i>	<ul style="list-style-type: none"> • LTE Band <ul style="list-style-type: none"> – 1 ~ 40 (Band in decimal) – 0xFF - Invalid
-------------	--

<i>bandwidth</i>	<ul style="list-style-type: none">• BandWidth.<ul style="list-style-type: none">– 0x00 - 1.4 MHz– 0x01 - 3 MHz– 0x02 - 5 MHz– 0x03 - 10 MHz– 0x04 - 15 MHz– 0x05 - 20 MHz– 0x06 - Invalid– 0xFF - Unknown
<i>RXChan</i>	<ul style="list-style-type: none">• RX channel number in decimal<ul style="list-style-type: none">– 0xFFFF - Not Available
<i>TXChan</i>	<ul style="list-style-type: none">• TX channel number in decimal<ul style="list-style-type: none">– 0xFFFF - Not Available
<i>emmState</i>	<ul style="list-style-type: none">• EMM State.<ul style="list-style-type: none">– 0x00 - Deregistered– 0x01 - Reg Initiated– 0x02 - Registered– 0x03 - TAU Initiated– 0x04 - SR Initiated– 0x05 - Dereg Initiated– 0x06 - Invalid– 0xFF - Unknown

<i>emmSubState</i>	<ul style="list-style-type: none"> • EMM Sub State. <ul style="list-style-type: none"> – 0xFF - NOT Applicable • When EMM_state is 0x00: <ul style="list-style-type: none"> – 0x00 - No IMSI – 0x01 - PLMN Search – 0x02 - Attach Needed – 0x03 - No Cell – 0x04 - Attaching – 0x05 - Normal Service – 0x06 - Limited Service – 0x07 - Waiting for PDN • When EMM_state is 0x01: <ul style="list-style-type: none"> – 0x00 - Waiting for NW – 0x01 - Waiting for ESM • When EMM_state is 0x02: <ul style="list-style-type: none"> – 0x00 - Normal Service – 0x01 - Update Needed – 0x02 - Attempt Update – 0x03 - No Cell – 0x04 - PLMN Search – 0x05 - Limited Service – 0x06 - MM Update – 0x07 - IMSI Detach – 0x08 - Waiting for ESM
<i>emmConnState</i>	<ul style="list-style-type: none"> • EMM Connected Mode State. <ul style="list-style-type: none"> – 0x00 - RRC Idle – 0x01 - Waiting RRC Cfm – 0x02 - RRC Connected – 0x03 - RRC Releasing – 0xFF - Unknown

8.205.2 Field Documentation

8.205.2.1 uint8_t nas_LTEInfo::band

8.205.2.2 uint8_t nas_LTEInfo::bandwidth

8.205.2.3 uint8_t nas_LTEInfo::emmConnState

8.205.2.4 uint8_t nas_LTEInfo::emmState

8.205.2.5 uint8_t nas_LTEInfo::emmSubState

8.205.2.6 uint16_t nas_LTEInfo::RXChan

8.205.2.7 uint16_t nas_LTEInfo::TXChan

8.206 nas_LTEInfoInterfreq Struct Reference

Data Fields

- uint8_t [ueInIdle](#)
- uint8_t [freqsLen](#)
- [nas_infoInterFreq](#) [InfoInterfreq](#) [255]

8.206.1 Detailed Description

This structure contains information about the LTE Inter-Frequency Network.

Parameters

<i>ueInIdle</i>	<ul style="list-style-type: none"> • TRUE if the UE is in Idle mode, otherwise FALSE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>freqsLen</i>	<ul style="list-style-type: none"> • Provides the number of set of inter frequency information. • If 0(zero), then no information follows it.
<i>InfoInterfreq[NA-S_MAX_DESC-RIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_infoInterFreq for more information.

8.206.2 Field Documentation

8.206.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

8.206.2.2 nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]

8.206.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

8.207 nas_LTEInfoIntrafreq Struct Reference

Data Fields

- uint8_t [ueInIdle](#)
- uint8_t [plmn](#) [3]
- uint16_t [tac](#)
- uint32_t [globalCellId](#)
- uint16_t [earfcn](#)
- uint16_t [servingCellId](#)
- uint8_t [cellReselPriority](#)
- uint8_t [sNonIntraSearch](#)
- uint8_t [threshServingLow](#)
- uint8_t [sIntraSearch](#)
- uint8_t [cellsLen](#)
- [nas_cellParams](#) [CellParams](#) [255]

8.207.1 Detailed Description

This structure contains information about the LTE Intra-Frequency Network.

Parameters

<i>ueInIdle</i>	<ul style="list-style-type: none"> • TRUE if the UE is in Idle mode, otherwise FALSE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>plmn[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • PLMN ID coded as octet 3, 4, and 5.
<i>tac</i>	<ul style="list-style-type: none"> • Tracking area code. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>globalCellId</i>	<ul style="list-style-type: none"> • Global cell ID in the system information block. <ul style="list-style-type: none"> – 0xFFFFFFFF - Not Available
<i>earfcn</i>	<ul style="list-style-type: none"> • E-UTRA absolute radio frequency channel number of the serving cell. • Range: 0 to 65535. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>servingCellId</i>	<ul style="list-style-type: none"> • LTE serving cell ID. • Range: 0 to 503. • This is the cell ID of the serving cell and can be found in the cell list. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>cellReselPriority</i>	<ul style="list-style-type: none"> • Priority for serving frequency. • Range: 0 to 7. • This field is only valid when <i>ue_in_idle</i> is TRUE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>sNonIntraSearch</i>	<ul style="list-style-type: none"> • S non-intra search threshold to control non-intrafrequency searches. • Range: 0 to 31. • This field is only valid when <i>ue_in_idle</i> is TRUE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>threshServing-Low</i>	<ul style="list-style-type: none"> • Serving cell low threshold. • Range: 0 to 31. • This field is only valid when <i>ue_in_idle</i> is TRUE. <ul style="list-style-type: none"> – 0xFF - Not Available

<i>sIntraSearch</i>	<ul style="list-style-type: none"> • S Intra search threshold. • Range: 0 to 31. • The current cell measurement must fall below this threshold to consider intrafrequency for reselection. • This field is only valid when ue_in_idle is TRUE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>cellsLen</i>	<ul style="list-style-type: none"> • Provides the number of set of cell params. • If 0(zero), then no information follows it.
<i>CellParams[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_cellParams for more information.

8.207.2 Field Documentation

8.207.2.1 `nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]`

8.207.2.2 `uint8_t nas_LTEInfoIntrafreq::cellReselPriority`

8.207.2.3 `uint8_t nas_LTEInfoIntrafreq::cellsLen`

8.207.2.4 `uint16_t nas_LTEInfoIntrafreq::earfcn`

8.207.2.5 `uint32_t nas_LTEInfoIntrafreq::globalCellId`

8.207.2.6 `uint8_t nas_LTEInfoIntrafreq::plmn[3]`

8.207.2.7 `uint16_t nas_LTEInfoIntrafreq::servingCellId`

8.207.2.8 `uint8_t nas_LTEInfoIntrafreq::sIntraSearch`

8.207.2.9 `uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch`

8.207.2.10 `uint16_t nas_LTEInfoIntrafreq::tac`

8.207.2.11 `uint8_t nas_LTEInfoIntrafreq::threshServingLow`

8.207.2.12 `uint8_t nas_LTEInfoIntrafreq::ueInIdle`

8.208 nas_LTEInfoNeighboringGSM Struct Reference

Data Fields

- `uint8_t ueInIdle`
- `uint8_t freqsLen`
- `nas_lteGsmCellInfo lteGsmCellInfo [255]`

8.208.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

Parameters

<i>ueInIdle</i>	<ul style="list-style-type: none"> • TRUE if the UE is in Idle mode, otherwise FALSE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>freqsLen</i>	<ul style="list-style-type: none"> • Provides the number of set of LTE GSM cell information. • If 0(zero), then no information follows it.
<i>LteGsmCellInfo[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_LteGsmCellInfo for more information.

8.208.2 Field Documentation

8.208.2.1 `uint8_t nas_LTEInfoNeighboringGSM::freqsLen`

8.208.2.2 `nas_LteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]`

8.208.2.3 `uint8_t nas_LTEInfoNeighboringGSM::ueInIdle`

8.209 nas_LTEInfoNeighboringWCDMA Struct Reference

Data Fields

- `uint8_t ueInIdle`
- `uint8_t freqsLen`
- `nas_LteWcdmaCellInfo LTEWCDMACellInfo [255]`

8.209.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

Parameters

<i>ueInIdle</i>	<ul style="list-style-type: none"> • TRUE if the UE is in Idle mode, otherwise FALSE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>freqsLen</i>	<ul style="list-style-type: none"> • Provides the number of set of LTE WCDMA cell information. • If 0(zero), then no information follows it.
<i>LTEWCDMA-CellInfo[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_LteWcdmaCellInfo for more information.

8.209.2 Field Documentation

8.209.2.1 uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen

8.209.2.2 nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.209.2.3 uint8_t nas_LTEInfoNeighboringWCDMA::uelIdle

8.210 nas_LteM1BandPrefTlv Struct Reference

Data Fields

- uint64_t [lteM1BandPref](#)
- uint8_t [TlvPresent](#)

8.210.1 Detailed Description

Contain the LTE M1 band Preference.

Parameters

<i>lteM1BandPref</i>	<ul style="list-style-type: none"> • Bitmask representing the LTE M1 band preference • Bit Values <ul style="list-style-type: none"> – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – Bit 61 - E-UTRA Operating Band 126 – Bit 62 - E-UTRA Operating Band 127
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.210.2 Field Documentation

8.210.2.1 uint64_t nas_LteM1BandPrefTlv::lteM1BandPref

8.210.2.2 uint8_t nas_LteM1BandPrefTlv::TlvPresent

8.211 nas_LteNb1BandPrefTlv Struct Reference

Data Fields

- uint64_t [lteNb1BandPref](#)
- uint8_t [TlvPresent](#)

8.211.1 Detailed Description

Contain the LTE NB1 band Preference.

Parameters

<i>lteNb1BandPref</i>	<ul style="list-style-type: none"> • Bitmask representing the LTE NB1 band preference. • Bit Values <ul style="list-style-type: none"> – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – Bit 61 - E-UTRA Operating Band 126 – Bit 62 - E-UTRA Operating Band 127
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.211.2 Field Documentation

8.211.2.1 uint64_t nas_LteNb1BandPrefTlv::lteNb1BandPref

8.211.2.2 uint8_t nas_LteNb1BandPrefTlv::TlvPresent

8.212 nas_LTEOperationMode Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t * [pLTEOperationMode](#)

8.212.1 Detailed Description

This structure contains the [nas_LTEOperationMode](#) response parameters.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• Tlv Present
<i>pLTEOperation-Mode</i>	<ul style="list-style-type: none">• LTE Operational Mode

8.212.2 Field Documentation

8.212.2.1 uint32_t* nas_LTEOperationMode::pLTEOperationMode

8.212.2.2 uint8_t nas_LTEOperationMode::TlvPresent

8.213 nas_LteOpMode Struct Reference

Data Fields

- uint32_t [lteOpMode](#)
- uint8_t [TlvPresent](#)

8.213.1 Detailed Description

This structure contains LTE Operational Mode

Parameters

<i>lteOpMode</i>	<ul style="list-style-type: none">• LTE Operational Mode• CIOT LTE mode on which reject indication is received.• Values<ul style="list-style-type: none">– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1
------------------	--

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

8.213.2 Field Documentation

8.213.2.1 uint32_t nas_LteOpMode::lteOpMode

8.213.2.2 uint8_t nas_LteOpMode::TlvPresent

8.214 nas_LteOpModeTlv Struct Reference

Data Fields

- uint8_t [lteOpModeLen](#)
- uint16_t [MCC](#) [255]
- uint16_t [MNC](#) [255]
- uint32_t [lteOpMode](#) [255]
- uint8_t [TlvPresent](#)

8.214.1 Detailed Description

This structure contains the LTE Operational Mode response parameters.

Parameters

<i>lteOpModeLen</i>	<ul style="list-style-type: none"> • Number of sets of the following elements <ul style="list-style-type: none"> – MCC – MNC – lteOpMode
<i>MCC</i>	<ul style="list-style-type: none"> • Mobile Country Code • A 16-bit integer representation of MCC • Range: 0 to 999
<i>MNC</i>	<ul style="list-style-type: none"> • Mobile Network Code • A 16-bit integer representation of MNC • Range: 0 to 999
<i>lteOpMode</i>	<ul style="list-style-type: none"> • Indicates the LTE mode of operation. • Values: <ul style="list-style-type: none"> – NAS_CIOT_SYS_MODE_NO_SRV (0x00) - No service – NAS_CIOT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband – NAS_CIOT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1 – NAS_CIOT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1

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

8.214.2 Field Documentation

8.214.2.1 uint32_t nas_IteOpModeTlv::lteOpMode[255]

8.214.2.2 uint8_t nas_IteOpModeTlv::lteOpModeLen

8.214.2.3 uint16_t nas_IteOpModeTlv::MCC[255]

8.214.2.4 uint16_t nas_IteOpModeTlv::MNC[255]

8.214.2.5 uint8_t nas_IteOpModeTlv::TlvPresent

8.215 nas_ItePCI Struct Reference

Data Fields

- uint8_t [status](#)
- uint32_t [earfcn](#)
- uint32_t [pci](#)

8.215.1 Detailed Description

This structure contains LTE PCI information.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • 0 - Disable • 1 - Enable
<i>earfcn</i>	<ul style="list-style-type: none"> • UARFCN to which UE is locked
<i>pci</i>	<ul style="list-style-type: none"> • PCI to which the UE is locked

8.215.2 Field Documentation

8.215.2.1 uint32_t nas_ItePCI::earfcn

8.215.2.2 uint32_t nas_ItePCI::pci

8.215.2.3 uint8_t nas_ItePCI::status

8.216 nas_IteRsrpinformation Struct Reference

Data Fields

- int16_t [rsrplevel](#)

8.216.1 Detailed Description

This structure contains the LTE RSRP Information

Parameters

<i>rsrplevel</i>	<ul style="list-style-type: none"> • LTE RSRP in dBm as a mesaured by L1. Range: -44 to -140(-44 means -44dBm, -140 means -140dBm).
------------------	--

8.216.2 Field Documentation

8.216.2.1 int16_t nas_lteRsrpinformation::rsrplevel

8.217 nas_LTERSRPThresh Struct Reference

Data Fields

- uint8_t [LTERSRPThreshListLen](#)
- int16_t * [pLTERSRPThreshList](#)

8.217.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

Parameters

<i>LTERSRP- ThreshListLen</i>	<ul style="list-style-type: none"> • Length of the LTE RSRP threshold list parameter to follow
<i>pLTERSRP- ThreshList</i>	<ul style="list-style-type: none"> • Array of RSRP thresholds (in units of 0.1 dBm) • Maximum of 32 values • Range for RSRP values: -140 to -44 (in dBm).

8.217.2 Field Documentation

8.217.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

8.217.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

8.218 nas_LTERSRQThresh Struct Reference

Data Fields

- uint8_t [LTERSRQThreshListLen](#)

- int16_t * [pLTERSRQThreshList](#)

8.218.1 Detailed Description

This structure contains LTE RSRQ threshold related parameters.

Parameters

<i>LTERSRQ-ThreshListLen</i>	<ul style="list-style-type: none"> • Length of the LTE RSRQ threshold list parameter to follow
<i>pLTERSRQ-ThreshList</i>	<ul style="list-style-type: none"> • Array of RSRQ thresholds (in units of 0.1 dBm) • Maximum of 32 values. • Range for RSRQ values: -20 to -3 (in dBm)

8.218.2 Field Documentation

8.218.2.1 uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen

8.218.2.2 int16_t* nas_LTERSRQThresh::pLTERSRQThreshList

8.219 nas_LTERSSIThresh Struct Reference

Data Fields

- uint8_t [LTERSSIThreshListLen](#)
- int16_t * [pLTERSSIThreshList](#)

8.219.1 Detailed Description

This structure contains LTE RSSI threshold related parameters.

Parameters

<i>LTERSSI-ThreshListLen</i>	<ul style="list-style-type: none"> • Length of the LTE RSSI threshold list parameter to follow
<i>pLTERSSI-ThreshList</i>	<ul style="list-style-type: none"> • Array of RSSI thresholds (in units of 0.1 dBm) • Maximum of 32 values. • Range for RSSI values: -120 to 0 (in dBm)

8.219.2 Field Documentation

8.219.2.1 uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen

8.219.2.2 int16_t* nas_LTERSSIThresh::pLTERSSIThreshList

8.220 nas_LTESigRptCfg Struct Reference

Data Fields

- uint8_t [rptRate](#)
- uint8_t [avgPeriod](#)

8.220.1 Detailed Description

This structure contains LTE signal report config.

Parameters

<i>rptRate</i>	<ul style="list-style-type: none"> • Rate on how often the LTE signal must be checked for reporting • Values: <ul style="list-style-type: none"> – 0 - Report using the default configuration – 1 - Report every 1 sec – 2 - Report every 2 sec – 3 - Report every 3 sec – 4 - Report every 4 sec – 5 - Report every 5 sec
<i>avgPeriod</i>	<ul style="list-style-type: none"> • Averaging period to be used for the LTE signal • Values: <ul style="list-style-type: none"> – 0 - Average using the default configuration – 1 - Average over 1 sec – 2 - Average over 2 sec – 3 - Average over 3 sec – 4 - Average over 4 sec – 5 - Average over 5 sec – 6 - Average over 6 sec – 7 - Average over 7 sec – 8 - Average over 8 sec – 9 - Average over 9 sec – 10 - Average over 10 sec

8.220.2 Field Documentation

8.220.2.1 uint8_t nas_LTESigRptCfg::avgPeriod

8.220.2.2 uint8_t nas_LTESigRptCfg::rptRate

8.221 nas_LTESigRptConfig Struct Reference

Data Fields

- uint8_t [rptRate](#)
- uint8_t [avgPeriod](#)

8.221.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

Parameters

<i>rptRate</i>	<ul style="list-style-type: none"> • Rate on how often the LTE signal must be checked for reporting Values • 0 - Report using the default configuration • 1 - Report every 1 sec • 2 - Report every 2 sec • 3 - Report every 3 sec • 4 - Report every 4 sec • 5 - Report every 5 sec
<i>avgPeriod</i>	<ul style="list-style-type: none"> • Averaging period to be used for the LTE signal. • Values <ul style="list-style-type: none"> – 0 - Average using the default configuration – 1 - Average over 1 sec – 2 - Average over 2 sec – 3 - Average over 3 sec – 4 - Average over 4 sec – 5 - Average over 5 sec – 6 - Average over 6 sec – 7 - Average over 7 sec – 8 - Average over 8 sec – 9 - Average over 9 sec – 10 - Average over 10 sec

8.221.2 Field Documentation

8.221.2.1 uint8_t nas_LTESigRptConfig::avgPeriod

8.221.2.2 uint8_t nas_LTESigRptConfig::rptRate

8.222 nas_IteSnrinformation Struct Reference

Data Fields

- int16_t [snrlevel](#)

8.222.1 Detailed Description

This structure contains the LTE SNR Information

Parameters

<i>snrlevel</i>	<ul style="list-style-type: none"> • 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.
-----------------	---

8.222.2 Field Documentation

8.222.2.1 `int16_t nas_lteSnrinformation::snrlevel`

8.223 nas_LTESNRThresh Struct Reference

Data Fields

- `uint8_t` [LTESNRThresListLen](#)
- `int16_t *` [pLTESNRThresList](#)

8.223.1 Detailed Description

This structure contains LTE SNR threshold list.

Parameters

<i>LTESNRThres-ListLen</i>	<ul style="list-style-type: none"> • Length of the LTE SNR threshold list parameter to follow
<i>pLTESNRThres-List</i>	<ul style="list-style-type: none"> • Sequence of thresholds delimiting SNR event reporting bands • 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"> – For LTE, each SNR threshold value is a signed 2 Byte value – Maximum number of threshold values is 16 – At least one value must be specified – 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

8.223.2 Field Documentation

8.223.2.1 `uint8_t nas_LTESNRThresh::LTESNRThresListLen`

8.223.2.2 `int16_t*` `nas_LTESNRThresh::pLTESNRThresList`

8.224 nas_LTESNRThreshold Struct Reference

Data Fields

- `uint8_t` [LTESNRThreshListLen](#)
- `int16_t *` [pLTESNRThreshList](#)

8.224.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

Parameters

<i>LTESNRThresh-ListLen</i>	<ul style="list-style-type: none"> • Length of the LTE SNR threshold list parameter to follow
-----------------------------	--

<i>pLTESNR- ThreshList</i>	<ul style="list-style-type: none"> • Array of SNR thresholds (in units of 0.1 dB) • Maximum of 32 values • Range for SNR values: -20 to 30 (in dB).
--------------------------------	--

8.224.2 Field Documentation

8.224.2.1 `uint8_t nas_LTESNRThreshold::LTESNRThreshListLen`

8.224.2.2 `int16_t* nas_LTESNRThreshold::pLTESNRThreshList`

8.225 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`

8.225.1 Detailed Description

Structure for storing the LTE System Information.

Parameters

<i>sysInfoLTE</i>	<ul style="list-style-type: none"> • See nas_sysInfoCommon for more information.
<i>lacValid</i>	<ul style="list-style-type: none"> • Indicates whether the location area code is valid.. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>lac</i>	<ul style="list-style-type: none"> • Location area code. • Only applies to 3GPP. <ul style="list-style-type: none"> – 0xFFFF - Not Available

<i>cellIdValid</i>	<ul style="list-style-type: none"> Indicates whether the cell ID is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>cellId</i>	<ul style="list-style-type: none"> Cell ID. <ul style="list-style-type: none"> 0xFFFFFFFF - Not Available
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service 0x01 - Circuit-switched only 0x02 - Packet-switched only 0x03 - Circuit-switched and packet-switched 0x04 - Camped 0xFF - Not Available
<i>rejCause</i>	<ul style="list-style-type: none"> 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"> 0xFF - Not Available
<i>networkIdValid</i>	<ul style="list-style-type: none"> Indicates whether the network ID is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>MCC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> Mobile Country Code. MCC digits in ASCII characters
<i>MNC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> Mobile Network Code. MNC digits in ASCII characters An unused byte is set to 0xFF. 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.

<i>tacValid</i>	<ul style="list-style-type: none"> Indicates whether tracking area code is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>tac</i>	<ul style="list-style-type: none"> Tracking area code. Only applicable for LTE. <ul style="list-style-type: none"> 0xFFFF - Not Available

8.225.2 Field Documentation

- 8.225.2.1 `uint32_t nas_LTESysInfo::cellId`
- 8.225.2.2 `uint8_t nas_LTESysInfo::cellIdValid`
- 8.225.2.3 `uint16_t nas_LTESysInfo::lac`
- 8.225.2.4 `uint8_t nas_LTESysInfo::lacValid`
- 8.225.2.5 `uint8_t nas_LTESysInfo::MCC[3]`
- 8.225.2.6 `uint8_t nas_LTESysInfo::MNC[3]`
- 8.225.2.7 `uint8_t nas_LTESysInfo::networkIdValid`
- 8.225.2.8 `uint8_t nas_LTESysInfo::regRejectInfoValid`
- 8.225.2.9 `uint8_t nas_LTESysInfo::rejCause`
- 8.225.2.10 `uint8_t nas_LTESysInfo::rejectSrvDomain`
- 8.225.2.11 `nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE`
- 8.225.2.12 `uint16_t nas_LTESysInfo::tac`
- 8.225.2.13 `uint8_t nas_LTESysInfo::tacValid`

8.226 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]

8.226.1 Detailed Description

This structure contains information about the LTE WCDMA Cell.

Parameters

<i>uarfcn</i>	<ul style="list-style-type: none"> WCDMA layer frequency. Range: 0 to 16383.
<i>cellReselPriority</i>	<ul style="list-style-type: none"> Cell re-selection priority. Range: 0 to 7. This field is only valid when <i>ue_in_idle</i> is TRUE.
<i>threshXhigh</i>	<ul style="list-style-type: none"> Re-selection low threshold. Range: 0 to 31. This field is only valid when <i>ue_in_idle</i> is TRUE.
<i>threshXlow</i>	<ul style="list-style-type: none"> Re-selection high threshold. Range: 0 to 31. This field is only valid when <i>ue_in_idle</i> is TRUE.
<i>cellsLen</i>	<ul style="list-style-type: none"> Provides the number of set of WCDMA cells.
<i>WCDMACell-Info[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> See nas_wcdmaCellInfo for more information.

8.226.2 Field Documentation

8.226.2.1 `uint8_t nas_lteWcdmaCellInfo::cellReselPriority`

8.226.2.2 `uint8_t nas_lteWcdmaCellInfo::cellsLen`

8.226.2.3 `uint16_t nas_lteWcdmaCellInfo::threshXhigh`

8.226.2.4 `uint16_t nas_lteWcdmaCellInfo::threshXlow`

8.226.2.5 `uint16_t nas_lteWcdmaCellInfo::uarfcn`

8.226.2.6 `nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]`

8.227 nas_minBasedIMSI Struct Reference

Data Fields

- `uint8_t mccM` [3]
- `uint16_t imsiM1112`
- `uint8_t imsiMS1` [7]

- uint8_t [imsiMS2](#) [3]

8.227.1 Detailed Description

This structure contains MIN-based IMSI.

Parameters

<i>mccM</i>	<ul style="list-style-type: none"> • ASCII character representation of MCC_M
<i>imsiM1112</i>	<ul style="list-style-type: none"> • ASCII character representation of IMSI_M_11_12 value <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>imsiMS1</i>	<ul style="list-style-type: none"> • ASCII character representation of IMSI_M_S1 value
<i>imsiMS2</i>	<ul style="list-style-type: none"> • ASCII character representation of IMSI_M_S2 value

8.227.2 Field Documentation

8.227.2.1 uint16_t nas_minBasedIMSI::imsiM1112

8.227.2.2 uint8_t nas_minBasedIMSI::imsiMS1[7]

8.227.2.3 uint8_t nas_minBasedIMSI::imsiMS2[3]

8.227.2.4 uint8_t nas_minBasedIMSI::mccM[3]

8.228 nas_MNRInfo Struct Reference

Data Fields

- uint16_t [mcc](#)
- uint16_t [mnc](#)
- uint32_t [rat](#)

8.228.1 Detailed Description

Structure contains Manual Network Register Information parameters

Parameters

<i>mcc</i>	<ul style="list-style-type: none"> • A 16-bit integer representation of Mobile Country Code. Range - 0 to 999.
<i>mnc</i>	<ul style="list-style-type: none"> • A 16-bit integer representation of Mobile Network Code. Range - 0 to 999.

<i>rat</i>	<ul style="list-style-type: none"> Radio access technology for which to register. <ul style="list-style-type: none"> 0x04 - RADIO_IF_GSM 0x05 - RADIO_IF_UMTS 0x08 - RADIO_IF_LTE
------------	--

8.228.2 Field Documentation

8.228.2.1 uint16_t nas_MNRInfo::mcc

8.228.2.2 uint16_t nas_MNRInfo::mnc

8.228.2.3 uint32_t nas_MNRInfo::rat

8.229 nas_ModePrefTlv Struct Reference

Data Fields

- uint16_t [ModePref](#)
- uint8_t [TlvPresent](#)

8.229.1 Detailed Description

Contain the radio technology mode preference

Parameters

<i>ModePref</i>	<ul style="list-style-type: none"> Bit Mask indicating the radio technology mode preference Bit values: <ul style="list-style-type: none"> Bit 0 - cdma2000 1x Bit 1 - cdma2000 HRPD(1xEV-DO) Bit 2 - GSM Bit 3 - UMTS Bit 4 - LTE Bit 5 - TD-SCDMA Bit 6 - NR5G
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present

8.229.2 Field Documentation

8.229.2.1 uint16_t nas_ModePrefTlv::ModePref

8.229.2.2 uint8_t nas_ModePrefTlv::TlvPresent

8.230 nas_namName Struct Reference

Data Fields

- uint8_t [namNameLen](#)
- uint8_t [namName](#) [12]

8.230.1 Detailed Description

This structure contains NAM Name.

Parameters

<i>namNameLen</i>	<ul style="list-style-type: none">• Number of sets of the following elements:<ul style="list-style-type: none">– <code>nam_name</code>• If zero(0), then no information follows.
<i>namName</i>	<ul style="list-style-type: none">• Name information in ASCII. The maximum length of <code>nam_name</code> is 12.

8.230.2 Field Documentation

8.230.2.1 uint8_t nas_namName::namName[12]

8.230.2.2 uint8_t nas_namName::namNameLen

8.231 nas_netSelectionPref Struct Reference

Data Fields

- uint8_t [netReg](#)
- uint16_t [mcc](#)
- uint16_t [mnc](#)

8.231.1 Detailed Description

Contain the network selection preference.

Parameters

<i>netReg</i>	<ul style="list-style-type: none">• specifies one of the following actions:<ul style="list-style-type: none">– 0x00 - Automatic registration<ul style="list-style-type: none">* Device registers according to its provisioning; mcc and mnc fields are ignored– 0x01 - Manual Registration<ul style="list-style-type: none">* Device registers to specified network; mcc and mnc must contain valid values
<i>mcc</i>	<ul style="list-style-type: none">• MCC value. Range 0 to 999
<i>mnc</i>	<ul style="list-style-type: none">• MNC value. Range 0 to 999

8.231.2 Field Documentation

8.231.2.1 uint16_t nas_netSelectionPref::mcc

8.231.2.2 uint16_t nas_netSelectionPref::mnc

8.231.2.3 uint8_t nas_netSelectionPref::netReg

8.232 nas_NetSelPrefTlv Struct Reference

Data Fields

- uint8_t [NetSelPref](#)
- uint8_t [TlvPresent](#)

8.232.1 Detailed Description

Contain the Network selection Preference for system selection preferences.

Parameters

<i>NetSelPref</i>	<ul style="list-style-type: none"> • Parameter indicating network selection preference • Values: <ul style="list-style-type: none"> – 0x00 - Automatic network selection – 0x01 - Manual network selection
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.232.2 Field Documentation

8.232.2.1 uint8_t nas_NetSelPrefTlv::NetSelPref

8.232.2.2 uint8_t nas_NetSelPrefTlv::TlvPresent

8.233 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](#)

8.233.1 Detailed Description

This structure contains network stat 1x.

Parameters

<i>State</i>	<ul style="list-style-type: none"> • CDMA current access state <ul style="list-style-type: none"> – 0x01 - Searching – 0x02 - Idle – 0x03 - Traffic – Others - NA
<i>SO</i>	<ul style="list-style-type: none"> • CDMA service option <ul style="list-style-type: none"> – 0xFFFF - Not in a call – 0x0001 - Basic Variable Rate Voice Service(8kbps) – 0x0002 - Mobile Station Loopback(8kbps) – 0x0003 - Enhanced Variable Rate Codec(EVRC) Voice Service(8kbps) – 0x0006 - Short message Services(Rate Set 1) – 0x0009 - Mobile Station Loopback(13kbps) – 0x000E - Short Message Service (Rate Set 2) – 0x0011 - High Rate Voice Service(13kbps) – 0x0020 - Test Data Service Option(TDSO) – 0x0021 - cdma2000 High Speed Packet Data Service, Internet or ISO Protocol Stack – 0x0044 - EVRC-B Voice Service(8 kbps) – 0x0046 - EVRC-WB Voice Service(8 kbps) – 0x0049 - Voice Echo mode supports smart blanking(EVRC-NW) – 0x004B - Enhanced loopback – 0x8000 - Proprietary Service Option (Qualcomm Inc.)
<i>RX_PWR</i>	<ul style="list-style-type: none"> • RX Pwr(dBm) <ul style="list-style-type: none"> – 0xABCD00EF - -ABCD.EF dBm – ABCD00EF should be transferred to decimal while displaying – Example: 0x12340056 - -4660.86dBm 0x1234 = 4660, 0x0056 = 86 – 0xFFFFFFFF - NA
<i>RX_EC_IO</i>	<ul style="list-style-type: none"> • RX EC/IO(dB) <ul style="list-style-type: none"> – 0xABCD - -AB.CD dB – ABCD should be transferred to decimal while displaying – Example: 0x1234 - -18.52dB 0x12 = 18, 0x34 = 52 – 0xFFFF - NA
<i>TX_PWR</i>	<ul style="list-style-type: none"> • TX PWR(dBm) <ul style="list-style-type: none"> – 0xFFFFFFFF - NA – Others - display actual value in decimal – Example: 0x1234 - -4660dBm 0x1234 = 4660

<i>ActSetCnt</i>	<ul style="list-style-type: none"> Count of active pilot PN elements As input specifies number of sets of parameter pActPilotElements for which memory has been assigned As output specifies the actual number of sets of parameter pActPilotElements returned by device
<i>pActPilotPN-Elements</i>	<ul style="list-style-type: none"> See nas_ActPilotPNElement for more information
<i>NeighborSetCnt</i>	<ul style="list-style-type: none"> Count of neighbor pilot PN elements As input specifies number of sets of parameter pNeighborSetPilotPN for which memory has been assigned As output specifies the actual number of sets of parameter pNeighborSetPilotPN returned by device
<i>pNeighborSet-PilotPN</i>	<ul style="list-style-type: none"> Neighbor pilot PN

8.233.2 Field Documentation

8.233.2.1 `uint8_t nas_NetworkStat1x::ActSetCnt`

8.233.2.2 `uint8_t nas_NetworkStat1x::NeighborSetCnt`

8.233.2.3 `nas_ActPilotPNElement* nas_NetworkStat1x::pActPilotPNElements`

8.233.2.4 `uint16_t* nas_NetworkStat1x::pNeighborSetPilotPN`

8.233.2.5 `uint16_t nas_NetworkStat1x::RX_EC_IO`

8.233.2.6 `uint32_t nas_NetworkStat1x::RX_PWR`

8.233.2.7 `uint16_t nas_NetworkStat1x::SO`

8.233.2.8 `uint8_t nas_NetworkStat1x::State`

8.233.2.9 `uint32_t nas_NetworkStat1x::TX_PWR`

8.234 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](#)

8.234.1 Detailed Description

This structure contains network stat EVDO.

Parameters

<i>State</i>	<ul style="list-style-type: none"> • EVDO network access state <ul style="list-style-type: none"> – 0x00 - Sleep – 0x01 - Searching – 0x02 - Idle – 0x03 - Active – 0xFF - NA
<i>MACIndex</i>	<ul style="list-style-type: none"> • HDR Mac index <ul style="list-style-type: none"> – 0xFF - NA – Others - Display the actual value in decimal – Example: 0x12 - 18 0x12 = 18
<i>SectorIDLen</i>	(IN/OUT) <ul style="list-style-type: none"> • Sector ID length • As input specifies length of parameter pSectorID for which memory has been assigned • As output specifies the actual length of parameter pSectorID returned by device
<i>pSectorID</i>	<ul style="list-style-type: none"> • Sector ID
<i>RX_PWR</i>	<ul style="list-style-type: none"> • TX PWR(dBm) <ul style="list-style-type: none"> – 0xABCD - -ABCD dBm – ABCD should be transferred to decimal while displaying – Example: 0x1234 - -4660dBm 0x1234 = 4660 – 0xFFFF - NA
<i>PER</i>	<ul style="list-style-type: none"> • HDR Packet Error Rate <ul style="list-style-type: none"> – 0xFFFF - Unknown – Others - display the actual value in decimal – Example: 0x1234 - -4660dBm 0x1234 = 4660
<i>PilotEnergy</i>	<ul style="list-style-type: none"> • Pilt Energy (dB) <ul style="list-style-type: none"> – 0xFFFF - NA – 0xABCD should be transferred to decimal while displaying – Example: 0x1234 - -4660dBm 0x1234 = 4660
<i>SNR</i>	<ul style="list-style-type: none"> • Signal to Noise ratio (dB)

8.234.2 Field Documentation

8.234.2.1 `uint8_t nas_NetworkStatEVDO::MACIndex`

8.234.2.2 `uint16_t nas_NetworkStatEVDO::PER`

8.234.2.3 `uint16_t nas_NetworkStatEVDO::PilotEnergy`

8.234.2.4 `uint16_t* nas_NetworkStatEVDO::pSectorID`

8.234.2.5 `uint16_t nas_NetworkStatEVDO::RX_PWR`

8.234.2.6 `uint8_t nas_NetworkStatEVDO::SectorIDLen`

8.234.2.7 `uint8_t nas_NetworkStatEVDO::SNR`

8.234.2.8 `uint8_t nas_NetworkStatEVDO::State`

8.235 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`

8.235.1 Detailed Description

This structure contains information about the Network Measurement Report (NMR) Cell Information.

Parameters

<i>nmrCellID</i>	<ul style="list-style-type: none"> • Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
<i>nmrPlmn</i> [NAS_ - PLMN_LEN- H]	<ul style="list-style-type: none"> • MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when <i>nmrCellID</i> is not present.
<i>nmrLac</i>	<ul style="list-style-type: none"> • Location area code. • This field is ignored when <i>nmrCellID</i> is not present. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>nmrArfcn</i>	<ul style="list-style-type: none"> • Absolute RF channel number. <ul style="list-style-type: none"> – 0xFFFF - Not Available

<i>nmrBsic</i>	<ul style="list-style-type: none"> Base station identity code. <ul style="list-style-type: none"> 0xFF - Not Available
<i>nmrRxLev</i>	<ul style="list-style-type: none"> Cell Rx measurement. Values range between 0 and 63. Mapped to a measured signal level: <ul style="list-style-type: none"> Rxlev 0 is a signal strength less than -110 dBm Rxlev 1 is -110 dBm to -109 dBm Rxlev 2 is -109 dBm to -108 dBm ... Rxlev 62 is -49 dBm to -48 dBm Rxlev 63 is greater than -48 dBm 0xFFFF - Not Available

8.235.2 Field Documentation

8.235.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

8.235.2.2 uint8_t nas_nmrCellInfo::nmrBsic

8.235.2.3 uint32_t nas_nmrCellInfo::nmrCellID

8.235.2.4 uint16_t nas_nmrCellInfo::nmrLac

8.235.2.5 uint8_t nas_nmrCellInfo::nmrPlmn[3]

8.235.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

8.236 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](#)

8.236.1 Detailed Description

Contain the NR5G Band Preference.

Parameters

<i>bits_1_64</i>	<ul style="list-style-type: none"> Bits 1 to 64 of the 256-bit NR5G Operating Band bitmask.
<i>bits_65_128</i>	<ul style="list-style-type: none"> Bits 65 to 128 of the 256-bit NR5G Operating Band bitmask.

<i>bits_129_192</i>	<ul style="list-style-type: none"> • Bits 129 to 192 of the 256-bit NR5G Operating Band bitmask.
<i>bits_193_256</i>	<ul style="list-style-type: none"> • Bits 193 to 256 of the 256-bit NR5G Operating Band bitmask.

8.236.2 Field Documentation

8.236.2.1 `uint64_t nas_nr5gBandPref::bits_129_192`

8.236.2.2 `uint64_t nas_nr5gBandPref::bits_193_256`

8.236.2.3 `uint64_t nas_nr5gBandPref::bits_1_64`

8.236.2.4 `uint64_t nas_nr5gBandPref::bits_65_128`

8.237 nas_NR5GCellStatusTlv Struct Reference

Data Fields

- `uint32_t` [nr5gCellStatus](#)
- `uint8_t` [TlvPresent](#)

8.237.1 Detailed Description

Structure for storing the NR5G Cell Access Status Info parameters

Parameters

<i>nr5gCellStatus</i>	<ul style="list-style-type: none"> • Cell access status for NR5G calls. <ul style="list-style-type: none"> – Values: <ul style="list-style-type: none"> * <code>NAS_CELL_ACCESS_NORMAL_ONLY</code> (0x00) - Cell access is allowed for normal calls only * <code>NAS_CELL_ACCESS_EMERGENCY_ONLY</code> (0x01) - Cell access is allowed for emergency calls only * <code>NAS_CELL_ACCESS_NO_CALLS</code> (0x02) - Cell access is not allowed for any call type * <code>NAS_CELL_ACCESS_ALL_CALLS</code> (0x03) - Cell access is allowed for all call types * <code>NAS_CELL_ACCESS_UNKNOWN</code> (-1) - Cell access type is unknown
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present or not

8.237.2 Field Documentation

8.237.2.1 `uint32_t nas_NR5GCellStatusTlv::nr5gCellStatus`

8.237.2.2 `uint8_t nas_NR5GCellStatusTlv::TlvPresent`

8.238 nas_NR5GSerStatTlv Struct Reference

Data Fields

- uint8_t [srvStatus](#)
- uint8_t [trueSrvStatus](#)
- uint8_t [isPrefDataPath](#)
- uint8_t [TlvPresent](#)

8.238.1 Detailed Description

This structure contains NR5G Service Status Info parameters.

Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"> • Service status of the system. • Values: <ul style="list-style-type: none"> – 0x00 - SYS_SRV_STATUS_NO_SRV - No service – 0x01 - SYS_SRV_STATUS_LIMITED - Limited service – 0x02 - SYS_SRV_STATUS_SRV - Service – 0x03 - SYS_SRV_STATUS_LIMITED_REGIONAL - Limited regional service – 0x04 - SYS_SRV_STATUS_PWR_SAVE - Power save
<i>trueSrvStatus</i>	<ul style="list-style-type: none"> • True service status of the system (not applicable to CDMA/HDR). • Values: <ul style="list-style-type: none"> – 0x00 - SYS_SRV_STATUS_NO_SRV - No service – 0x01 - SYS_SRV_STATUS_LIMITED - Limited service – 0x02 - SYS_SRV_STATUS_SRV - Service – 0x03 - SYS_SRV_STATUS_LIMITED_REGIONAL - Limited regional service – 0x04 - SYS_SRV_STATUS_PWR_SAVE - Power save
<i>isPrefDataPath</i>	<ul style="list-style-type: none"> • Whether the RAT is the preferred data path: • Values <ul style="list-style-type: none"> – 0x00 - Not preferred – 0x01 - Preferred
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present or not

8.238.2 Field Documentation

8.238.2.1 uint8_t nas_NR5GSerStatTlv::isPrefDataPath

8.238.2.2 uint8_t nas_NR5GSerStatTlv::srvStatus

8.238.2.3 uint8_t nas_NR5GSerStatTlv::TlvPresent

8.238.2.4 uint8_t nas_NR5GSerStatTlv::trueSrvStatus

8.239 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 [nwIdValid](#)
- int8_t [MCC](#) [3]
- int8_t [MNC](#) [3]
- uint8_t [tacValid](#)
- uint16_t [tac](#)
- uint8_t [TlvPresent](#)

8.239.1 Detailed Description

This structure contains NR5G System Info parameters.

Parameters

<i>srvDomainValid</i>	<ul style="list-style-type: none"> • Indicates whether the service domain is valid.
<i>srvDomain</i>	<ul style="list-style-type: none"> • Service domain registered on the system. <ul style="list-style-type: none"> – Values: <ul style="list-style-type: none"> * 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service * 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only * 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only * 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched * 0x04 - SYS_SRV_DOMAIN_CAMPED -Camped
<i>srvCapValid</i>	<ul style="list-style-type: none"> • Indicates whether the service capability is valid.

<i>srvCapability</i>	<ul style="list-style-type: none"> Current system service capability. <ul style="list-style-type: none"> Values: <ul style="list-style-type: none"> * 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service * 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only * 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only * 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched * 0x04 - SYS_SRV_DOMAIN_CAMPED - Camped
<i>roamStatusValid</i>	<ul style="list-style-type: none"> Indicates whether the roaming status is valid.
<i>roamStatus</i>	<ul style="list-style-type: none"> Current roaming status. <ul style="list-style-type: none"> Values: <ul style="list-style-type: none"> * 0x00 - SYS_ROAM_STATUS_OFF - Off * 0x01 - SYS_ROAM_STATUS_ON - On * 0x02 - SYS_ROAM_STATUS_BLINK - Blinking * 0x03 - SYS_ROAM_STATUS_OUT_OF_NEIGHBORHOOD - Out of the neighborhood * 0x04 - SYS_ROAM_STATUS_OUT_OF_BLDG - Out of the building * 0x05 - SYS_ROAM_STATUS_PREF_SYS - Preferred system * 0x06 - SYS_ROAM_STATUS_AVAIL_SYS - Available system * 0x07 - SYS_ROAM_STATUS_ALLIANCE_PARTNER - Alliance partner * 0x08 - SYS_ROAM_STATUS_PREMIUM_PARTNER - Premium partner * 0x09 - SYS_ROAM_STATUS_FULL_SVC - Full service * 0x0A - SYS_ROAM_STATUS_PARTIAL_SVC - Partial service * 0x0B - SYS_ROAM_STATUS_BANNER_ON - Banner is on * 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.
<i>sysForbidden-Valid</i>	<ul style="list-style-type: none"> Indicates whether the forbidden system is valid.
<i>sysForbidden</i>	<ul style="list-style-type: none"> Whether the system is forbidden: <ul style="list-style-type: none"> 0x00 Not forbidden 0x01 Forbidden
<i>lacValid</i>	<ul style="list-style-type: none"> Indicates whether the location area code is valid.
<i>lac</i>	<ul style="list-style-type: none"> Location area code (only applicable for 3GPP).
<i>cellIdValid</i>	<ul style="list-style-type: none"> Indicates whether the cell ID is valid.
<i>cellId</i>	<ul style="list-style-type: none"> Cell ID.

<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> Type of service domain in which the registration is rejected. Values: <ul style="list-style-type: none"> 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched 0x04 - SYS_SRV_DOMAIN_CAMPED - Camped
<i>rejCause</i>	<ul style="list-style-type: none"> 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.
<i>nwldValid</i>	<ul style="list-style-type: none"> Indicates whether the network ID is valid.
<i>MCC</i>	<ul style="list-style-type: none"> MCC digits in ASCII characters. For CDMA, the MCC wildcard value is returned as {3, 0xFF, 0xFF}..
<i>MNC</i>	<ul style="list-style-type: none"> MNC digits in ASCII characters. For this field: Unused byte is set to 0xFF 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}.
<i>tacValid</i>	<ul style="list-style-type: none"> Indicates whether the tracking area code is valid.
<i>tac</i>	<ul style="list-style-type: none"> Tracking area code (only applicable for NR5G).
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present or not

8.239.2 Field Documentation

8.239.2.1 `uint32_t nas_NR5GSystemInfoTlv::cellId`

8.239.2.2 `uint8_t nas_NR5GSystemInfoTlv::cellIdValid`

8.239.2.3 `uint16_t nas_NR5GSystemInfoTlv::lac`

8.239.2.4 `uint8_t nas_NR5GSystemInfoTlv::lacValid`

8.239.2.5 `int8_t nas_NR5GSystemInfoTlv::MCC[3]`

8.239.2.6 `int8_t nas_NR5GSystemInfoTlv::MNC[3]`

- 8.239.2.7 uint8_t nas_NR5GSystemInfoTlv::nwIdValid
- 8.239.2.8 uint8_t nas_NR5GSystemInfoTlv::regRejectInfoValid
- 8.239.2.9 uint8_t nas_NR5GSystemInfoTlv::rejCause
- 8.239.2.10 uint8_t nas_NR5GSystemInfoTlv::rejectSrvDomain
- 8.239.2.11 uint8_t nas_NR5GSystemInfoTlv::roamStatus
- 8.239.2.12 uint8_t nas_NR5GSystemInfoTlv::roamStatusValid
- 8.239.2.13 uint8_t nas_NR5GSystemInfoTlv::srvcapability
- 8.239.2.14 uint8_t nas_NR5GSystemInfoTlv::srvCapValid
- 8.239.2.15 uint8_t nas_NR5GSystemInfoTlv::srvDomain
- 8.239.2.16 uint8_t nas_NR5GSystemInfoTlv::srvDomainValid
- 8.239.2.17 uint8_t nas_NR5GSystemInfoTlv::sysForbidden
- 8.239.2.18 uint8_t nas_NR5GSystemInfoTlv::sysForbiddenValid
- 8.239.2.19 uint16_t nas_NR5GSystemInfoTlv::tac
- 8.239.2.20 uint8_t nas_NR5GSystemInfoTlv::tacValid
- 8.239.2.21 uint8_t nas_NR5GSystemInfoTlv::TlvPresent

8.240 nas_operatorNameString Struct Reference

Data Fields

- uint8_t [PLMNName](#) [255]

8.240.1 Detailed Description

This structure contains PLMN name.

Parameters

<i>PLMNName</i>	<ul style="list-style-type: none">• PLMN name must be coded in a default 7-bit alphabet with b8 set to 0.
-----------------	---

8.240.2 Field Documentation

- 8.240.2.1 uint8_t nas_operatorNameString::PLMNName[255]

8.241 nas_OperatorPLMNData Struct Reference

Data Fields

- uint8_t [mcc](#) [3]
- uint8_t [mnc](#) [3]
- uint16_t [lac1](#)
- uint16_t [lac2](#)
- uint8_t [PLMNRecID](#)

8.241.1 Detailed Description

This structure contains Operator PLMN.

Parameters

<i>mcc</i>	<ul style="list-style-type: none"> • 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).
<i>mnc</i>	<ul style="list-style-type: none"> • 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).
<i>lac1</i>	<ul style="list-style-type: none"> • Location area code 1.
<i>lac2</i>	<ul style="list-style-type: none"> • Location area code 1.
<i>PLMNRecID</i>	<ul style="list-style-type: none"> • PLMN network name record identifier.

8.241.2 Field Documentation

8.241.2.1 uint16_t nas_OperatorPLMNData::lac1

8.241.2.2 uint16_t nas_OperatorPLMNData::lac2

8.241.2.3 uint8_t nas_OperatorPLMNData::mcc[3]

8.241.2.4 uint8_t nas_OperatorPLMNData::mnc[3]

8.241.2.5 uint8_t nas_OperatorPLMNData::PLMNRecID

8.242 nas_operatorPLMNList Struct Reference

Data Fields

- uint16_t [numInstance](#)
- [nas_OperatorPLMNData](#) [PLMNData](#) [255]

8.242.1 Detailed Description

This structure contains Operator PLMN List.

Parameters

<i>numInstance</i>	<ul style="list-style-type: none"> Number of sets of the elements.
<i>PLMNData</i>	<ul style="list-style-type: none"> Refer OperatorPLMNData for details (Optional).

8.242.2 Field Documentation

8.242.2.1 `uint16_t nas_operatorPLMNList::numInstance`

8.242.2.2 `nas_OperatorPLMNData nas_operatorPLMNList::PLMNData[255]`

8.243 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](#)

8.243.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Pcell Information.

Parameters

<i>pci</i>	<ul style="list-style-type: none"> Physical cell ID of the SCell Range. Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none"> Frequency of the absolute cell Range. Range for ID values: 0 to 65535.
<i>dl_bw_value</i>	<ul style="list-style-type: none"> Downlink Bandwidth Values. See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information.
<i>scell_state</i>	<ul style="list-style-type: none"> Scell state Values. See NAS_LTE_CPHY_SCELL_STATE_LITE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present.

8.243.2 Field Documentation

8.243.2.1 **NAS_LTE_CPHY_CA_BW_NRB_LITE** nas_PhyCaAggPcellInfo::dl_bw_value

8.243.2.2 int nas_PhyCaAggPcellInfo::freq

8.243.2.3 int nas_PhyCaAggPcellInfo::lTEbandValue

8.243.2.4 int nas_PhyCaAggPcellInfo::pci

8.243.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

8.244 nas_PhyCaAggScellIDIBw Struct Reference

Data Fields

- [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) dl_bw_value
- uint8_t [TlvPresent](#)

8.244.1 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"> • Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information.
--------------------	---

8.244.2 Field Documentation

8.244.2.1 **NAS_LTE_CPHY_CA_BW_NRB_LITE** nas_PhyCaAggScellIDIBw::dl_bw_value

8.244.2.2 uint8_t nas_PhyCaAggScellIDIBw::TlvPresent

8.245 nas_PhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t [scell_idx](#)
- uint8_t [TlvPresent](#)

8.245.1 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"> • Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
------------------	--

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

8.245.2 Field Documentation

8.245.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

8.245.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

8.246 nas_PhyCaAggScellIndType Struct Reference

Data Fields

- int [pci](#)
- int [freq](#)
- [NAS_LTE_CPHY_SCELL_STATE_LITE](#) [scell_state](#)
- uint8_t [TlvPresent](#)

8.246.1 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"> • Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none"> • Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
<i>scell_state</i>	<ul style="list-style-type: none"> • Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE_LITE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.246.2 Field Documentation

8.246.2.1 int nas_PhyCaAggScellIndType::freq

8.246.2.2 int nas_PhyCaAggScellIndType::pci

8.246.2.3 [NAS_LTE_CPHY_SCELL_STATE_LITE](#) nas_PhyCaAggScellIndType::scell_state

8.246.2.4 uint8_t nas_PhyCaAggScellIndType::TlvPresent

8.247 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](#)

8.247.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Information.

Parameters

<i>pci</i>	<ul style="list-style-type: none">• Physical cell ID of the SCell Range.• Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none">• Frequency of the absolute cell Range.• Range for ID values: 0 to 65535.
<i>dl_bw_value</i>	<ul style="list-style-type: none">• Downlink Bandwidth Values.• See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information.

<i>iLTEbandValue</i>	<ul style="list-style-type: none"> • Band value. • Range for LTE Band class 120 to 160. <ul style="list-style-type: none"> – 120 - LTE E-UTRA Operating Band 1 – 121 - LTE E-UTRA Operating Band 2 – 122 - LTE E-UTRA Operating Band 3 – 123 - LTE E-UTRA Operating Band 4 – 124 - LTE E-UTRA Operating Band 5 – 125 - LTE E-UTRA Operating Band 6 – 126 - LTE E-UTRA Operating Band 7 – 127 - LTE E-UTRA Operating Band 8 – 128 - LTE E-UTRA Operating Band 9 – 129 - LTE E-UTRA Operating Band 10 – 130 - LTE E-UTRA Operating Band 11 – 131 - LTE E-UTRA Operating Band 12 – 132 - LTE E-UTRA Operating Band 13 – 133 - LTE E-UTRA Operating Band 14 – 134 - LTE E-UTRA Operating Band 17 – 135 - LTE E-UTRA Operating Band 33 – 136 - LTE E-UTRA Operating Band 34 – 137 - LTE E-UTRA Operating Band 35 – 138 - LTE E-UTRA Operating Band 36 – 139 - LTE E-UTRA Operating Band 37 – 140 - LTE E-UTRA Operating Band 38 – 141 - LTE E-UTRA Operating Band 39 – 142 - LTE E-UTRA Operating Band 40 – 143 - LTE E-UTRA Operating Band 18 – 144 - LTE E-UTRA Operating Band 19 – 145 - LTE E-UTRA Operating Band 20 – 146 - LTE E-UTRA Operating Band 21 – 147 - LTE E-UTRA Operating Band 24 – 148 - LTE E-UTRA Operating Band 25 – 149 - LTE E-UTRA Operating Band 41 – 150 - LTE E-UTRA Operating Band 42 – 151 - LTE E-UTRA Operating Band 43 – 152 - LTE E-UTRA Operating Band 23 – 153 - LTE E-UTRA Operating Band 26 – 154 - LTE E-UTRA Operating Band 32 – 155 - LTE E-UTRA Operating Band 125 – 156 - LTE E-UTRA Operating Band 126 – 157 - LTE E-UTRA Operating Band 127 – 158 - LTE E-UTRA Operating Band 28 – 159 - LTE E-UTRA Operating Band 29 – 160 - LTE E-UTRA Operating Band 30
<i>scell_state</i>	<ul style="list-style-type: none"> • Scell state Values. • See NAS_LTE_CPHY_SCELL_STATE_LITE for more information.

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

8.247.2 Field Documentation

8.247.2.1 **NAS_LTE_CPHY_CA_BW_NRB_LITE** nas_PhyCaAggScellInfo::dl_bw_value

8.247.2.2 int nas_PhyCaAggScellInfo::freq

8.247.2.3 uint16_t nas_PhyCaAggScellInfo::ltebandValue

8.247.2.4 int nas_PhyCaAggScellInfo::pci

8.247.2.5 **NAS_LTE_CPHY_SCELL_STATE_LITE** nas_PhyCaAggScellInfo::scell_state

8.247.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

8.248 nas_PilotSetData Struct Reference

Data Fields

- uint8_t NumPilots
- nas_PilotSetParams * pPilotSetInfo

8.248.1 Detailed Description

This structure contains pilot sets data.

Parameters

<i>NumPilots</i>	<ul style="list-style-type: none"> • Number of Pilot Sets • As input specifies number of sets of parameter pPilotSetInfo for which memory has been assigned • As output specifies the actual number of sets of parameter pPilotSetInfo returned by device
<i>pPilotSetInfo</i>	<ul style="list-style-type: none"> • Pilot Set Parameters • See nas_PilotSetParams for more information.

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.

8.248.2 Field Documentation

8.248.2.1 uint8_t nas_PilotSetData::NumPilots

8.248.2.2 nas_PilotSetParams* nas_PilotSetData::pPilotSetInfo

8.249 nas_PilotSetParams Struct Reference

Data Fields

- uint32_t [PilotType](#)
- uint16_t [PilotPN](#)
- uint16_t [PilotStrength](#)

8.249.1 Detailed Description

This structure contains pilot sets parameters.

Parameters

<i>PilotType</i>	<ul style="list-style-type: none">• 0x00 - NAS_HRPD_PILOT_CURR_ACT_PLT Current Active Pilot• 0x01 - NAS_HRPD_PILOT_NEIGHBOR_PLT Neighbor pilot information
<i>PilotPN</i>	<ul style="list-style-type: none">• Pilot PN sequence offset index
<i>PilotStrength</i>	<ul style="list-style-type: none">• Strength of the pilot (in dB)

8.249.2 Field Documentation

8.249.2.1 uint16_t nas_PilotSetParams::PilotPN

8.249.2.2 uint16_t nas_PilotSetParams::PilotStrength

8.249.2.3 uint32_t nas_PilotSetParams::PilotType

8.250 nas_PlmnID Struct Reference

Data Fields

- uint16_t [mcc](#)
- uint16_t [mnc](#)
- uint8_t [pcsDigit](#)
- uint8_t [TlvPresent](#)

8.250.1 Detailed Description

This structure contains PLMN ID information

Parameters

<i>mcc</i>	<ul style="list-style-type: none">• A 16-bit integer representation of MCC. Range: 0 to 999
<i>mnc</i>	<ul style="list-style-type: none">• A 16-bit integer representation of MNC. Range: 0 to 999

<i>pcsDigit</i>	<ul style="list-style-type: none"> This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field. Values: <ul style="list-style-type: none"> – TRUE - MNC is a three-digit value; for example, a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a two-digit value; for example, a reported value of 90 corresponds to an MNC value of 90
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present or not

8.250.2 Field Documentation

8.250.2.1 uint16_t nas_PlmnID::mcc

8.250.2.2 uint16_t nas_PlmnID::mnc

8.250.2.3 uint8_t nas_PlmnID::pcsDigit

8.250.2.4 uint8_t nas_PlmnID::TlvPresent

8.251 nas_PLMNNetworkName Struct Reference

Data Fields

- uint8_t numInstance
- nas_PLMNNetworkNameData PLMNNetName [255]

8.251.1 Detailed Description

This structure contains NITZ Information list.

Parameters

<i>numInstance</i>	<ul style="list-style-type: none"> Number of sets of the elements.
<i>PLMNNetName</i>	<ul style="list-style-type: none"> Refer nas_PLMNNetworkNameData for details (Optional).

8.251.2 Field Documentation

8.251.2.1 uint8_t nas_PLMNNetworkName::numInstance

8.251.2.2 nas_PLMNNetworkNameData nas_PLMNNetworkName::PLMNNetName[255]

8.252 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]

8.252.1 Detailed Description

This structure contains NITZ Information.

Parameters

<i>codingScheme</i>	<ul style="list-style-type: none"> • Coding scheme: <ul style="list-style-type: none"> – 0 - CODING_SCHEME_CELL_BROADCAST_GSM - Cell broadcast data coding scheme, GSM default alphabet, language unspecified;defined in 3GPP TS 23.038. – 1 - CODING_SCHEME_UCS2 - UCS2 (16 bit);defined in ISO/IEC 10646
<i>countryInitials</i>	<ul style="list-style-type: none"> • Country's initials: <ul style="list-style-type: none"> – 0 - COUNTRY_INITIALS_DO_NOT_ADD - MS should not add the letters for the country's initials to the text string. – 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.
<i>longNameSpare-Bits</i>	<ul style="list-style-type: none"> • Long Name Spare Bits: <ul style="list-style-type: none"> – 1 - SPARE_BITS_8 - Bit 8 is spare and set to 0 in octet n – 2 - SPARE_BITS_7_TO_8 - Bits 7 and 8 are spare and set to 0 in octet n. – 3 - SPARE_BITS_6_TO_8 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n. – 4 - SPARE_BITS_5_TO_8 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n. – 5 - SPARE_BITS_4_TO_8 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n. – 6 - SPARE_BITS_3_TO_8 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n. – 7 - SPARE_BITS_2_TO_8 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n. – 0 - SPARE_BITS_UNKNOWN - Carries no information about the number of spare bits in octet n.
<i>shortName-SpareBits</i>	<ul style="list-style-type: none"> • Short Name Spare Bits: <ul style="list-style-type: none"> – 1 - SPARE_BITS_8 - Bit 8 is spare and set to 0 in octet n. – 2 - SPARE_BITS_7_TO_8 - Bits 7 and 8 are spare and set to 0 in octet n. – 3 - SPARE_BITS_6_TO_8 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n. – 4 - SPARE_BITS_5_TO_8 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n. – 5 - SPARE_BITS_4_TO_8 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n. – 6 - SPARE_BITS_3_TO_8 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n. – 7 - SPARE_BITS_2_TO_8 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n. – 0 - SPARE_BITS_UNKNOWN - Carries no information about the number of spare bits in octet n.

<i>longNameLen</i>	<ul style="list-style-type: none"> • It provides the length of long name.
<i>longName</i>	<ul style="list-style-type: none"> • Long name string in coding_scheme.
<i>shortNameLen</i>	<ul style="list-style-type: none"> • It provides the length of short name.
<i>shortName</i>	<ul style="list-style-type: none"> • Short name string in coding_scheme.

8.252.2 Field Documentation

8.252.2.1 uint8_t nas_PLMNNetworkNameData::codingScheme

8.252.2.2 uint8_t nas_PLMNNetworkNameData::countryInitials

8.252.2.3 uint8_t nas_PLMNNetworkNameData::longName[255]

8.252.2.4 uint8_t nas_PLMNNetworkNameData::longNameLen

8.252.2.5 uint8_t nas_PLMNNetworkNameData::longNameSpareBits

8.252.2.6 uint8_t nas_PLMNNetworkNameData::shortName[255]

8.252.2.7 uint8_t nas_PLMNNetworkNameData::shortNameLen

8.252.2.8 uint8_t nas_PLMNNetworkNameData::shortNameSpareBits

8.253 nas_PRLPrefTlv Struct Reference

Data Fields

- uint16_t [PRLPref](#)
- uint8_t [TlvPresent](#)

8.253.1 Detailed Description

Contain the PRL preference for system selection preferences.

Parameters

<i>PRLPref</i>	<ul style="list-style-type: none"> • Parameter indicating the CDMA PRL Preference • Values: <ul style="list-style-type: none"> – 0x0001 - Acquire available system only on the A side – 0x0002 - Acquire available system only on the B side – 0x3FFF - Acquire any available systems
----------------	---

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

8.253.2 Field Documentation

8.253.2.1 uint16_t nas_PRLPrefTlv::PRLPref

8.253.2.2 uint8_t nas_PRLPrefTlv::TlvPresent

8.254 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](#)

8.254.1 Detailed Description

This structure contains protocol subtype element.

Parameters

<i>PhysicalLayer</i>	<ul style="list-style-type: none"> • Specifies Physical Layer Protocol subtype
<i>ControlMac</i>	<ul style="list-style-type: none"> • Specifies Control Channel MAC Protocol subtype
<i>AccessMac</i>	<ul style="list-style-type: none"> • Specifies Access Channel MAC Protocol subtype
<i>ForwardMac</i>	<ul style="list-style-type: none"> • Specifies Forward Traffic Channel MAC Protocol subtype
<i>ReverseMac</i>	<ul style="list-style-type: none"> • Specifies Reverse Traffic Channel MAC Protocol subtype
<i>KeyExchange</i>	<ul style="list-style-type: none"> • Specifies Key exchange Protocol subtype

<i>AuthProt</i>	<ul style="list-style-type: none"> • Specifies Authentication Protocol subtype
<i>EncryptProt</i>	<ul style="list-style-type: none"> • Specifies Encryption Protocol subtype
<i>SecProt</i>	<ul style="list-style-type: none"> • Specifies Security Protocol subtype
<i>IdleState</i>	<ul style="list-style-type: none"> • Specifies Idle state Protocol subtype
<i>MultDisc</i>	<ul style="list-style-type: none"> • Specifies Generic multimode capability discovery Protocol subtype
<i>VirtStream</i>	<ul style="list-style-type: none"> • Specifies Generic Virtual Stream Protocol subtype

8.254.2 Field Documentation

8.254.2.1 `uint16_t nas_protocolSubtypeElement::AccessMac`

8.254.2.2 `uint16_t nas_protocolSubtypeElement::AuthProt`

8.254.2.3 `uint16_t nas_protocolSubtypeElement::ControlMac`

8.254.2.4 `uint16_t nas_protocolSubtypeElement::EncryptProt`

8.254.2.5 `uint16_t nas_protocolSubtypeElement::ForwardMac`

8.254.2.6 `uint16_t nas_protocolSubtypeElement::IdleState`

8.254.2.7 `uint16_t nas_protocolSubtypeElement::KeyExchange`

8.254.2.8 `uint16_t nas_protocolSubtypeElement::MultDisc`

8.254.2.9 `uint16_t nas_protocolSubtypeElement::PhysicalLayer`

8.254.2.10 `uint16_t nas_protocolSubtypeElement::ReverseMac`

8.254.2.11 `uint16_t nas_protocolSubtypeElement::SecProt`

8.254.2.12 `uint16_t nas_protocolSubtypeElement::VirtStream`

8.255 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- `uint8_t leapSeconds`
- `uint8_t localTimeOffset`
- `uint8_t daylightSavings`

8.255.1 Detailed Description

This structure contains the 3GPP2TimeZone parameters

Parameters

<i>leapSeconds</i>	<ul style="list-style-type: none"> • leap seconds - Number of leap seconds since the start of CDMA system time.
<i>localTimeOffset</i>	<ul style="list-style-type: none"> • 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.
<i>daylightSavings</i>	<ul style="list-style-type: none"> • Day Light Savings Indicator <ul style="list-style-type: none"> – 0x00 - OFF (daylight savings not in effect) – 0x01 - ON (daylight savings in effect)

8.255.2 Field Documentation

8.255.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.255.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.255.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

8.256 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]

8.256.1 Detailed Description

Contain the 3GPP network information.

Parameters

<i>MCC</i>	<ul style="list-style-type: none"> • Mobile Country Code
<i>MNC</i>	<ul style="list-style-type: none"> • Mobile Network Code

<i>InUse</i>	<ul style="list-style-type: none"> • Is the Network the current serving Network <ul style="list-style-type: none"> – 0 - Unknown – 1 - Current serving network – 2 - Not current serving network, available
<i>Roaming</i>	<ul style="list-style-type: none"> • Home/Roam Status of the Network <ul style="list-style-type: none"> – 0 - Unknown – 1 - Home – 2 - Roam
<i>Forbidden</i>	<ul style="list-style-type: none"> • Is the Network in the forbidden network list <ul style="list-style-type: none"> – 0 - Unknown – 1 - Forbidden – 2 - Not Forbidden
<i>Preferred</i>	<ul style="list-style-type: none"> • Is the Network in the Preferred network list <ul style="list-style-type: none"> – 0 - Unknown – 1 - Preferred – 2 - Not Preferred
<i>Description</i>	<ul style="list-style-type: none"> • Network Name/Description • This is a NULL terminated string.

8.256.2 Field Documentation

8.256.2.1 `char nas_QmiNas3GppNetworkInfo::Description[255]`

8.256.2.2 `uint32_t nas_QmiNas3GppNetworkInfo::Forbidden`

8.256.2.3 `uint32_t nas_QmiNas3GppNetworkInfo::InUse`

8.256.2.4 `uint16_t nas_QmiNas3GppNetworkInfo::MCC`

8.256.2.5 `uint16_t nas_QmiNas3GppNetworkInfo::MNC`

8.256.2.6 `uint32_t nas_QmiNas3GppNetworkInfo::Preferred`

8.256.2.7 `uint32_t nas_QmiNas3GppNetworkInfo::Roaming`

8.257 nas_QmiNas3GppNetworkRAT Struct Reference

Data Fields

- `uint16_t` [MCC](#)
- `uint16_t` [MNC](#)
- `uint8_t` [RAT](#)

8.257.1 Detailed Description

Contain the 3GPP radio access technology information.

Parameters

<i>MCC</i>	<ul style="list-style-type: none">• Mobile Country Code
<i>MNC</i>	<ul style="list-style-type: none">• Mobile Network Code
<i>RAT</i>	<ul style="list-style-type: none">• Radio Access Technology<ul style="list-style-type: none">– 0x04 - GERAN– 0x05 - UMTS– 0x08 - LTE– 0x09 - TD-SCDMA

8.257.2 Field Documentation

8.257.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.257.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.257.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.258 nas_QmisNasPcsDigit Struct Reference

Data Fields

- uint16_t [MCC](#)
- uint16_t [MNC](#)
- uint8_t [includes_pcs_digit](#)

8.258.1 Detailed Description

Contain the PCS Digit information

Parameters

<i>MCC</i>	<ul style="list-style-type: none">• Mobile Country Code
<i>MNC</i>	<ul style="list-style-type: none">• Mobile Network Code
<i>includes_pcs_digit</i>	<ul style="list-style-type: none">• this field is use to interpret the length of corresponding MNC reported• 0x01 - MNC is a three-digit value• 0x00 - MNC is a two-digit value

8.258.2 Field Documentation

8.258.2.1 `uint8_t nas_QmisNasPcsDigit::includes_pcs_digit`

8.258.2.2 `uint16_t nas_QmisNasPcsDigit::MCC`

8.258.2.3 `uint16_t nas_QmisNasPcsDigit::MNC`

8.259 `nas_QmisNasSlqsNasPCICellInfo` Struct Reference

Data Fields

- `uint32_t freq`
- `uint16_t cellID`
- `uint32_t GlobalCellID`
- `uint8_t PlmnLen`
- `nas_QmisNasPcsDigit nasQmisNasPcsDigit` [255]

8.259.1 Detailed Description

Contain the PCI Cell Info.

Parameters

<i>freq</i>	<ul style="list-style-type: none"> • Absolute cell's frequency. Range 0 to 65535.
<i>cellID</i>	<ul style="list-style-type: none"> • Cell ID.
<i>GlobalCellID</i>	<ul style="list-style-type: none"> • Global Cell ID.
<i>PlmnLen</i>	<ul style="list-style-type: none"> • Number of Plmn(<code>nasQmisNasPcsDigit</code>) information sets.
<i>nasQmisNasPcsDigit</i>	<ul style="list-style-type: none"> • See nas_QmisNasPcsDigit for more information.

8.259.2 Field Documentation

8.259.2.1 `uint16_t nas_QmisNasSlqsNasPCICellInfo::cellID`

8.259.2.2 `uint32_t nas_QmisNasSlqsNasPCICellInfo::freq`

8.259.2.3 `uint32_t nas_QmisNasSlqsNasPCICellInfo::GlobalCellID`

8.259.2.4 `nas_QmisNasPcsDigit nas_QmisNasSlqsNasPCICellInfo::nasQmisNasPcsDigit`[255]

8.259.2.5 `uint8_t nas_QmisNasSlqsNasPCICellInfo::PlmnLen`

8.260 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](#)

8.260.1 Detailed Description

Contain the NAS Perform network scan PCI information

Parameters

<i>PCICellInfoLen</i>	<ul style="list-style-type: none"> • Number of PCI Cell Info(PCICellInfo) information sets.
<i>nasQmisNasSlqsNasPCICellInfo</i>	<ul style="list-style-type: none"> • See nas_QmisNasSlqsNasPCICellInfo for more information.
<i>rsrp</i>	<ul style="list-style-type: none"> • Combined RSRP.
<i>rsrpRx0</i>	<ul style="list-style-type: none"> • Rx0 RSRP.
<i>rsrpRx1</i>	<ul style="list-style-type: none"> • Rx1 RSRP.
<i>rsrq</i>	<ul style="list-style-type: none"> • Combined RSRQ.
<i>rsrqRx0</i>	<ul style="list-style-type: none"> • Rx0 RSRQ.
<i>rsrpRx1</i>	<ul style="list-style-type: none"> • Rx1 RSRQ.

8.260.2 Field Documentation

8.260.2.1 [nas_QmisNasSlqsNasPCICellInfo](#) [nas_QmisNasSlqsNasPCIInfo::nasQmisNasSlqsNasPCICellInfo](#)[255]

8.260.2.2 uint8_t [nas_QmisNasSlqsNasPCIInfo::PCICellInfoLen](#)

8.260.2.3 uint16_t [nas_QmisNasSlqsNasPCIInfo::rsrp](#)

8.260.2.4 uint16_t [nas_QmisNasSlqsNasPCIInfo::rsrpRx0](#)

8.260.2.5 uint16_t nas_QmisNasSlqsNasPCInfo::rsrpRx1

8.260.2.6 uint16_t nas_QmisNasSlqsNasPCInfo::rsrq

8.260.2.7 uint16_t nas_QmisNasSlqsNasPCInfo::rsrqRx0

8.260.2.8 uint16_t nas_QmisNasSlqsNasPCInfo::rsrqRx1

8.261 nas_RankIndicatorTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [count0](#)
- uint32_t [count1](#)

8.261.1 Detailed Description

Elements for Rank indicator TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • value is 1 if TLV is present, otherwise 0
<i>count0</i>	<ul style="list-style-type: none"> • count of Rank1
<i>count1</i>	<ul style="list-style-type: none"> • count of Rank2

8.261.2 Field Documentation

8.261.2.1 uint32_t nas_RankIndicatorTlv::count0

8.261.2.2 uint32_t nas_RankIndicatorTlv::count1

8.261.2.3 uint8_t nas_RankIndicatorTlv::TlvPresent

8.262 nas_RatDisabledMaskTlv Struct Reference

Data Fields

- uint16_t [ratDisabledMask](#)
- uint8_t [TlvPresent](#)

8.262.1 Detailed Description

Contain the RAT disabled mask

Parameters

<i>ratDisabledMask</i>	<ul style="list-style-type: none"> • Bitmask representing the radio technologies that are disabled. • Bit values: <ul style="list-style-type: none"> – Bit 0 (0x01) - cdma2000 @ 1X – Bit 1 (0x02) - cdma2000 @ HRPD(1xEV-DO) – Bit 2 (0x04) - GSM – Bit 3 (0x08) - UMTS – Bit 4 (0x10) - LTE – Bit 5 (0x20) - TD-SCDMA – Bit 6 (0x40) - NR5G All unlisted bits are reserved for future use and the service point ignores them if used.
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.262.2 Field Documentation

8.262.2.1 uint16_t nas_RatDisabledMaskTlv::ratDisabledMask

8.262.2.2 uint8_t nas_RatDisabledMaskTlv::TlvPresent

8.263 nas_RejectReasonTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [serviceDomain](#)
- uint32_t [rejectCause](#)

8.263.1 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

8.263.2 Field Documentation

8.263.2.1 uint32_t nas_RejectReasonTlv::rejectCause

8.263.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

8.263.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

8.264 nas_RFBandInfoElements Struct Reference

Data Fields

- uint8_t [radioInterface](#)
- uint16_t [activeBandClass](#)
- uint16_t [activeChannel](#)

8.264.1 Detailed Description

This structure contains the RFBandInfo response parameters.

Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio interface technology <ul style="list-style-type: none"> – See Tables for Radio Interface
<i>activeBandClass</i>	<ul style="list-style-type: none"> • Active Band Class <ul style="list-style-type: none"> – See Tables for Band Classes
<i>activeChannel</i>	<ul style="list-style-type: none"> • Active channel (0 if channel is not relevant to the reported technology)

8.264.2 Field Documentation

8.264.2.1 uint16_t nas_RFBandInfoElements::activeBandClass

8.264.2.2 uint16_t nas_RFBandInfoElements::activeChannel

8.264.2.3 uint8_t nas_RFBandInfoElements::radioInterface

8.265 nas_RfBandInfoExtFormat Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t * [pInstancesSize](#)
- struct [nas_RfBandInfoExtFormatElements](#) * [pRfBandInfoExtFormatParam](#)

8.265.1 Detailed Description

This structure contains the [nas_RfBandInfoExtFormat](#) response parameters.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present
<i>pInstancesSize[IN/OUT]</i>	<ul style="list-style-type: none"> • Number of sets of the RfBandInfoElements • Upon input, maximum number of elements that the RF info instances array can contain. • Upon successful output, actual number of elements in RF info instances array.
<i>pRfBandInfoExtFormatParam</i>	<ul style="list-style-type: none"> • See nas_RfBandInfoExtFormatElements for more information

8.265.2 Field Documentation

8.265.2.1 `uint8_t* nas_RfBandInfoExtFormat::pInstancesSize`8.265.2.2 `struct nas_RfBandInfoExtFormatElements* nas_RfBandInfoExtFormat::pRfBandInfoExtFormatParam`8.265.2.3 `uint8_t nas_RfBandInfoExtFormat::TlvPresent`

8.266 nas_RfBandInfoExtFormatElements Struct Reference

Data Fields

- `uint8_t` [radioInterface](#)
- `uint16_t` [activeBand](#)
- `uint32_t` [activeChannel](#)

8.266.1 Detailed Description

This structure contains the RF Band Information, Extended Format.

Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio interface technology <ul style="list-style-type: none"> – See Tables for Radio Interface
<i>activeBand</i>	<ul style="list-style-type: none"> • Active Band Class <ul style="list-style-type: none"> – See Tables for Band Classes
<i>activeChannel</i>	<ul style="list-style-type: none"> • Active channel (0 if channel is not relevant to the reported technology)

8.266.2 Field Documentation

8.266.2.1 `uint16_t nas_RfBandInfoExtFormatElements::activeBand`8.266.2.2 `uint32_t nas_RfBandInfoExtFormatElements::activeChannel`

8.266.2.3 `uint8_t nas_RfBandInfoExtFormatElements::radioInterface`

8.267 nas_RfBandInfoList Struct Reference

Data Fields

- `uint8_t * pInstanceSize`
- `struct nas_RFBandInfoElements * pRFBandInfoParam`

8.267.1 Detailed Description

This structure contains the RfBandInfoList response parameters.

Parameters

<i>pInstanceSize[IN/OUT]</i>	<ul style="list-style-type: none"> • Number of sets of the RFBandInfoElements • Upon input, maximum number of elements that the RF info instances array can contain. • Upon successful output, actual number of elements in RF info instances array.
<i>pRFBandInfoParam</i>	<ul style="list-style-type: none"> • See nas_RFBandInfoElements for more information

8.267.2 Field Documentation

8.267.2.1 `uint8_t* nas_RfBandInfoList::pInstanceSize`

8.267.2.2 `struct nas_RFBandInfoElements* nas_RfBandInfoList::pRFBandInfoParam`

8.268 nas_RfBandwidthInfo Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t * pInstancesSize`
- `struct nas_RfBandwidthInfoElements * pRfBandwidthInfoParam`

8.268.1 Detailed Description

This structure contains the [nas_RfBandwidthInfo](#) response parameters.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present
<i>pInstancesSize[IN/OUT]</i>	<ul style="list-style-type: none"> • Number of sets of the RFBandInfoElements • Upon input, maximum number of elements that the RF info instances array can contain. • Upon successful output, actual number of elements in RF info instances array.

<i>pRfBandwidth-InfoParam</i>	<ul style="list-style-type: none"> See nas_RfBandwidthInfoElements for more information
-------------------------------	--

8.268.2 Field Documentation

8.268.2.1 `uint8_t* nas_RfBandwidthInfo::pInstancesSize`

8.268.2.2 `struct nas_RfBandwidthInfoElements* nas_RfBandwidthInfo::pRfBandwidthInfoParam`

8.268.2.3 `uint8_t nas_RfBandwidthInfo::TlvPresent`

8.269 nas_RfBandwidthInfoElements Struct Reference

Data Fields

- `uint8_t` [radioInterface](#)
- `uint32_t` [bandwidth](#)

8.269.1 Detailed Description

This structure contains the RF Bandwidth Information List parameters.

Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> Radio interface technology <ul style="list-style-type: none"> See Tables for Radio Interface
<i>bandwidth</i>	<ul style="list-style-type: none"> Bandwidth <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> * NAS_LTE_BW_NRB_6 (0) - LTE 1.4 MHz bandwidth * NAS_LTE_BW_NRB_15 (1) - LTE 3 MHz bandwidth * NAS_LTE_BW_NRB_25 (2) - LTE 5 MHz bandwidth * NAS_LTE_BW_NRB_50 (3) - LTE 10 MHz bandwidth * NAS_LTE_BW_NRB_75 (4) - LTE 15 MHz bandwidth * NAS_LTE_BW_NRB_100 (5) - LTE 20 MHz bandwidth * NAS_NR5G_BW_NRB_6 (6) - NR5G 1.4 MHz bandwidth * NAS_NR5G_BW_NRB_15 (7) - NR5G 3 MHz bandwidth * NAS_NR5G_BW_NRB_25 (8) - NR5G 5 MHz bandwidth * NAS_NR5G_BW_NRB_50 (9) - NR5G 10 MHz bandwidth * NAS_NR5G_BW_NRB_75 (10) - NR5G 15 MHz bandwidth * NAS_NR5G_BW_NRB_100 (11) - NR5G 20 MHz bandwidth

8.269.2 Field Documentation

8.269.2.1 `uint32_t nas_RfBandwidthInfoElements::bandwidth`

8.269.2.2 `uint8_t nas_RfBandwidthInfoElements::radioInterface`

8.270 `nas_RfDedicatedBandInfo` Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t * pInstancesSize`
- `struct nas_RfDedicatedBandInfoElements * pRfDedicatedBandInfoParam`

8.270.1 Detailed Description

This structure contains the `nas_RfDedicatedBandInfo` response parameters.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present
<i>pInstancesSize[-IN/OUT]</i>	<ul style="list-style-type: none"> • Number of sets of the <code>RFBandInfoElements</code> • Upon input, maximum number of elements that the RF info instances array can contain. • Upon successful output, actual number of elements in RF info instances array.
<i>pRfDedicated-BandInfoParam</i>	<ul style="list-style-type: none"> • See <code>nas_RfDedicatedBandInfoElements</code> for more information

8.270.2 Field Documentation

8.270.2.1 `uint8_t* nas_RfDedicatedBandInfo::pInstancesSize`

8.270.2.2 `struct nas_RfDedicatedBandInfoElements* nas_RfDedicatedBandInfo::pRfDedicatedBandInfoParam`

8.270.2.3 `uint8_t nas_RfDedicatedBandInfo::TlvPresent`

8.271 `nas_RfDedicatedBandInfoElements` Struct Reference

Data Fields

- `uint8_t radioInterface`
- `uint16_t dedicatedBand`

8.271.1 Detailed Description

This structure contains the RF Dedicated Band Info.

Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio interface technology <ul style="list-style-type: none"> – See Tables for Radio Interface
-----------------------	--

<i>dedicatedBand</i>	<ul style="list-style-type: none"> Dedicated Band Class <ul style="list-style-type: none"> See Tables for Band Classes
----------------------	---

8.271.2 Field Documentation

8.271.2.1 uint16_t nas_RfDedicatedBandInfoElements::dedicatedBand

8.271.2.2 uint8_t nas_RfDedicatedBandInfoElements::radioInterface

8.272 nas_RFInfoTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [radioInterfaceSize](#)
- uint32_t [radioInterface](#) [255]
- uint32_t [activeBandClass](#) [255]
- uint32_t [activeChannel](#) [255]

8.272.1 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>radioInterface-Size</i>	Number of sets of the following elements <ul style="list-style-type: none"> radioInterface activeBandClass activeChannel
<i>radioInterface</i>	radio interface technology of the signal being measured <ul style="list-style-type: none"> 0x01 - cdma2000 1X 0x02 - cdma2000 HRPD (1xEV-DO) 0x03 - AMPS 0x04 - GSM 0x05 - UMTS 0x08 - LTE 0x09 - TD-SCDMA
<i>activeBandClass</i>	active band class <ul style="list-style-type: none"> 00 to 39 - CDMA band classes 40 to 79 - GSM band classes 80 to 91 - WCDMA band classes 120 to 160 - LTE band classes 200 to 205 - TD-SCDMA band classes
<i>activeChannel</i>	active channel <ul style="list-style-type: none"> Active channel. If the channel is not relevant to the technology, a value of 0 is returned.

8.272.2 Field Documentation

8.272.2.1 `uint32_t nas_RFInfoTlv::activeBandClass[255]`

8.272.2.2 `uint32_t nas_RFInfoTlv::activeChannel[255]`

8.272.2.3 `uint32_t nas_RFInfoTlv::radioInterface[255]`

8.272.2.4 `uint8_t nas_RFInfoTlv::radioInterfaceSize`

8.272.2.5 `uint8_t nas_RFInfoTlv::TlvPresent`

8.273 nas_roamIndList Struct Reference

Data Fields

- `uint8_t numInstances`
- `uint8_t radioInterface [32]`
- `uint8_t roamIndicator [32]`

8.273.1 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"> • number of sets of radio interface currently in use and roaming indicator <ul style="list-style-type: none"> – defaults to zero
<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio Interface currently in use • Values: <ul style="list-style-type: none"> – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE
<i>roamIndicator</i>	<ul style="list-style-type: none"> • Roaming Indicator • Values: <ul style="list-style-type: none"> – 0x00 - Roaming – 0x01 - Home

8.273.2 Field Documentation

8.273.2.1 uint8_t nas_roamIndList::numInstances

8.273.2.2 uint8_t nas_roamIndList::radioInterface[32]

8.273.2.3 uint8_t nas_roamIndList::roamIndicator[32]

8.274 nas_RoamPrefTlv Struct Reference

Data Fields

- uint16_t [RoamPref](#)
- uint8_t [TlvPresent](#)

8.274.1 Detailed Description

Contain the roaming preference for system selection preferences.

Parameters

<i>RoamPref</i>	<ul style="list-style-type: none"> • Parameter indicating the roaming Preference • Values: <ul style="list-style-type: none"> – 0x01 - Acquire only systems for which the roaming indicator is off – 0x02 - Acquire a system as long as its roaming indicator is not off – 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only – 0xFF - Acquire systems, regardless of their roaming indicator
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.274.2 Field Documentation

8.274.2.1 uint16_t nas_RoamPrefTlv::RoamPref

8.274.2.2 uint8_t nas_RoamPrefTlv::TlvPresent

8.275 nas_RSRPThresh Struct Reference

Data Fields

- uint8_t [RSRPThresListLen](#)
- int16_t * [pRSRPThresList](#)

8.275.1 Detailed Description

This structure contains RSRP threshold list.

Parameters

<i>RSRPThresListLen</i>	<ul style="list-style-type: none"> Length of the RSRP threshold list parameter to follow
<i>pRSRPThresList</i>	<ul style="list-style-type: none"> Sequence of thresholds delimiting current RSRP event reporting bands 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"> RSRP values are applicable only for LTE RSRP values are measured in dBm, with a range of -44 dBm to -140 dBm Each RSRP threshold value is a signed byte value Maximum number of threshold values is 16 At least one value must be specified

8.275.2 Field Documentation

8.275.2.1 `int16_t* nas_RSRPThresh::pRSRPThresList`8.275.2.2 `uint8_t nas_RSRPThresh::RSRPThresListLen`8.276 `nas_rsrqInformation` Struct Reference

Data Fields

- `int8_t` [rsrq](#)
- `uint8_t` [radiolf](#)

8.276.1 Detailed Description

This structure contains the RSRQ Information

Parameters

<i>rsrq</i>	<ul style="list-style-type: none"> RSRQ value in dB (signed integer value); valid range is -3 to -20 (-3 means -3 dB, -20 means -20 dB)
<i>radiolf</i>	<ul style="list-style-type: none"> Radio interface technology of the signal being measured <ul style="list-style-type: none"> 0x08 - LTE

8.276.2 Field Documentation

8.276.2.1 `uint8_t nas_rsrqInformation::radiolf`8.276.2.2 `int8_t nas_rsrqInformation::rsrq`8.277 `nas_RSRQThresh` Struct Reference

Data Fields

- uint8_t [RSRQThresListLen](#)
- int8_t * [pRSRQThresList](#)

8.277.1 Detailed Description

This structure contains RSRQ threshold list.

Parameters

<i>RSRQThresListLen</i>	<ul style="list-style-type: none"> • Length of the RSRQ threshold list parameter to follow
<i>pRSRQThresList</i>	<ul style="list-style-type: none"> • Sequence of thresholds delimiting current RSRQ event reporting bands • 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"> – RSRQ values are applicable only for LTE – RSRQ values are measured in dBm, with a range of -20 dBm to -3 dBm – Each RSRQ threshold value is a signed byte value – Maximum number of threshold values is 16 – At least one value must be specified

8.277.2 Field Documentation

8.277.2.1 int8_t* nas_RSRQThresh::pRSRQThresList

8.277.2.2 uint8_t nas_RSRQThresh::RSRQThresListLen

8.278 nas_RSSIThresh Struct Reference

Data Fields

- uint8_t [RSSIThresListLen](#)
- int8_t * [pRSSIThresList](#)

8.278.1 Detailed Description

This sturcture contain RSSI threshold list.

Parameters

<i>RSSIThresListLen</i>	<ul style="list-style-type: none"> Length of the RSSI threshold list parameter to follow
<i>pRSSIThresList</i>	<ul style="list-style-type: none"> RSSI in dBm(signed bytes) A value of -125 dBm or lower is used to indicate No Signal RSSI values have the following ranges (in dBm) <ul style="list-style-type: none"> CDMA is -105 to -21 HDR is -118 to -13 GSM is -111 to -48 WCDMA is -121 to 0 LTE is -120 to 0 Threshold values specified above are used for all RATs The maximum number of threshold values is 16, each a signed byte value.

8.278.2 Field Documentation

8.278.2.1 `int8_t* nas_RSSIThresh::pRSSIThresList`8.278.2.2 `uint8_t nas_RSSIThresh::RSSIThresListLen`8.279 `nas_rxInfo` Struct Reference

Data Fields

- `uint8_t isRadioTuned`
- `int32_t rxPower`
- `int32_t ecio`
- `int32_t rscp`
- `int32_t rsrp`
- `uint32_t phase`

8.279.1 Detailed Description

This structure contains rx radio information.

Parameters

<i>isRadioTuned</i>	<ul style="list-style-type: none"> Whether Rx is tuned to a channel: <ul style="list-style-type: none"> 0x00 - Not tuned 0x01 - Tuned 0xFF - Not Available If the radio is tuned, instantaneous values are set for the signal information fields below. If the radio is not tuned, or is delayed or invalid, the values are set depending on each technology.
<i>rx_pwr</i>	<ul style="list-style-type: none"> Rx power value in 1/10 dbm resolution.

<i>ecio</i>	<ul style="list-style-type: none"> • ECIO in 1/10 dB; valid for CDMA, HDR, GSM, WCDMA, and LTE.
<i>rscp</i>	<ul style="list-style-type: none"> • Received signal code power in 1/10 dbm. • Valid for WCDMA.
<i>rsrp</i>	<ul style="list-style-type: none"> • Current reference signal received power in 1/10 dbm valid for LTE.
<i>phase</i>	<ul style="list-style-type: none"> • Phase in 1/100 degrees; valid for LTE. • When the phase is unknown, 0xFFFFFFFF is used.

8.279.2 Field Documentation

8.279.2.1 int32_t nas_rxInfo::ecio

8.279.2.2 uint8_t nas_rxInfo::isRadioTuned

8.279.2.3 uint32_t nas_rxInfo::phase

8.279.2.4 int32_t nas_rxInfo::rscp

8.279.2.5 int32_t nas_rxInfo::rsrp

8.279.2.6 int32_t nas_rxInfo::rxPower

8.280 nas_RxSigInfo Struct Reference

Data Fields

- uint8_t [rxChainIndex](#)
- uint8_t [isRadioTuned](#)
- int32_t [rxPower](#)
- int32_t [rsrp](#)

8.280.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

Parameters

<i>rxChainIndex</i>	<ul style="list-style-type: none"> • Rx antenna path • Valid Values <ul style="list-style-type: none"> – 0 - Primary Rx – 1 - Diversity Rx
---------------------	---

<i>isRadioTuned</i>	<ul style="list-style-type: none"> • Rx path is tuned to a channel or Not • Values <ul style="list-style-type: none"> – 0x00 - Not tuned – 0x01 - Tuned
---------------------	--

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"> • Rx power value in 1/10 dBm resolution
<i>rsrp</i>	<ul style="list-style-type: none"> • Current reference signal received power in 1/10 dBm resolution

8.280.2 Field Documentation

8.280.2.1 `uint8_t nas_RxSigInfo::isRadioTuned`

8.280.2.2 `int32_t nas_RxSigInfo::rsrp`

8.280.2.3 `uint8_t nas_RxSigInfo::rxChainIndex`

8.280.2.4 `int32_t nas_RxSigInfo::rxPower`

8.281 nas_rxSignalStrengthListElement Struct Reference**Data Fields**

- `int16_t rxSignalStrength`
- `uint8_t radioIlf`

8.281.1 Detailed Description

This structure contains the Received Signal Strength Information

Parameters

<i>rxSignalStrength</i>	<ul style="list-style-type: none"> • Received signal strength in dBm <ul style="list-style-type: none"> – For CDMA and UMTS, this indicates forward link pilotEc. – For GSM, the received signal strength. – For LTE, this indicates the total received wideband power observed by UE.
-------------------------	---

<i>radioIf</i>	<ul style="list-style-type: none"> Radio interface technology of the signal being radio_if measured <ul style="list-style-type: none"> 0x00 - RADIO_IF_NO_SVC - None (no service) 0x01 - RADIO_IF_CDMA_1X - cdma2000@ 1X 0x02 - RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO) 0x03 - RADIO_IF_AMPS - AMPS 0x04 - RADIO_IF_GSM - GSM 0x05 - RADIO_IF_UMTS - UMTS 0x08 - RADIO_IF_LTE - LTE
----------------	---

Note

First element of the RSSI list always contains the current Signal strength and Radio Interface.

8.281.2 Field Documentation

8.281.2.1 `uint8_t nas_rxSignalStrengthListElement::radioIf`

8.281.2.2 `int16_t nas_rxSignalStrengthListElement::rxSignalStrength`

8.282 nas_SccRxInfo Struct Reference**Data Fields**

- `int32_t rsrq`
- `int16_t snr`
- `uint8_t numInstances`
- `nas_RxSigInfo sigInfo` [255]
- `uint8_t TlvPresent`

8.282.1 Detailed Description

This structure contains information about the SccRxInfo parameters.

Parameters

<i>rsrq</i>	<ul style="list-style-type: none"> Current reference signal Receive quality in 1/10 dB resolution
<i>snr</i>	<ul style="list-style-type: none"> Reference signal signal-to-noise ratio in dB. Range -10 to 30
<i>numInstances</i>	<ul style="list-style-type: none"> Number of sets of the following <ul style="list-style-type: none"> rxChainIndex isRadioTuned rxPower rsrp

<i>sigInfo</i>	<ul style="list-style-type: none"> • See nas_RxSigInfo for more information
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.282.2 Field Documentation

8.282.2.1 `uint8_t nas_SccRxInfo::numInstances`

8.282.2.2 `int32_t nas_SccRxInfo::rsrq`

8.282.2.3 `nas_RxSigInfo nas_SccRxInfo::sigInfo[255]`

8.282.2.4 `int16_t nas_SccRxInfo::snr`

8.282.2.5 `uint8_t nas_SccRxInfo::TlvPresent`

8.283 nas_serviceProviderName Struct Reference

Data Fields

- `uint8_t displayCondition`
- `uint8_t spnLength`
- `uint8_t spn [255]`

8.283.1 Detailed Description

This structure contains service provider name.

Parameters

<i>displayCondition</i>	<ul style="list-style-type: none"> • Display condition.
<i>spnLength</i>	<ul style="list-style-type: none"> • It provides length of spn.
<i>spn</i>	<ul style="list-style-type: none"> • 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.

8.283.2 Field Documentation

8.283.2.1 `uint8_t nas_serviceProviderName::displayCondition`

8.283.2.2 `uint8_t nas_serviceProviderName::spn[255]`

8.283.2.3 `uint8_t nas_serviceProviderName::spnLength`

8.284 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]

8.284.1 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"> • Registration state - Registration state of the mobile • Values: <ul style="list-style-type: none"> – 0 - Not Registered; mobile is not currently searching for a new network to provide service – 1 - Registered with a network – 2 - Not registered, but mobile is currently searching for a new network to provide service – 3 - Registration denied by visible network – 4 - Registration state is unknown
<i>csAttachState</i>	<ul style="list-style-type: none"> • CS Attach State - Circuit-switched domain attach state of the mobile • Values: <ul style="list-style-type: none"> – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
<i>psAttachState</i>	<ul style="list-style-type: none"> • PS Attach State - Packet-switched domain attach state of the mobile • Values: <ul style="list-style-type: none"> – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
<i>selNetwork</i>	<ul style="list-style-type: none"> • Selected Network - Type of selected radio access network • Values: <ul style="list-style-type: none"> – 0 - Unknown – 1 - 3GPP2 network – 2 - 3GPP network
<i>numRadio-Interfaces</i>	<ul style="list-style-type: none"> • In Use Radio Interfaces Number <ul style="list-style-type: none"> – Number of radio interfaces currently in use – defaults to zero

<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio Interface currently in use • Values: <ul style="list-style-type: none"> – 0x00 - RADIO_IF_NO_SVC - None(no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1xEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE
-----------------------	--

8.284.2 Field Documentation

8.284.2.1 `uint8_t nas_servSystem::csAttachState`

8.284.2.2 `uint8_t nas_servSystem::numRadioInterfaces`

8.284.2.3 `uint8_t nas_servSystem::psAttachState`

8.284.2.4 `uint8_t nas_servSystem::radioInterface[32]`

8.284.2.5 `uint8_t nas_servSystem::regState`

8.284.2.6 `uint8_t nas_servSystem::selNetwork`

8.285 nas_sidNid Struct Reference

Data Fields

- `uint16_t` [nid](#)
- `uint16_t` [sid](#)

8.285.1 Detailed Description

This structure contains the parameters for SidNid Information

Parameters

<i>nid</i>	<ul style="list-style-type: none"> • Network ID
<i>sid</i>	<ul style="list-style-type: none"> • System ID

8.285.2 Field Documentation

8.285.2.1 `uint16_t nas_sidNid::nid`

8.285.2.2 `uint16_t nas_sidNid::sid`

8.286 nas_SignalStrengthTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- int8_t [signalStrength](#)
- uint32_t [radioInterface](#)

8.286.1 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

8.286.2 Field Documentation

8.286.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

8.286.2.2 int8_t nas_SignalStrengthTlv::signalStrength

8.286.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

8.287 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](#)

8.287.1 Detailed Description

Structure for storing the input parameters passed for SLQSSetSignalStrengthsCallback by the user.

Parameters

<i>rxSignalStrengthDelta</i>	<ul style="list-style-type: none"> • RSSI delta(in dBm) at which an event report indication, including the current RSSI, will be sent to the requesting control point.
------------------------------	---

<i>ecioDelta</i>	<ul style="list-style-type: none"> • ECIO delta at which an event report indication, <i>ecioDelta</i> including the current ECIO, will be sent to the requesting control point. • ECIO delta is an unsigned 1 byte value that increments in negative 0.5 dBm, e.g., <i>ecio_delta</i> of 2 means a change of -1 dBm.
<i>ioDelta</i>	<ul style="list-style-type: none"> • IO delta (in dBm) at which an event report indication, <i>ioDelta</i> including the current IO, will be sent to the requesting control point.
<i>sinrDelta</i>	<ul style="list-style-type: none"> • SINR delta level at which an event report indication, <i>sinrDelta</i> including the current SINR, will be sent to the requesting control point.
<i>rsrqDelta</i>	<ul style="list-style-type: none"> • RSRQ delta level at which an event report indication, including the current RSRQ, will be sent to the requesting control point.
<i>ecioThreshold-ListLen</i>	<ul style="list-style-type: none"> • Number of elements in the ECIO threshold list.
<i>ecioThreshold-List</i>	<ul style="list-style-type: none"> • 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"> – Maximum number of threshold values is 10 – At least one value must be specified.
<i>sinrThreshold-ListLen</i>	<ul style="list-style-type: none"> • Number of elements in the SINR threshold list.
<i>sinrThreshold-List</i>	<ul style="list-style-type: none"> • 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 <i>sinr</i> value is sent to the requesting control point. For this field: <ul style="list-style-type: none"> – Maximum number of threshold values is 5 – At least one value must be specified.
<i>ltesnrdelta</i>	<ul style="list-style-type: none"> • 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., <i>lte_snr_delta</i> of 3 means a change 0.3dB.
<i>ltersrpdelta</i>	<ul style="list-style-type: none"> • -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.

8.287.2 Field Documentation

8.287.2.1 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta`

8.287.2.2 `int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]`

8.287.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.287.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

8.287.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

8.287.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

8.287.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

8.287.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.287.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

8.287.2.10 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.287.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.288 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](#)

8.288.1 Detailed Description

This structure contains paramaters signal strengths information.

Parameters

<i>rxSignal- StrengthInfo</i>	<ul style="list-style-type: none"> • signal strength info list • Bit to check in ParamPresenceMask - 19
<i>ecioInfo</i>	<ul style="list-style-type: none"> • ecio info list • Bit to check in ParamPresenceMask - 20
<i>io</i>	<ul style="list-style-type: none"> • received IO in dBm; IO is only applicable for 1xEV-DO • Bit to check in ParamPresenceMask - 21
<i>sinr</i>	<ul style="list-style-type: none"> • SINR level • Bit to check in ParamPresenceMask - 22

<i>errorRateInfo</i>	<ul style="list-style-type: none"> error rate info Bit to check in ParamPresenceMask - 23
<i>rsrqInfo</i>	<ul style="list-style-type: none"> rsrq info Bit to check in ParamPresenceMask - 24
<i>lteSnrinfo</i>	<ul style="list-style-type: none"> lte Snr information Bit to check in ParamPresenceMask - 25
<i>lteRsrpinfo</i>	<ul style="list-style-type: none"> lte rsrp info Bit to check in ParamPresenceMask - 26

8.288.2 Field Documentation

8.288.2.1 `nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo`

8.288.2.2 `nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo`

8.288.2.3 `uint32_t nas_SLQSSignalStrengthsInformation::io`

8.288.2.4 `nas_lteRsrpInformation nas_SLQSSignalStrengthsInformation::lteRsrpinfo`

8.288.2.5 `nas_lteSnrInformation nas_SLQSSignalStrengthsInformation::lteSnrinfo`

8.288.2.6 `nas_rsrqInformation nas_SLQSSignalStrengthsInformation::rsrqInfo`

8.288.2.7 `nas_rxSignalStrengthListElement nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo`

8.288.2.8 `uint8_t nas_SLQSSignalStrengthsInformation::sinr`

8.289 nas_SLQSSignalStrengthsTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `nas_SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo`

8.289.1 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

8.289.2 Field Documentation

8.289.2.1 nas_SLQSSignalStrengthsInformation nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

8.289.2.2 uint8_t nas_SLQSSignalStrengthsTlv::TlvPresent

8.290 nas_SrvDomainPrefTlv Struct Reference

Data Fields

- uint32_t [SrvDomainPref](#)
- uint8_t [TlvPresent](#)

8.290.1 Detailed Description

Contain the Service domain Preference for system selection preferences.

Parameters

<i>SrvDomainPref</i>	<ul style="list-style-type: none"> • Parameter indicating Service domain preference • Values: <ul style="list-style-type: none"> – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present

8.290.2 Field Documentation

8.290.2.1 uint32_t nas_SrvDomainPrefTlv::SrvDomainPref

8.290.2.2 uint8_t nas_SrvDomainPrefTlv::TlvPresent

8.291 nas_SrvStatusInfo Struct Reference

Data Fields

- uint8_t [srvStatus](#)
- uint8_t [isPrefDataPath](#)

8.291.1 Detailed Description

Structure for storing the service status information for CDMA and HDR networks.

Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"> • Service status of the system. <ul style="list-style-type: none"> – 0x00 - No service – 0x01 - Limited service – 0x02 - Service – 0x03 - Limited regional service – 0x04 - Power save – 0xFF - Not Available
<i>isPrefDataPath</i>	<ul style="list-style-type: none"> • Whether the RAT is the preferred data path. <ul style="list-style-type: none"> – 0x00 - Not preferred – 0x01 - Preferred – 0xFF - Not Available

8.291.2 Field Documentation

8.291.2.1 `uint8_t nas_SrvStatusInfo::isPrefDataPath`8.291.2.2 `uint8_t nas_SrvStatusInfo::srvStatus`8.292 `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`

8.292.1 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"> • Indicates whether the service domain is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
-----------------------	--

<i>srvDomain</i>	<ul style="list-style-type: none">• Service domain registered on the system.<ul style="list-style-type: none">– 0x00 - No service– 0x01 - Circuit-switched only– 0x02 - Packet-switched only– 0x03 - Circuit-switched and packet-switched– 0x04 - Camped– 0xFF - Not Available
<i>srvCapability-Valid</i>	<ul style="list-style-type: none">• Indicates whether the service capability is valid.<ul style="list-style-type: none">– 0x00 - Invalid– 0x01 - Valid– 0xFF - Not Available
<i>srvCapability</i>	<ul style="list-style-type: none">• Current system's service capability.<ul style="list-style-type: none">– 0x00 - No service– 0x01 - Circuit-switched only– 0x02 - Packet-switched only– 0x03 - Circuit-switched and packet-switched– 0x04 - Camped– 0xFF - Not Available
<i>roamStatusValid</i>	<ul style="list-style-type: none">• Indicates whether the roaming status is valid.<ul style="list-style-type: none">– 0x00 - Invalid– 0x01 - Valid– 0xFF - Not Available

<i>roamStatus</i>	<ul style="list-style-type: none"> • Current roaming status. <ul style="list-style-type: none"> – 0x00 - Off – 0x01 - On – 0x02 - Blinking – 0x03 - Out of the neighborhood – 0x04 - Out of the building – 0x05 - Preferred system – 0x06 - Available system – 0x07 - Alliance partner – 0x08 - Premium partner – 0x09 - Full service – 0x0A - Partial service – 0x0B - Banner is on – 0x0C - Banner is off – 0x0D to 0x3F - Reserved for Standard Enhanced Roaming Indicator Numbers – 0x40 to 0x7F - Reserved for Non-Standard Enhanced Roaming Indicator Numbers – 0x80 to 0xFF - Reserved. – 0xFF - Not Available • Values from 0x02 onward are only applicable for 3GPP2
<i>isSysForbidden-Valid</i>	<ul style="list-style-type: none"> • Indicates whether the forbidden system is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>isSysForbidden</i>	<ul style="list-style-type: none"> • Whether the system is forbidden. <ul style="list-style-type: none"> – 0x00 - Not forbidden – 0x01 - Forbidden – 0xFF - Not Available

8.292.2 Field Documentation

8.292.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

8.292.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

8.292.2.3 uint8_t nas_sysInfoCommon::roamStatus

8.292.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

8.292.2.5 uint8_t nas_sysInfoCommon::srvCapability

8.292.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

8.292.2.7 uint8_t nas_sysInfoCommon::srvDomain

8.292.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

8.293 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

- uint8_t [TDSCDMAECIOThreshListLen](#)
- float * [pTDSCDMAECIOThreshList](#)

8.293.1 Detailed Description

This structure contains TDSCDMA ECIO threshold related parameters.

Parameters

<i>TDSCDMAECIOThreshListLen</i>	<ul style="list-style-type: none"> • Length of the TDSCDMA ECIO threshold list parameter to follow
<i>pTDSCDMAECIOThreshList</i>	<ul style="list-style-type: none"> • Array of ECIO thresholds (in dB) used by TD-SCDMA • Maximum of 32 values.

8.293.2 Field Documentation

8.293.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.293.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.294 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

- uint8_t [TDSCDMARSCPThreshListLen](#)
- int16_t * [pTDSCDMARSCPThreshList](#)

8.294.1 Detailed Description

This structure contains TDSCDMA RSCP threshold related parameters.

Parameters

<i>TDSCDMARSCPThreshListLen</i>	<ul style="list-style-type: none"> • Length of the TDSCDMA RSCP threshold list parameter to follow
<i>pTDSCDMARSCPThreshList</i>	<ul style="list-style-type: none"> • Array of RSCP thresholds (in units of 0.1 dBm) • Maximum of 32 values • Range for RSCP values: -120 to -25 (in dBm).

8.294.2 Field Documentation

8.294.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.294.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.295 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

- uint8_t [TDSCDMARSSIThreshListLen](#)
- float * [pTDSCDMARSSIThreshList](#)

8.295.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

Parameters

<i>TDSCDMARSSIThreshListLen</i>	<ul style="list-style-type: none"> • Length of the TDSCDMA RSSI threshold list parameter to follow
<i>pTDSCDMARSSIThreshList</i>	<ul style="list-style-type: none"> • Array of RSSI thresholds (in dBm) used by TD-SCDMA • Maximum of 32 values.

8.295.2 Field Documentation

8.295.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.295.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.296 nas_TDSCDMASINRCONFThresh Struct Reference

Data Fields

- uint8_t [TDSCDMASINRCONFThreshListLen](#)
- float * [pTDSCDMASINRCONFThreshList](#)

8.296.1 Detailed Description

This structure contains TDSCDMA SINR threshold list.

Parameters

<i>TDSCDMASINRCONFThreshListLen</i>	<ul style="list-style-type: none"> • Length of the TDSCDMA SINR threshold list parameter to follow
<i>pTDSCDMASINRCONFThreshList</i>	<ul style="list-style-type: none"> • Array of SINR thresholds (in dB) used by TD-SCDMA • Maximum of 32 values

8.296.2 Field Documentation

8.296.2.1 float* nas_TDSCDMASINRCONFTthresh::pTDSCDMASINRCONFTthreshList

8.296.2.2 uint8_t nas_TDSCDMASINRCONFTthresh::TDSCDMASINRCONFTthreshListLen

8.297 nas_TDSCDMASINRThresh Struct Reference

Data Fields

- uint8_t [TDSCDMASINRThreshListLen](#)
- float * [pTDSCDMASINRThreshList](#)

8.297.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

Parameters

<i>TDSCDMASINRThreshListLen</i>	<ul style="list-style-type: none"> • Length of the TDSCDMA SINR threshold list parameter to follow
<i>pTDSCDMASINRThreshList</i>	<ul style="list-style-type: none"> • Array of SINR thresholds (in dB) used by TD-SCDMA • Maximum of 32 values

8.297.2 Field Documentation

8.297.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.297.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.298 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](#)

8.298.1 Detailed Description

This structure contains the parameters for Network Time.

Parameters

<i>year</i>	<ul style="list-style-type: none"> • Year
<i>month</i>	<ul style="list-style-type: none"> • Month • 1 is January and 12 is December
<i>day</i>	<ul style="list-style-type: none"> • Day • Range - 1 to 31
<i>hour</i>	<ul style="list-style-type: none"> • Hour • Range - 0 to 59
<i>minute</i>	<ul style="list-style-type: none"> • Minute • Range - 0 to 59
<i>second</i>	<ul style="list-style-type: none"> • Second • Range - 0 to 59
<i>dayOfWeek</i>	<ul style="list-style-type: none"> • Day of the week • 0 is Monday and 6 is Sunday
<i>timeZone</i>	<ul style="list-style-type: none"> • Offset from Universal time • The difference between local time and Universal time, in increments of 15 min • Signed Value
<i>dayLtSavingAdj</i>	<ul style="list-style-type: none"> • Daylight saving adjustment in hours • Possible values - 0, 1, and 2. • This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVD0
<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio interface from which the information comes • Values <ul style="list-style-type: none"> – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.298.2 Field Documentation

8.298.2.1 uint8_t nas_timeInfo::day

8.298.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

8.298.2.3 uint8_t nas_timeInfo::dayOfWeek

8.298.2.4 uint8_t nas_timeInfo::hour

8.298.2.5 uint8_t nas_timeInfo::minute

8.298.2.6 uint8_t nas_timeInfo::month

8.298.2.7 uint8_t nas_timeInfo::radioInterface

8.298.2.8 uint8_t nas_timeInfo::second

8.298.2.9 int8_t nas_timeInfo::timeZone

8.298.2.10 uint8_t nas_timeInfo::TlvPresent

8.298.2.11 uint16_t nas_timeInfo::year

8.299 nas_trueIMSI Struct Reference

Data Fields

- uint8_t [mccT](#) [3]
- uint16_t [imsiT1112](#)
- uint8_t [imsiTS1](#) [7]
- uint8_t [imsiTS2](#) [3]
- uint8_t [imsiTaddrNum](#)

8.299.1 Detailed Description

This structure contains True IMSI.

Parameters

<i>mccT</i>	<ul style="list-style-type: none"> • ASCII character representation of MCC_T
<i>imsiT1112</i>	<ul style="list-style-type: none"> • ASCII character representation of IMSI_T_11_12 value <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>imsiTS1</i>	<ul style="list-style-type: none"> • ASCII character representation of IMSI_T_S1 value
<i>imsiTS2</i>	<ul style="list-style-type: none"> • ASCII character representation of IMSI_T_S2 value

<i>imsiTaddrNum</i>	<ul style="list-style-type: none"> Value of IMSI_T_ADDR_NUM <ul style="list-style-type: none"> 0xFF - Not Available
---------------------	--

8.299.2 Field Documentation

8.299.2.1 uint16_t nas_trueIMSI::imsiT1112

8.299.2.2 uint8_t nas_trueIMSI::imsiTaddrNum

8.299.2.3 uint8_t nas_trueIMSI::imsiTS1[7]

8.299.2.4 uint8_t nas_trueIMSI::imsiTS2[3]

8.299.2.5 uint8_t nas_trueIMSI::mccT[3]

8.300 nas_txInfo Struct Reference

Data Fields

- uint8_t [isInTraffic](#)
- int32_t [txPower](#)

8.300.1 Detailed Description

This structure contains tx radio information.

Parameters

<i>isInTraffic</i>	<ul style="list-style-type: none"> Whether the device is in traffic. <ul style="list-style-type: none"> 0x00 - not in traffic 0x01 - in traffic The txPower field is only meaningful when in the device is in traffic.
<i>txPower</i>	<ul style="list-style-type: none"> Tx power value in 1/10 dbm.

8.300.2 Field Documentation

8.300.2.1 uint8_t nas_txInfo::isInTraffic

8.300.2.2 int32_t nas_txInfo::txPower

8.301 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]

8.301.1 Detailed Description

This structure contains information about the UMTS Network.

Parameters

<i>cellID</i>	<ul style="list-style-type: none"> • Cell ID. • 0xFFFFFFFF indicates cell ID information is not present.
<i>plmn[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • MCC/MNC information coded as octet 3, 4, and 5. • This field is ignored when nmrCellID is not present.
<i>lac</i>	<ul style="list-style-type: none"> • Location area code. • This field is ignored when nmrCellID is not present. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>uarfcn</i>	<ul style="list-style-type: none"> • UTRA absolute RF channel number. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>psc</i>	<ul style="list-style-type: none"> • Primary scrambling code. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>rscp</i>	<ul style="list-style-type: none"> • Received signal code power. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>ecio</i>	<ul style="list-style-type: none"> • ECIO(Signal-to-Interference-ratio). <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>umtsInst</i>	<ul style="list-style-type: none"> • Provides the number of set of UMTS info instances. • If 0(zero), then no information follows it.
<i>UMTSInstInfo[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_UMTSInstInfo for more information.

<i>geranInst</i>	<ul style="list-style-type: none"> Provides the number of set of GERAN info instances. If 0(zero), then no information follows it.
<i>GeranInstInfo[N-AS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> See nas_geranInstInfo for more information.

8.301.2 Field Documentation

8.301.2.1 uint16_t nas_UMTSInfo::cellID

8.301.2.2 int16_t nas_UMTSInfo::ecio

8.301.2.3 uint8_t nas_UMTSInfo::geranInst

8.301.2.4 nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]

8.301.2.5 uint16_t nas_UMTSInfo::lac

8.301.2.6 uint8_t nas_UMTSInfo::plmn[3]

8.301.2.7 uint16_t nas_UMTSInfo::psc

8.301.2.8 int16_t nas_UMTSInfo::rscp

8.301.2.9 uint16_t nas_UMTSInfo::uarfcn

8.301.2.10 uint8_t nas_UMTSInfo::umtsInst

8.301.2.11 nas_UMTSinstInfo nas_UMTSInfo::UMTSinstInfo[255]

8.302 nas_UMTSinstInfo Struct Reference

Data Fields

- uint16_t [umtsUarfcn](#)
- uint16_t [umtsPsc](#)
- int16_t [umtsRscp](#)
- int16_t [umtsEcio](#)

8.302.1 Detailed Description

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

Parameters

<i>umtsUarfcn</i>	<ul style="list-style-type: none"> UTRA absolute RF channel number.
<i>umtsPsc</i>	<ul style="list-style-type: none"> Primary scrambling code.

<i>umtsRscp</i>	<ul style="list-style-type: none"> Received signal code power.
<i>umtsEcio</i>	<ul style="list-style-type: none"> ECIO(Signal-to-Interference-ratio).

8.302.2 Field Documentation

8.302.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.302.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.302.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.302.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.303 nas_umtsLTENbrCell Struct Reference

Data Fields

- uint16_t [earfcn](#)
- uint16_t [pci](#)
- uint32_t [rsrp](#)
- uint32_t [rsrq](#)
- int16_t [srxlev](#)
- uint8_t [cellsTDD](#)

8.303.1 Detailed Description

This structure contains information about the UMTS LTE neighbour Cell.

Parameters

<i>earfcn</i>	<ul style="list-style-type: none"> E-UTRA absolute RF channel number of the detected cell.
<i>pci</i>	<ul style="list-style-type: none"> Physical cell ID of the detected cell. Range is defined in 3GPP TS 36.211
<i>rsrp</i>	<ul style="list-style-type: none"> Current received signal strength indication (in dBm) of the detected cell.
<i>rsrq</i>	<ul style="list-style-type: none"> Current reference signal received quality (in dB) of the detected cell.
<i>srxlev</i>	<ul style="list-style-type: none"> Cell selection Rx level (Srxlev) value of the detected cell in linear scale. This field is only valid when wcdma_rrc_state is not NAS_WCDMA_RRC_STATE_CEL_FACH or NAS_WCDMA_RRC_STATE_CELL_DCH.

<i>cellsTDD</i>	<ul style="list-style-type: none"> • TRUE if the cell is TDD; FALSE if the cell is FDD.
-----------------	--

8.303.2 Field Documentation

8.303.2.1 uint8_t nas_umtsLTENbrCell::cellsTDD

8.303.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.303.2.3 uint16_t nas_umtsLTENbrCell::pci

8.303.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.303.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.303.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.304 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](#)

8.304.1 Detailed Description

This structure contains the parameters for Universal Time Information.

Parameters

<i>year</i>	<ul style="list-style-type: none"> • Year.
<i>month</i>	<ul style="list-style-type: none"> • Month. <ul style="list-style-type: none"> – 1 is January and 12 is December.
<i>day</i>	<ul style="list-style-type: none"> • Day. <ul style="list-style-type: none"> – Range 1 to 31.
<i>hour</i>	<ul style="list-style-type: none"> • Hour. <ul style="list-style-type: none"> – Range 0 to 59.

<i>minute</i>	<ul style="list-style-type: none"> Minute. <ul style="list-style-type: none"> Range 0 to 59.
<i>second</i>	<ul style="list-style-type: none"> Second. <ul style="list-style-type: none"> Range 0 to 59.
<i>dayOfWeek</i>	<ul style="list-style-type: none"> Day of the Week. <ul style="list-style-type: none"> 0 is Monday and 6 is Sunday.

8.304.2 Field Documentation

8.304.2.1 uint8_t nas_UniversalTime::day

8.304.2.2 uint8_t nas_UniversalTime::dayOfWeek

8.304.2.3 uint8_t nas_UniversalTime::hour

8.304.2.4 uint8_t nas_UniversalTime::minute

8.304.2.5 uint8_t nas_UniversalTime::month

8.304.2.6 uint8_t nas_UniversalTime::second

8.304.2.7 uint16_t nas_UniversalTime::year

8.305 nas_wcdmaCellInfo Struct Reference

Data Fields

- uint16_t [psc](#)
- int16_t [cpich_rscp](#)
- int16_t [cpich_ecno](#)
- int16_t [srxlev](#)

8.305.1 Detailed Description

This structure contains information about the WCDMA Cell.

Parameters

<i>psc</i>	<ul style="list-style-type: none"> Primary scrambling code. Range: 0 to 511.
<i>cpich_rscp</i>	<ul style="list-style-type: none"> Absolute power level (in 1/10 dBm) of the common pilot channel as received by the UE. Range: -120.0 dBm to -25.0 dBm

<i>cpich_ecno</i>	<ul style="list-style-type: none"> • 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. • Range: -50.0 dB to 0.
<i>srxlev</i>	<ul style="list-style-type: none"> • Cell selection Rx level (Srxlev) value. • Range: -128 to 128. • This field is only valid when ue_in_idle is TRUE.

8.305.2 Field Documentation

8.305.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.305.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.305.2.3 uint16_t nas_wcdmaCellInfo::psc

8.305.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.306 nas_WCDMAECIOThresh Struct Reference

Data Fields

- uint8_t [WCDMAECIOThreshListLen](#)
- int16_t * [pWCDMAECIOThreshList](#)

8.306.1 Detailed Description

This structure contains WCDMA ECIO threshold related parameters.

Parameters

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

8.306.2 Field Documentation

8.306.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.306.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

8.307 nas_WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- uint32_t [wcdmaRRCState](#)
- uint8_t [umtsLTENbrCellLen](#)
- [nas_umtsLTENbrCell](#) UMTSLTENbrCell [255]

8.307.1 Detailed Description

This structure contains information about the WCDMA - LTE Neighboring Cell Info Set.

Parameters

<i>wcdmaRRC-State</i>	<ul style="list-style-type: none"> • WCDMA RRC states. • Defined in 3GPP TS 25.331 • Values: <ul style="list-style-type: none"> – 0x00 - NAS_WCDMA_RRC_STATE_DISCONNECTED <ul style="list-style-type: none"> * WCDMA RRC State is IDLE – 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH <ul style="list-style-type: none"> * WCDMA RRC state is CELL_PCH – 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH <ul style="list-style-type: none"> * WCDMA RRC state is URA_PCH – 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH <ul style="list-style-type: none"> * WCDMA RRC state is CELL_FACH – 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH <ul style="list-style-type: none"> * WCDMA RRC state is CELL_DCH
<i>umtsLTENbr-CellLen</i>	<ul style="list-style-type: none"> • Number of sets of UMTS LTE Neighbors.
<i>UMTSLTENbr-Cell</i>	<ul style="list-style-type: none"> • See nas_umtsLTENbrCell for more information.

8.307.2 Field Documentation

8.307.2.1 [nas_umtsLTENbrCell](#) nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.307.2.2 [uint8_t](#) nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.307.2.3 [uint32_t](#) nas_WCDMAInfoLTENeighborCell::wcdmaRRCState

8.308 nas_WCDMARSSIThresh Struct Reference

Data Fields

- [uint8_t](#) [WCDMARSSIThreshListLen](#)
- [int16_t](#) * [pWCDMARSSIThreshList](#)

8.308.1 Detailed Description

This structure contains WCDMA RSSI threshold related parameters.

Parameters

<i>WCDMARSSI- ThreshListLen</i>	<ul style="list-style-type: none"> Length of the WCDMA RSSI threshold list parameter to follow
<i>pWCDMARSSI- ThreshList</i>	<ul style="list-style-type: none"> Array of RSSI thresholds (in units of 0.1 dBm) Maximum of 32 values. Range for RSSI values: -121 to 0 (in dBm)

8.308.2 Field Documentation

8.308.2.1 `int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList`8.308.2.2 `uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen`8.309 `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`

8.309.1 Detailed Description

Structure for storing the WCDMA System Information.

Parameters

<i>sysInfoWCDMA</i>	<ul style="list-style-type: none"> See nas_sysInfoCommon for more information.
<i>lacValid</i>	<ul style="list-style-type: none"> Indicates whether the location area code is valid.. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available

<i>lac</i>	<ul style="list-style-type: none"> • Location area code. • Only applies to 3GPP. <ul style="list-style-type: none"> – 0xFFFF - Not Available
<i>cellIdValid</i>	<ul style="list-style-type: none"> • Indicates whether the cell ID is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>cellId</i>	<ul style="list-style-type: none"> • Cell ID. <ul style="list-style-type: none"> – 0xFFFFFFFF - Not Available
<i>regRejectInfo-Valid</i>	<ul style="list-style-type: none"> • Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> • Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> – 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched – 0x04 - Camped – 0xFF - Not Available
<i>rejCause</i>	<ul style="list-style-type: none"> • Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>networkIdValid</i>	<ul style="list-style-type: none"> • Indicates whether the network ID is valid. <ul style="list-style-type: none"> – 0x00 - Invalid – 0x01 - Valid – 0xFF - Not Available
<i>MCC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Country Code. • MCC digits in ASCII characters
<i>MNC[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Network Code. • MNC digits in ASCII characters • An unused byte is set to 0xFF. • 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.

<i>hsCallStatus-Valid</i>	<ul style="list-style-type: none"> Indicates whether the high-speed call status is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>hsCallStatus</i>	<ul style="list-style-type: none"> Call status on high speed. Only applicable for WCDMA. <ul style="list-style-type: none"> 0x00 - HSDPA and HSUPA are unsupported 0x01 - HSDPA is supported 0x02 - HSUPA is supported 0x03 - HSDPA and HSUPA are supported 0x04 - HSDPA+ is supported 0x05 - HSDPA+ and HSUPA are supported 0x06 - Dual-cell HSDPA+ is supported 0x07 - Dual-cell HSDPA+ and HSUPA are supported 0xFF - Not Available
<i>hsIndValid</i>	<ul style="list-style-type: none"> Indicates whether high-speed service indication is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>hsInd</i>	<ul style="list-style-type: none"> High-speed service indication Only applicable for WCDMA. <ul style="list-style-type: none"> 0x00 - HSDPA and HSUPA are unsupported 0x01 - HSDPA is supported 0x02 - HSUPA is supported 0x03 - HSDPA and HSUPA are supported 0x04 - HSDPA+ is supported 0x05 - HSDPA+ and HSUPA are supported 0x06 - Dual-cell HSDPA+ is supported 0x07 - Dual-cell HSDPA+ and HSUPA are supported 0xFF - Not Available
<i>pscValid</i>	<ul style="list-style-type: none"> Indicates whether primary scrambling code is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>psc</i>	<ul style="list-style-type: none"> Primary scrambling code. <ul style="list-style-type: none"> 0xFFFF - Not Available

8.309.2 Field Documentation

- 8.309.2.1 uint32_t nas_WCDMASysInfo::cellId
- 8.309.2.2 uint8_t nas_WCDMASysInfo::cellIdValid
- 8.309.2.3 uint8_t nas_WCDMASysInfo::hsCallStatus
- 8.309.2.4 uint8_t nas_WCDMASysInfo::hsCallStatusValid
- 8.309.2.5 uint8_t nas_WCDMASysInfo::hsInd
- 8.309.2.6 uint8_t nas_WCDMASysInfo::hsIndValid
- 8.309.2.7 uint16_t nas_WCDMASysInfo::lac
- 8.309.2.8 uint8_t nas_WCDMASysInfo::lacValid
- 8.309.2.9 uint8_t nas_WCDMASysInfo::MCC[3]
- 8.309.2.10 uint8_t nas_WCDMASysInfo::MNC[3]
- 8.309.2.11 uint8_t nas_WCDMASysInfo::networkIdValid
- 8.309.2.12 uint16_t nas_WCDMASysInfo::psc
- 8.309.2.13 uint8_t nas_WCDMASysInfo::pscValid
- 8.309.2.14 uint8_t nas_WCDMASysInfo::regRejectInfoValid
- 8.309.2.15 uint8_t nas_WCDMASysInfo::rejCause
- 8.309.2.16 uint8_t nas_WCDMASysInfo::rejectSrvDomain
- 8.309.2.17 nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA

8.310 nas_wcdmaUARFCN Struct Reference

Data Fields

- uint8_t [status](#)
- uint32_t [uarfcn](#)

8.310.1 Detailed Description

This structure contains WCDMA UARFCN information.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • 0 - Disable • 1 - Enable
<i>uarfcn</i>	<ul style="list-style-type: none"> • UARFCN to which UE is locked

8.310.2 Field Documentation

8.310.2.1 uint8_t nas_wcdmaUARFCN::status

8.310.2.2 uint32_t nas_wcdmaUARFCN::uarfcn

8.311 NASAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [AcqOrderLen](#)
- uint8_t [AcqOrderPref](#) [255]

8.311.1 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

8.311.2 Field Documentation

8.311.2.1 uint8_t NASAcqOrderPrefTlv::AcqOrderLen

8.311.2.2 uint8_t NASAcqOrderPrefTlv::AcqOrderPref[255]

8.311.2.3 uint8_t NASAcqOrderPrefTlv::TlvPresent

8.312 NASBandPreferenceTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint64_t [band_pref](#)

8.312.1 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

8.312.2 Field Documentation

8.312.2.1 uint64_t NASBandPreferenceTlv::band_pref

8.312.2.2 uint8_t NASBandPreferenceTlv::TlvPresent

8.313 NASCiotAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [CiotAcqOrderLen](#)
- uint32_t [CiotAcqOrderPref](#) [255]

8.313.1 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

8.313.2 Field Documentation

8.313.2.1 uint8_t NASCiotAcqOrderPrefTlv::CiotAcqOrderLen

8.313.2.2 uint32_t NASCiotAcqOrderPrefTlv::CiotAcqOrderPref[255]

8.313.2.3 uint8_t NASCiotAcqOrderPrefTlv::TlvPresent

8.314 NASCiotLteOpModePrefTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint64_t [CiotLteOpModePref](#)

8.314.1 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

8.314.2 Field Documentation

8.314.2.1 uint64_t NASCiotLteOpModePrefTlv::CiotLteOpModePref

8.314.2.2 uint8_t NASCiotLteOpModePrefTlv::TlvPresent

8.315 NASEmergencyModeTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [EmerMode](#)

8.315.1 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

8.315.2 Field Documentation

8.315.2.1 uint8_t NASEmergencyModeTlv::EmerMode

8.315.2.2 uint8_t NASEmergencyModeTlv::TlvPresent

8.316 NasGetLTECphyCalInfo Struct Reference

Data Fields

- [NASPhyCaAggScellIndType](#) [PhyCaAggScellIndType](#)
- [NASPhyCaAggScellDIBw](#) [PhyCaAggScellDIBw](#)
- [NASPhyCaAggScellInfo](#) [PhyCaAggScellInfo](#)
- [NASPhyCaAggPcellInfo](#) [PhyCaAggPcellInfo](#)
- [NASPhyCaAggScellIndex](#) [PhyCaAggScellIndex](#)
- [NASPhyCaAggScellArray](#) [PhyCaAggScellArray](#)

8.316.1 Detailed Description

This structure contains the parameters for aggregation event information.

Parameters

<i>PhyCaAggScell-IndType</i>	<ul style="list-style-type: none"> • see NASPhyCaAggScellIndType • Bit to check in ParamPresenceMask - 16
<i>PhyCaAggScell-DIBw</i>	<ul style="list-style-type: none"> • see NASPhyCaAggScellDIBw • Bit to check in ParamPresenceMask - 17
<i>PhyCaAggScell-Info</i>	<ul style="list-style-type: none"> • see NASPhyCaAggScellInfo • Bit to check in ParamPresenceMask - 18
<i>PhyCaAggPcell-Info</i>	<ul style="list-style-type: none"> • see NASPhyCaAggPcellInfo • Bit to check in ParamPresenceMask - 19

<i>PhyCaAggScell-Index</i>	<ul style="list-style-type: none"> • see NASPhyCaAggScellIndex • Bit to check in ParamPresenceMask - 20
<i>PhyCaAggScell-Array</i>	<ul style="list-style-type: none"> • see NASPhyCaAggScellArray • Bit to check in ParamPresenceMask - 21

8.316.2 Field Documentation

8.316.2.1 **NASPhyCaAggPcellInfo** NasGetLTECphyCalInfo::PhyCaAggPcellInfo

8.316.2.2 **NASPhyCaAggScellArray** NasGetLTECphyCalInfo::PhyCaAggScellArray

8.316.2.3 **NASPhyCaAggScellIDIBw** NasGetLTECphyCalInfo::PhyCaAggScellIDIBw

8.316.2.4 **NASPhyCaAggScellIndex** NasGetLTECphyCalInfo::PhyCaAggScellIndex

8.316.2.5 **NASPhyCaAggScellIndType** NasGetLTECphyCalInfo::PhyCaAggScellIndType

8.316.2.6 **NASPhyCaAggScellInfo** NasGetLTECphyCalInfo::PhyCaAggScellInfo

8.317 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [GWAcqOrderPref](#)

8.317.1 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

8.317.2 Field Documentation

8.317.2.1 **uint32_t** NASGWAcqOrderPrefTlv::GWAcqOrderPref

8.317.2.2 **uint8_t** NASGWAcqOrderPrefTlv::TlvPresent

8.318 NASLTEBandPreferenceTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint64_t [LTEBandPref](#)

8.318.1 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

8.318.2 Field Documentation

8.318.2.1 `uint64_t NASLTEBandPreferenceTlv::LTEBandPref`

8.318.2.2 `uint8_t NASLTEBandPreferenceTlv::TlvPresent`

8.319 NASLteM1BandPrefTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint64_t LteM1BandPref`

8.319.1 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

8.319.2 Field Documentation

8.319.2.1 `uint64_t NASLteM1BandPrefTlv::LteM1BandPref`

8.319.2.2 `uint8_t NASLteM1BandPrefTlv::TlvPresent`

8.320 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t nas_release`
- `uint8_t nas_major`
- `uint8_t nas_minor`

8.320.1 Detailed Description

This structure contains LTE Nas Release Information

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack result
<i>nas_release</i>	<ul style="list-style-type: none"> LTE NAS release
<i>nas_major</i>	<ul style="list-style-type: none"> LTE NAS version major
<i>nas_minor</i>	<ul style="list-style-type: none"> LTE NAS version minor

8.320.2 Field Documentation

8.320.2.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

8.320.2.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

8.320.2.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

8.320.2.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

8.321 NASLteNB1BandPrefTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint64_t [LteNB1BandPref](#)

8.321.1 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

8.321.2 Field Documentation

8.321.2.1 uint64_t NASLteNB1BandPrefTlv::LteNB1BandPref

8.321.2.2 uint8_t NASLteNB1BandPrefTlv::TlvPresent

8.322 NASModePreferenceTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint16_t [ModePref](#)

8.322.1 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

8.322.2 Field Documentation

8.322.2.1 `uint16_t NASModePreferenceTlv::ModePref`

8.322.2.2 `uint8_t NASModePreferenceTlv::TlvPresent`

8.323 NASNetSelPreferenceTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t NetSelPref`

8.323.1 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

8.323.2 Field Documentation

8.323.2.1 `uint8_t NASNetSelPreferenceTlv::NetSelPref`

8.323.2.2 `uint8_t NASNetSelPreferenceTlv::TlvPresent`

8.324 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`

8.324.1 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

8.324.2 Field Documentation

8.324.2.1 uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits1_64

8.324.2.2 uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits_129_192

8.324.2.3 uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits_193_256

8.324.2.4 uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits_65_128

8.324.2.5 uint8_t NASNr5gBandPrefTlv::TlvPresent

8.325 NASOTAMessageTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [message_type](#)
- uint16_t [data_len](#)
- uint8_t [data_buf](#) [2048]

8.325.1 Detailed Description

This structure contains OTA message

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result
<i>message_type</i>	<ul style="list-style-type: none"> • message type <ul style="list-style-type: none"> – 0 - LTE ESM uplink – 1 - LTE ESM downlink – 2 - LTE EMM uplink – 3 - LTE EMM downlink – 4 - GSM/UMTS uplink – 5 - GSM/UMTS downlink
<i>data_len</i>	<ul style="list-style-type: none"> • OTA Message Content Length
<i>data_buf</i>	<ul style="list-style-type: none"> • OTA Message Content

8.325.2 Field Documentation

8.325.2.1 `uint8_t NASOTAMessageTlv::data_buf[2048]`

8.325.2.2 `uint16_t NASOTAMessageTlv::data_len`

8.325.2.3 `uint32_t NASOTAMessageTlv::message_type`

8.325.2.4 `uint8_t NASOTAMessageTlv::TlvPresent`

8.326 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`

8.326.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Pcell Information.

Parameters

<i>pci</i>	<ul style="list-style-type: none"> • Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none"> • Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
<i>dl_bw_value</i>	<ul style="list-style-type: none"> • Downlink Bandwidth Values. • See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information.
<i>scell_state</i>	<ul style="list-style-type: none"> • SCell state Values. • See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.326.2 Field Documentation

8.326.2.1 `LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value`

8.326.2.2 `uint32_t NASPhyCaAggPcellInfo::freq`

8.326.2.3 `uint32_t NASPhyCaAggPcellInfo::iLTEbandValue`

8.326.2.4 `uint32_t NASPhyCaAggPcellInfo::pci`

8.326.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

8.327 NASPhyCaAggScellArray Struct Reference

Data Fields

- uint8_t [cphy_cell_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](#)

8.327.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Index.

Parameters

<i>cphy_cell_info_list_len</i>	<ul style="list-style-type: none"> • Number of sets of the following elements: <ul style="list-style-type: none"> – pci – freq – cphy_ca_dl_bandwidth – band – scell_state – scell_idx
<i>pci</i>	<ul style="list-style-type: none"> • Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none"> • Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
<i>cphy_ca_dl_bandwidth</i>	<ul style="list-style-type: none"> • Downlink Bandwidth Values. • See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information.
<i>band</i>	<ul style="list-style-type: none"> • Band value. • See LITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND for more information.
<i>scell_state</i>	<ul style="list-style-type: none"> • Scell state Values. • See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
<i>scell_idx</i>	<ul style="list-style-type: none"> • Scell index Values. • Range for index values: 0 to 7.

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

8.327.2 Field Documentation

8.327.2.1 uint16_t NASPhyCaAggScellArray::band[255]

8.327.2.2 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellArray::cphy_ca_dl_bandwidth[255]

8.327.2.3 uint8_t NASPhyCaAggScellArray::cphy_scell_info_list_len

8.327.2.4 uint16_t NASPhyCaAggScellArray::freq[255]

8.327.2.5 uint16_t NASPhyCaAggScellArray::pci[255]

8.327.2.6 uint16_t NASPhyCaAggScellArray::scell_idx[255]

8.327.2.7 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellArray::scell_state[255]

8.327.2.8 uint8_t NASPhyCaAggScellArray::TlvPresent

8.328 NASPhyCaAggScellIDBw Struct Reference

Data Fields

- [LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value](#)
- uint8_t [TlvPresent](#)

8.328.1 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"> • Downlink Bandwidth Values. • See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information.
--------------------	--

8.328.2 Field Documentation

8.328.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellIDBw::dl_bw_value

8.328.2.2 uint8_t NASPhyCaAggScellIDBw::TlvPresent

8.329 NASPhyCaAggScellIndex Struct Reference

Data Fields

- uint8_t [scell_idx](#)
- uint8_t [TlvPresent](#)

8.329.1 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"> Physical cell ID of the SCell Range. Range for ID values: 0 to 503.
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present.

8.329.2 Field Documentation

8.329.2.1 `uint8_t NASPhyCaAggScellIndex::scell_idx`

8.329.2.2 `uint8_t NASPhyCaAggScellIndex::TlvPresent`

8.330 NASPhyCaAggScellIndType Struct Reference

Data Fields

- `uint32_t pci`
- `uint32_t freq`
- `LITEQMI_NAS_LTE_CPHY_SCELL_STATE scell_state`
- `uint8_t TlvPresent`

8.330.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Indeicator Type.

Parameters

<i>pci</i>	<ul style="list-style-type: none"> Physical cell ID of the SCell Range. Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none"> Frequency of the absolute cell Range. Range for ID values: 0 to 65535.
<i>scell_state</i>	<ul style="list-style-type: none"> Scell state Values. See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present.

8.330.2 Field Documentation

8.330.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.330.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.330.2.3 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.330.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.331 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](#) scell_state
- uint8_t [TlvPresent](#)

8.331.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Information.

Parameters

<i>pci</i>	<ul style="list-style-type: none"> • Physical cell ID of the SCell Range. • Range for ID values: 0 to 503.
<i>freq</i>	<ul style="list-style-type: none"> • Frequency of the absolute cell Range. • Range for ID values: 0 to 65535.
<i>dl_bw_value</i>	<ul style="list-style-type: none"> • Downlink Bandwidth Values. • See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information.
<i>iLTEbandValue</i>	<ul style="list-style-type: none"> • Band value. • Range for LTE Band class 120 to 160.
<i>scell_state</i>	<ul style="list-style-type: none"> • Scell state Values. • See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.331.2 Field Documentation

8.331.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

8.331.2.2 uint32_t NASPhyCaAggScellInfo::freq

8.331.2.3 uint32_t NASPhyCaAggScellInfo::ltebandValue

8.331.2.4 uint32_t NASPhyCaAggScellInfo::pci

8.331.2.5 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

8.331.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

8.332 NASPRLPreferenceTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint16_t [PRLPref](#)

8.332.1 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

8.332.2 Field Documentation

8.332.2.1 uint16_t NASPRLPreferenceTlv::PRLPref

8.332.2.2 uint8_t NASPRLPreferenceTlv::TlvPresent

8.333 NASQmiCbkNasSwiOTAMessageInd Struct Reference

Data Fields

- [NASOTAMessageTlv](#) otaMsgTlv
- [NASLteNasReleaseInfoTlv](#) nasRelInfoTlv
- [NASTimeInfoTlv](#) timeTlv

8.333.1 Detailed Description

This structure contains OTA message

Parameters

<i>otaMsgTlv</i>	<ul style="list-style-type: none">• See NASOTAMessageTlv for more details• Bit to check in ParamPresenceMask - 1
<i>nasRelInfoTlv</i>	<ul style="list-style-type: none">• See NASLteNasReleaseInfoTlv for more details• Bit to check in ParamPresenceMask - 16

<i>timeTlv</i>	<ul style="list-style-type: none"> • See NASTimeInfoTlv for more details • Bit to check in ParamPresenceMask - 17
----------------	--

8.333.2 Field Documentation

8.333.2.1 **NASLteNasReleaseInfoTlv** NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

8.333.2.2 **NASOTAMessageTlv** NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

8.333.2.3 **NASTimeInfoTlv** NASQmiCbkNasSwiOTAMessageInd::timeTlv

8.334 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

8.334.1 Detailed Description

Structure used to store all QMI Notification parameters.

Parameters

<i>EMTlv</i>	- Emergency mode Tlv <ul style="list-style-type: none"> • See NASEmergencyModeTlv for more details • Bit to check in ParamPresenceMask - 16
<i>MPTlv</i>	- Mode preference Tlv <ul style="list-style-type: none"> • See NASModePreferenceTlv for more details • Bit to check in ParamPresenceMask - 17
<i>BPTlv</i>	- Band preference Tlv <ul style="list-style-type: none"> • See NASBandPreferenceTlv for more details • Bit to check in ParamPresenceMask - 18

<i>PRLPTlv</i>	- PRL preference Tlv <ul style="list-style-type: none"> • See NASPRLPreferenceTlv for more details • Bit to check in ParamPresenceMask - 19
<i>RPTlv</i>	- Roaming preference Tlv <ul style="list-style-type: none"> • See NASRoamPreferenceTlv for more details • Bit to check in ParamPresenceMask - 20
<i>LBPTlv</i>	- LTE band preference Tlv <ul style="list-style-type: none"> • See NASLTEBandPreferenceTlv for more details • Bit to check in ParamPresenceMask - 21
<i>NSPTlv</i>	- Network selection preference Tlv <ul style="list-style-type: none"> • See NASNetSelPreferenceTlv for more details • Bit to check in ParamPresenceMask - 22
<i>SDPTlv</i>	- Service domain preference Tlv <ul style="list-style-type: none"> • See NASServDomainPrefTlv for more details • Bit to check in ParamPresenceMask - 24
<i>GWAOPTlv</i>	- GSM WCDMA Acquisition Order Preference Tlv <ul style="list-style-type: none"> • See NASGWAcqOrderPrefTlv for more details • Bit to check in ParamPresenceMask - 25
<i>AOPTlv</i>	- Acquisition Order Preference Tlv <ul style="list-style-type: none"> • See NASAcqOrderPrefTlv for more details • Bit to check in ParamPresenceMask - 28
<i>RatDMTlv</i>	- Disabled RAT Bitmask Tlv <ul style="list-style-type: none"> • See NASRatDisabledMaskTlv for more details • Bit to check in ParamPresenceMask - 34
<i>CiotOpMPTlv</i>	- CIOT LTE Operational Mode Preference Tlv <ul style="list-style-type: none"> • See NASCiotLteOpModePrefTlv for more details • Bit to check in ParamPresenceMask - 36
<i>M1BandPTlv</i>	- LTE M1 Band Preference Tlv <ul style="list-style-type: none"> • See NASLteM1BandPrefTlv for more details • Bit to check in ParamPresenceMask - 37
<i>NB1BandPTlv</i>	- LTE NB1 Band Preference Tlv <ul style="list-style-type: none"> • See NASLteNB1BandPrefTlv for more details • Bit to check in ParamPresenceMask - 38
<i>CiotAOPTlv</i>	- CIOT acquisition order preference Tlv <ul style="list-style-type: none"> • See NASCiotAcqOrderPrefTlv for more details • Bit to check in ParamPresenceMask - 39
<i>NR5gBandPTlv</i>	- NR5G Band Preference Tlv <ul style="list-style-type: none"> • See NASNr5gBandPrefTlv for more details • Bit to check in ParamPresenceMask - 40

Note: None

8.334.2 Field Documentation

- 8.334.2.1 **NASAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::AOPTlv
- 8.334.2.2 **NASBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::BPTlv
- 8.334.2.3 **NASCiotAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::CiotAOPTlv
- 8.334.2.4 **NASCiotLteOpModePrefTlv** NASQmiCbkNasSystemSelPrefInd::CiotOpMPTlv
- 8.334.2.5 **NASEmergencyModeTlv** NASQmiCbkNasSystemSelPrefInd::EMTlv
- 8.334.2.6 **NASGWAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::GWAOPTlv
- 8.334.2.7 **NASLTEBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::LBPTlv
- 8.334.2.8 **NASLteM1BandPrefTlv** NASQmiCbkNasSystemSelPrefInd::M1BandPTlv
- 8.334.2.9 **NASModePreferenceTlv** NASQmiCbkNasSystemSelPrefInd::MPTlv
- 8.334.2.10 **NASLteNB1BandPrefTlv** NASQmiCbkNasSystemSelPrefInd::NB1BandPTlv
- 8.334.2.11 **NASNr5gBandPrefTlv** NASQmiCbkNasSystemSelPrefInd::NR5gBandPTlv
- 8.334.2.12 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv
- 8.334.2.13 **NASPRPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv
- 8.334.2.14 **NASRatDisabledMaskTlv** NASQmiCbkNasSystemSelPrefInd::RatDMTlv
- 8.334.2.15 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv
- 8.334.2.16 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.335 NASRatDisabledMaskTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint16_t [RatDisabledMask](#)

8.335.1 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>RatDisabledMask</i>	- Bitmask representing the radio technologies that are disabled

8.335.2 Field Documentation

8.335.2.1 uint16_t NASRatDisabledMaskTlv::RatDisabledMask

8.335.2.2 uint8_t NASRatDisabledMaskTlv::TlvPresent

8.336 NASRoamPreferenceTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint16_t [RoamPref](#)

8.336.1 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

8.336.2 Field Documentation

8.336.2.1 uint16_t NASRoamPreferenceTlv::RoamPref

8.336.2.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.337 NASServDomainPrefTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [SrvDomainPref](#)

8.337.1 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

8.337.2 Field Documentation

8.337.2.1 uint32_t NASServDomainPrefTlv::SrvDomainPref

8.337.2.2 uint8_t NASServDomainPrefTlv::TlvPresent

8.338 NAServingSystemInfo 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](#)

8.338.1 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"> • 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service • 1 - QMI_NAS_REGISTERED Registered with a network • 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service • 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network • 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown
<i>csAttachState</i>	- Circuit Switch domain attach state of the mobile <ul style="list-style-type: none"> • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
<i>psAttachState</i>	- Packet domain attach state of the mobile <ul style="list-style-type: none"> • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
<i>selectedNetwork</i>	- Type of selected radio access network <ul style="list-style-type: none"> • 0x00 - Unknown • 0x01 - 3GPP2 network • 0x02 - 3GPP network
<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"> • 0x00 - None (no service) • 0x01 - cdma2000 1X • 0x02 - cdma2000 HRPD (1xEV-DO) • 0x03 - AMPS • 0x04 - GSM • 0x05 - UMTS • 0x08 - LTE

<i>hdrPersonality</i>	<ul style="list-style-type: none">- HDR personality information (valid only for EVDO)• 0x00 - Unknown• 0x01 - HRPD• 0x02 - eHRPD
-----------------------	---

Note: None

8.338.2 Field Documentation

8.338.2.1 `uint8_t NAServingSystemInfo::csAttachState`

8.338.2.2 `uint8_t NAServingSystemInfo::hdrPersonality`

8.338.2.3 `uint8_t NAServingSystemInfo::psAttachState`

8.338.2.4 `uint8_t NAServingSystemInfo::radioInterfaceList[255]`

8.338.2.5 `uint8_t NAServingSystemInfo::radioInterfaceNo`

8.338.2.6 `uint8_t NAServingSystemInfo::registrationState`

8.338.2.7 `uint8_t NAServingSystemInfo::selectedNetwork`

8.339 NASTimeInfoTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint64_t time`

8.339.1 Detailed Description

This structure contains LTE Nas time Infomation

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack result
<i>time</i>	<ul style="list-style-type: none">• Seconds in local time since Jan. 6th 1980 00:00:00 UTC

8.339.2 Field Documentation

8.339.2.1 `uint64_t NASTimeInfoTlv::time`

8.339.2.2 `uint8_t NASTimeInfoTlv::TlvPresent`

8.340 newMTMessageTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [sMSMTMessageInfo](#) [MTMessageInfo](#)

8.340.1 Detailed Description

This structure contains MT message TLV information.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Boolean indicating the presence of the TLV in the QMI response
<i>MTMessageInfo</i>	<ul style="list-style-type: none"> • MT Message • See sMSMTMessageInfo for more information

8.340.2 Field Documentation

8.340.2.1 [sMSMTMessageInfo](#) newMTMessageTlv::MTMessageInfo

8.340.2.2 uint8_t newMTMessageTlv::TlvPresent

8.341 pack_audio_SLQSGetAudioPathConfig_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [Item](#)

8.341.1 Detailed Description

This structure contains the pack parameters for Get Audio Path Config.

Parameters

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

8.341.2 Field Documentation

8.341.2.1 uint8_t pack_audio_SLQSGetAudioPathConfig_t::Item

8.341.2.2 uint8_t pack_audio_SLQSGetAudioPathConfig_t::Profile

8.342 pack_audio_SLQSGetAudioProfile_t Struct Reference

Data Fields

- uint8_t [Generator](#)

8.342.1 Detailed Description

This structure contains the pack parameters to Get Audio Profile.

Parameters

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

8.342.2 Field Documentation

8.342.2.1 uint8_t pack_audio_SLQSGetAudioProfile_t::Generator

8.343 pack_audio_SLQSGetAudioVoTLBConfig_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [Generator](#)
- uint8_t [Volume](#)
- uint8_t [Item](#)

8.343.1 Detailed Description

This structure contains the pack parameters for SLQSGetAudioVoTLBConfig.

Parameters

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

8.343.2 Field Documentation

8.343.2.1 uint8_t pack_audio_SLQSGetAudioVoTLBConfig_t::Generator

8.343.2.2 uint8_t pack_audio_SLQSGetAudioVoTLBConfig_t::Item

8.343.2.3 uint8_t pack_audio_SLQSGetAudioVoTLBConfig_t::Profile

8.343.2.4 uint8_t pack_audio_SLQSGetAudioVoTLBConfig_t::Volume

8.344 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

8.344.1 Detailed Description

This structure contains the pack parameters for SLQSSetAudioPathConfig .

Parameters

<i>Profile</i>	[Mandatory] <ul style="list-style-type: none"> • Audio Profile <ul style="list-style-type: none"> – 0-9
<i>pECMode</i>	[Optional] <ul style="list-style-type: none"> • AV_EC <ul style="list-style-type: none"> – 0 - Echo cancellation off – 1 - Handset echo mode – 2 - Headset mode – 3 - Car kit mode – 4 - Speaker Mode
<i>pNSEnable</i>	[Optional] <ul style="list-style-type: none"> • Noise Suppression <ul style="list-style-type: none"> – 0 - Noise suppression off – 1 - Noise suppression on
<i>pTXGain</i>	[Optional] <ul style="list-style-type: none"> • TX Voice volume <ul style="list-style-type: none"> – 0x0000 - 0xffff
<i>pDTMFTXGain</i>	[Optional] <ul style="list-style-type: none"> • AV_DTMFTXG <ul style="list-style-type: none"> – 0x0000 - 0xffff
<i>pCodecSTGain</i>	[Optional] <ul style="list-style-type: none"> • AV_CODECASTG <ul style="list-style-type: none"> – 0x0000 - 0xffff
<i>pTXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> • See audio_TXPCMIIRFiltr for more information
<i>pRXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> • See audio_RXPCMIIRFiltr for more information
<i>pRXAVCAGC-Switch</i>	[Optional] <ul style="list-style-type: none"> • RX AVC/AGC Switch

<i>pTXAVCSwitch</i>	[Optional] • TX AVC Switch
<i>pRXAGCList</i>	[Optional] • See audio_RXAGCList for more information
<i>pRXAVCList</i>	[Optional] • See audio_RXAVCList for more information
<i>pTXAGCList</i>	[Optional] • See audio_TXAGCList for more information

8.344.2 Field Documentation

8.344.2.1 `uint16_t*` `pack_audio_SLQSSetAudioPathConfig_t::pCodecSTGain`

8.344.2.2 `uint16_t*` `pack_audio_SLQSSetAudioPathConfig_t::pDTMFTXGain`

8.344.2.3 `uint8_t*` `pack_audio_SLQSSetAudioPathConfig_t::pECMode`

8.344.2.4 `uint8_t*` `pack_audio_SLQSSetAudioPathConfig_t::pNSEnable`

8.344.2.5 `uint8_t` `pack_audio_SLQSSetAudioPathConfig_t::Profile`

8.344.2.6 `audio_RXAGCList*` `pack_audio_SLQSSetAudioPathConfig_t::pRXAGCList`

8.344.2.7 `uint8_t*` `pack_audio_SLQSSetAudioPathConfig_t::pRXAVCAGCSwitch`

8.344.2.8 `audio_RXAVCList*` `pack_audio_SLQSSetAudioPathConfig_t::pRXAVCList`

8.344.2.9 `audio_RXPCMIIRFitr*` `pack_audio_SLQSSetAudioPathConfig_t::pRXPCMIIRFitr`

8.344.2.10 `audio_TXAGCList*` `pack_audio_SLQSSetAudioPathConfig_t::pTXAGCList`

8.344.2.11 `uint8_t*` `pack_audio_SLQSSetAudioPathConfig_t::pTXAVCSwitch`

8.344.2.12 `uint16_t*` `pack_audio_SLQSSetAudioPathConfig_t::pTXGain`

8.344.2.13 `audio_TXPCMIIRFitr*` `pack_audio_SLQSSetAudioPathConfig_t::pTXPCMIIRFitr`

8.345 `pack_audio_SLQSSetAudioProfile_t` Struct Reference

Data Fields

- `uint8_t` [Profile](#)
- `uint8_t` [EarMute](#)
- `uint8_t` [MicMute](#)
- `uint8_t` [Generator](#)
- `uint8_t` [Volume](#)

8.345.1 Detailed Description

This structure contains pack parameters to Set Audio Profile.

Parameters

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

8.345.2 Field Documentation

8.345.2.1 uint8_t pack_audio_SLQSSetAudioProfile_t::EarMute

8.345.2.2 uint8_t pack_audio_SLQSSetAudioProfile_t::Generator

8.345.2.3 uint8_t pack_audio_SLQSSetAudioProfile_t::MicMute

8.345.2.4 uint8_t pack_audio_SLQSSetAudioProfile_t::Profile

8.345.2.5 uint8_t pack_audio_SLQSSetAudioProfile_t::Volume

8.346 pack_audio_SLQSSetAudioVoITLBConfig_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [Generator](#)
- uint8_t [Volume](#)
- uint8_t [Item](#)
- uint16_t [VolValue](#)

8.346.1 Detailed Description

This structure contains the pack parameters for SLQSSetAudioVoITLBConfig.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile <ul style="list-style-type: none"> – 0-9
<i>Generator</i>	<ul style="list-style-type: none"> • Audio Generator <ul style="list-style-type: none"> – 0-2
<i>Volume</i>	<ul style="list-style-type: none"> • Audio Volume Level <ul style="list-style-type: none"> – 0-7
<i>Item</i>	<ul style="list-style-type: none"> • Item <ul style="list-style-type: none"> – 13 - AV_RXVOLDB – 14 - AV_DTMFVOLDB – 15 - AV_PAD
<i>Value</i>	<ul style="list-style-type: none"> • Value to be set to the volume table

8.346.2 Field Documentation

8.346.2.1 uint8_t pack_audio_SLQSSetAudioVoITLBConfig_t::Generator

8.346.2.2 uint8_t pack_audio_SLQSSetAudioVoITLBConfig_t::Item

8.346.2.3 uint8_t pack_audio_SLQSSetAudioVoITLBConfig_t::Profile

8.346.2.4 uint8_t pack_audio_SLQSSetAudioVoITLBConfig_t::Volume

8.346.2.5 uint16_t pack_audio_SLQSSetAudioVoITLBConfig_t::VolValue

8.347 pack_cat_CATSendEnvelopeCommand_t Struct Reference

Data Fields

- uint32_t [cmdID](#)
- uint32_t [dataLen](#)
- uint8_t * [pData](#)

8.347.1 Detailed Description

Structure used to store envelope command data.

Parameters

<i>cmdID</i>	<ul style="list-style-type: none">• Envelope command type<ul style="list-style-type: none">– 0x01 - Menu Selection– 0x02 - Event DL User activity– 0x03 - Event DL Idle Screen Available– 0x04 - Event DL Language Selection
<i>dataLen</i>	<ul style="list-style-type: none">• Length of pData in bytes
<i>pData</i>	<ul style="list-style-type: none">• Encoded envelope data as defined in ETSI TS 102 223, section 7 [Smart Cards: Card Application Toolkit (CAT) - Release 4]

8.347.2 Field Documentation

8.347.2.1 uint32_t pack_cat_CATSendEnvelopeCommand_t::cmdID

8.347.2.2 uint32_t pack_cat_CATSendEnvelopeCommand_t::dataLen

8.347.2.3 uint8_t* pack_cat_CATSendEnvelopeCommand_t::pData

8.348 pack_cat_CATSendTerminalResponse_t Struct Reference

Data Fields

- uint32_t [refID](#)
- uint32_t [dataLen](#)
- uint8_t * [pData](#)

8.348.1 Detailed Description

Structure used to terminal response data.

Parameters

<i>refID</i>	<ul style="list-style-type: none">• Proactive command reference ID. The value should be the same as indicated in the CAT event callback data for the relevant proactive command.
--------------	--

<i>dataLen</i>	<ul style="list-style-type: none"> Terminal response data length
<i>pData</i>	<ul style="list-style-type: none"> 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]

8.348.2 Field Documentation

8.348.2.1 `uint32_t pack_cat_CATSendTerminalResponse_t::dataLen`

8.348.2.2 `uint8_t* pack_cat_CATSendTerminalResponse_t::pData`

8.348.2.3 `uint32_t pack_cat_CATSendTerminalResponse_t::refID`

8.349 `pack_cat_SetCATEventCallback_t` Struct Reference

Data Fields

- `uint32_t eventMask`

8.349.1 Detailed Description

This structure contains sEnables/disables the CAT event callback pack variable.

Parameters

<i>eventMask</i>	<ul style="list-style-type: none"> bitmask of CAT events to register for <ul style="list-style-type: none"> 0x00000001 - Display Text 0x00000002 - Get In-Key 0x00000004 - Get Input 0x00000008 - Setup Menu 0x00000010 - Select Item 0x00000020 - Send SMS - Alpha Identifier 0x00000040 - Setup Event: User Activity 0x00000080 - Setup Event: Idle Screen Notify 0x00000100 - Setup Event: Language Sel Notify 0x00000200 - Setup Idle Mode Text 0x00000400 - Language Notification 0x00000800 - Refresh 0x00001000 - End Proactive Session
------------------	---

8.349.2 Field Documentation

8.349.2.1 `uint32_t pack_cat_SetCATEventCallback_t::eventMask`

8.350 pack_dms_ActivateAutomatic_t Struct Reference

Data Fields

- uint8_t [actCode](#) [81]

8.350.1 Detailed Description

This structure contains UIM activate automation information

Parameters

<i>actCode</i>	<ul style="list-style-type: none">• NULL-terminated string representing activation code (maximum string length of 12).• Specific carrier requirements may dictate actual activation code that is applicable, e.g., "*22899"
----------------	--

8.350.2 Field Documentation

8.350.2.1 uint8_t pack_dms_ActivateAutomatic_t::actCode[81]

8.351 pack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- uint8_t [cust_id](#) [64+1]
- uint8_t [list_type](#)
- uint16_t [Tlvresult](#)

8.351.1 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">• Customization ID (Maximum 64 bytes)• NULL terminated ASCII string.
<i>list_type</i>	<ul style="list-style-type: none">• list type requested
<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack Result

8.351.2 Field Documentation

8.351.2.1 uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]

8.351.2.2 uint8_t pack_dms_GetCustFeaturesV2_t::list_type

8.351.2.3 uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult

8.352 pack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- uint8_t [spc](#) [6]

8.352.1 Detailed Description

This structure contains UIM reset to factory default information

Parameters

<i>spc</i> [IN]	<ul style="list-style-type: none"> • Service programming code in ASCII format (digits 0 to 9 only)
-----------------	---

8.352.2 Field Documentation

8.352.2.1 uint8_t pack_dms_ResetToFactoryDefaults_t::spc[6]

8.353 pack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- uint8_t [activationState](#)

8.353.1 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"> • Service activation state. • Values <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report activation state changes
------------------------	---

8.353.2 Field Documentation

8.353.2.1 uint8_t pack_dms_SetActivationStatusCallback_t::activationState

8.354 pack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t [crashAction](#)

8.354.1 Detailed Description

This structure is used to store pack_dms_SetCrashAction parameters.

Parameters

<i>crashAction</i>	<ul style="list-style-type: none"> • 0 - USB Memory Download. Modem will reset after a crash and will stay in USB download mode with only DM port enumerated. • 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 • 2 - No Action
--------------------	---

8.354.2 Field Documentation

8.354.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

8.355 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](#)

8.355.1 Detailed Description

This structure contains current settings of custom features pack parameters

Parameters

<i>GpsEnable</i>	<ul style="list-style-type: none"> • describes if GPS is enabled or disabled • values: <ul style="list-style-type: none"> – 0x00 - GPS is disabled – 0x01 - GPS is enabled
<i>DisableIMSI</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: <ul style="list-style-type: none"> – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI

<i>IPFamSupport</i>	<ul style="list-style-type: none"> • optional 2 byte BitMask • bitmask representing the IP families supported • values: <ul style="list-style-type: none"> – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6
<i>RMAutoConnect</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: <ul style="list-style-type: none"> – 0x00 - Not Supported – 0x01 - Supported
<i>GPSSel</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • GPS Antenna Select • values: <ul style="list-style-type: none"> – 0x00 - Dedicated GPS Port – 0x01 - GPS Rx over AUX Port – 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied
<i>SMSSupport</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • SMS support • values: <ul style="list-style-type: none"> – 0x00 - Not supported – 0x01 - supported • Used to determine whether or not to hide SMS from user
<i>IsVoiceEnabled</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • Voice support • values: <ul style="list-style-type: none"> – 0x00 - Enable voice on both AT and QMI interface (default) – 0x01 - Reserved – 0x02 - Disable voice on both AT and QMI interface
<i>DHCPRelay-Enabled</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • DHCP Relay support • values: <ul style="list-style-type: none"> – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay

<i>GPSLPM</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • GPSLPM support • values: <ul style="list-style-type: none"> – 0x00 - Enable GPS in Low Power Mode – 0x01 - Disable GPS in Low Power Mode
---------------	---

8.355.2 Field Documentation

8.355.2.1 `uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled`

8.355.2.2 `uint8_t pack_dms_SetCustFeature_t::DisableIMSI`

8.355.2.3 `uint32_t pack_dms_SetCustFeature_t::GpsEnable`

8.355.2.4 `uint8_t pack_dms_SetCustFeature_t::GPSLPM`

8.355.2.5 `uint8_t pack_dms_SetCustFeature_t::GPSSel`

8.355.2.6 `uint16_t pack_dms_SetCustFeature_t::IPFamSupport`

8.355.2.7 `uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled`

8.355.2.8 `uint8_t pack_dms_SetCustFeature_t::RMAutoConnect`

8.355.2.9 `uint8_t pack_dms_SetCustFeature_t::SMSSupport`

8.356 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`

8.356.1 Detailed Description

This structure contains customization settings set to modem pack

Parameters

<i>cust_id</i>	<ul style="list-style-type: none"> • Customization ID (Maximum 64 bytes) • NULL terminated ASCII string.
<i>value_length</i>	<ul style="list-style-type: none"> • length of cust_value field
<i>cust_value</i>	<ul style="list-style-type: none"> • Customization Setting Value (Maximum 8 bytes)

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Pack Result
------------------	---

8.356.2 Field Documentation

8.356.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.356.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.356.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.356.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.357 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t [mode](#)

8.357.1 Detailed Description

This structure is used to store pack_dms_SetEventReport parameters.

Parameters

<i>mode</i>	<ul style="list-style-type: none"> • Operating Mode • 0 - reset • 1 - set • 2 - no change
-------------	---

8.357.2 Field Documentation

8.357.2.1 uint8_t pack_dms_SetEventReport_t::mode

8.358 pack_dms_SetIndicationRegister_t Struct Reference

Data Fields

- uint8_t [PSMStatus](#)
- uint8_t [PSMCfgChangeInfo](#)
- uint8_t [RptIMSCapability](#)

8.358.1 Detailed Description

This structure contains the DMS Register Indication request parameters

Parameters

<i>PSMStatus</i>	<ul style="list-style-type: none"> • Power Save Mode Status • Values <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report PSM status – 2 - No Change
<i>PSMCfgChange-Info</i>	<ul style="list-style-type: none"> • Power Save Mode Configuration Change Information • Values <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report Power Save Mode configuration changes – 2 - No Change
<i>RptIMS-Capability</i>	<ul style="list-style-type: none"> • Report IMS Capability • Values <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report IMS capability – 2 - No Change

8.358.2 Field Documentation

8.358.2.1 uint8_t pack_dms_SetIndicationRegister_t::PSMCfgChangeInfo

8.358.2.2 uint8_t pack_dms_SetIndicationRegister_t::PSMStatus

8.358.2.3 uint8_t pack_dms_SetIndicationRegister_t::RptIMSCapability

8.359 pack_dms_SetPower_t Struct Reference

Data Fields

- uint32_t [mode](#)
- uint16_t [Tlvresult](#)

8.359.1 Detailed Description

Structure to store pack the operating mode of the device.

Parameters

<i>mode</i>	<ul style="list-style-type: none"> • Selected operating mode <ul style="list-style-type: none"> – See qaGobiApiTablePowerModes.h for power modes
<i>Tlvresult</i>	<ul style="list-style-type: none"> • pack Tlv Result.

8.359.2 Field Documentation

8.359.2.1 uint32_t pack_dms_SetPower_t::mode

8.359.2.2 uint16_t pack_dms_SetPower_t::Tlvresult

8.360 pack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint8_t [USBComp](#)
- uint16_t [Tlvresult](#)

8.360.1 Detailed Description

This structure is used to store USB composition information pack paramters

Parameters

<i>USBComp</i>	<ul style="list-style-type: none"> • Current USB Composition • Values: <ul style="list-style-type: none"> – 0..5 - Reserved (non-QMI) – 6 - DM NMEA AT QMI – 7 - DM NMEA AT QMI1 QMI2 QMI3 – 8 - DM NMEA AT MBIM – 9 - MBIM – 10 - NMEA MBIM – 11 - DM MBIM – 12 - DM NMEA MBIM – 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces – 13 - 6 for QMI, 8 for MBIM – 14 - 6 for QMI, 9 for MBIM – 15 - 6 for QMI, 10 for MBIM – 16 - 6 for QMI, 11 for MBIM – 17 - 6 for QMI, 12 for MBIM – 18 - 7 for QMI, 8 for MBIM – 19 - 7 for QMI, 9 for MBIM – 20 - 7 for QMI, 10 for MBIM – 21 - 7 for QMI, 11 for MBIM – 22 - 7 for QMI, 12 for MBIM
<i>Tlvresult</i>	<ul style="list-style-type: none"> • pack Tlv Result.

8.360.2 Field Documentation

8.360.2.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

8.360.2.2 uint8_t pack_dms_SetUSBComp_t::USBComp

8.361 pack_dms_SLQSDmsSwilIndicationRegister_t Struct Reference

Data Fields

- uint8_t [resetInfoInd](#)

8.361.1 Detailed Description

This structure contains set registration state for different indication pack

Parameters

<i>resetInfoInd</i> [IN]	<ul style="list-style-type: none">Values<ul style="list-style-type: none">0 - Disable1 - Enable
--------------------------	--

8.361.2 Field Documentation

8.361.2.1 uint8_t pack_dms_SLQSDmsSwilIndicationRegister_t::resetInfoInd

8.362 pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference

Data Fields

- uint8_t * [pPsmEnableState](#)
- uint32_t * [pActiveTimer](#)
- uint32_t * [pPeriodicUpdateTimer](#)

8.362.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

Parameters

<i>pPsmEnableState</i>	[Optional] <ul style="list-style-type: none">PSM Enable StateValues<ul style="list-style-type: none">0 - PSM is not enabled1 - PSM is enabled
<i>pActiveTimer</i>	[Optional] <ul style="list-style-type: none">Active TimerPSM active timer value (in seconds).ActiveTimer encoding is the GPRS timer 2 encoding specified in Table 10.5.172 of 3GPP TS 24.008.

<i>pPeriodic-UpdateTimer</i>	[Optional] <ul style="list-style-type: none"> • Periodic Update Timer • PSM periodic update timer value (in seconds) • Periodic Update Timer encoding is the GPRS timer 3 encoding specified in Table 10.5.163a of 3GPP TS 24.008.
------------------------------	---

8.362.2 Field Documentation

8.362.2.1 uint32_t* pack_dms_SLQSSetPowerSaveModeConfig_t::pActiveTimer

8.362.2.2 uint32_t* pack_dms_SLQSSetPowerSaveModeConfig_t::pPeriodicUpdateTimer

8.362.2.3 uint8_t* pack_dms_SLQSSetPowerSaveModeConfig_t::pPsmEnableState

8.363 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference

Data Fields

- uint8_t [clear](#)

8.363.1 Detailed Description

This structure contains pack_dms_SLQSSwiGetCrashInfo parameters

Parameters

<i>clear</i>	<ul style="list-style-type: none"> • 0 - do not clear crash data after response • 1 - clear crash data after response
--------------	---

8.363.2 Field Documentation

8.363.2.1 uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear

8.364 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint8_t * [pDestSMSNum](#)
- uint8_t * [pDestSMSContent](#)

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

8.364.2 Field Documentation

8.364.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.364.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.365 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

Data Fields

- char [manString](#) [255]
- char [modelString](#) [255]
- char [swVerString](#) [255]
- char [plasmaIDString](#) [255]
- char [hostID](#) [255]

8.365.1 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"> • Host Device Manufacturer Name • Null terminated ASCII String
<i>modelString</i> [<i>IN</i>]	<ul style="list-style-type: none"> • Host Device Model String • Null terminated ASCII string.
<i>swVerString</i> [<i>IN</i>]	<ul style="list-style-type: none"> • Host Device Software Version String • Null terminated ASCII string
<i>plasmaIDString</i> [<i>IN</i>]	<ul style="list-style-type: none"> • Host Device Plasma ID String • Null terminated alphanumeric ASCII string
<i>HostID</i> [<i>IN</i>]	<ul style="list-style-type: none"> • Device Host ID String • Null terminated alphanumeric ASCII string

8.365.2 Field Documentation

8.365.2.1 char pack_dms_SLQSSwiSetHostDevInfo_t::hostID[255]

8.365.2.2 char pack_dms_SLQSSwiSetHostDevInfo_t::manString[255]

8.365.2.3 char pack_dms_SLQSSwiSetHostDevInfo_t::modelString[255]

8.365.2.4 char pack_dms_SLQSSwiSetHostDevInfo_t::plasmaIDString[255]

8.365.2.5 char pack_dms_SLQSSwiSetHostDevInfo_t::swVerString[255]

8.366 pack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- char [nameString](#) [255]
- char [versionString](#) [255]

8.366.1 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"> • NULL terminated ASCII string.
<i>VersionString</i> [IN]	Host device model String <ul style="list-style-type: none"> • NULL terminated ASCII string.

8.366.2 Field Documentation

8.366.2.1 char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]

8.366.2.2 char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]

8.367 pack_dms_SwiSetEventReport_t Struct Reference

Data Fields

- uint8_t * [pTempReport](#)
- uint8_t * [pVoltReport](#)
- uint8_t * [pUIMStatusReport](#)

8.367.1 Detailed Description

This structure is used to store pack_dms_SwiSetEventReport parameters.

Parameters

<i>pTempReport</i>	(optional) <ul style="list-style-type: none"> Temperature indicator. <ul style="list-style-type: none"> 0 - Do not report 1 - Report when state changes
<i>pVoltReport</i>	(optional) <ul style="list-style-type: none"> Voltage Indicator. <ul style="list-style-type: none"> 0 - Do not report 1 - Report when state changes
<i>pUIMStatus-Report</i>	(optional) <ul style="list-style-type: none"> UIM status Indicator. <ul style="list-style-type: none"> 0 - Do not report 1 - Report when state changes

8.367.2 Field Documentation

8.367.2.1 uint8_t* pack_dms_SwiSetEventReport_t::pTempReport

8.367.2.2 uint8_t* pack_dms_SwiSetEventReport_t::pUIMStatusReport

8.367.2.3 uint8_t* pack_dms_SwiSetEventReport_t::pVoltReport

8.368 pack_dms_SwiUimSelect_t Struct Reference

Data Fields

- uint8_t [uim_select](#)

8.368.1 Detailed Description

This structure is used to store pack_dms_SwiUimSelect parameters.

Parameters

<i>uim_select</i>	<ul style="list-style-type: none"> Selection of UIM Card <ul style="list-style-type: none"> 0 - External UIM Interface 1 - Embedded UIM 2 - Remote UIM 3 - Auto switch
-------------------	--

Note

Auto switch is applicable only if UIMAUTOSWITCH feature is enabled. This feature will activate the auto-switch mechanism if it is inactive.

8.368.2 Field Documentation

8.368.2.1 `uint8_t pack_dms_SwiUimSelect_t::uim_select`

8.369 `pack_dms_UIMChangePIN_t` Struct Reference

Data Fields

- `uint8_t id`
- `uint8_t oldValue` [255]
- `uint8_t newValue` [255]

8.369.1 Detailed Description

This structure contains UIM Unblock PIN Information

Parameters

<i>id</i> [IN]	<ul style="list-style-type: none">• PIN ID<ul style="list-style-type: none">– 1 (PIN1 / CHV1)– 2 (PIN2 / CHV2)
<i>oldValue</i> [IN]	<ul style="list-style-type: none">• Old PIN value of PIN to change
<i>newValue</i> [IN]	<ul style="list-style-type: none">• New PIN value of PIN to change

8.369.2 Field Documentation

8.369.2.1 `uint8_t pack_dms_UIMChangePIN_t::id`

8.369.2.2 `uint8_t pack_dms_UIMChangePIN_t::newValue`[255]

8.369.2.3 `uint8_t pack_dms_UIMChangePIN_t::oldValue`[255]

8.370 `pack_dms_UIMGetControlKeyStatus_t` Struct Reference

Data Fields

- `uint8_t facility`

8.370.1 Detailed Description

This structure contains UIM get control key status information

Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none">• MT or network facility<ul style="list-style-type: none">– 0 - Network Personalization (PN)– 1 - Network Subset Personalization (PU)– 2 - Service Provider Personalization (PP)– 3 - Corporate Personalization (PC)– 4 - UIM Personalization (PF)
----------------------	--

8.370.2 Field Documentation

8.370.2.1 uint8_t pack_dms_UIMGetControlKeyStatus_t::facility

8.371 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.371.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result.
------------------	--

8.371.2 Field Documentation

8.371.2.1 [swi_uint256_t](#) pack_dms_UIMGetICCID_t::ParamPresenceMask

8.371.2.2 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

8.372 pack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- uint8_t [facility](#)
- uint8_t [facilityState](#)
- uint8_t [facilityCk](#) [8]

8.372.1 Detailed Description

This structure contains UIM Set control key protection information

Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> • MT or network facility <ul style="list-style-type: none"> – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
<i>facilityState</i> [IN]	<ul style="list-style-type: none"> • UIM facility state <ul style="list-style-type: none"> – 0 - Deactivated – 1 - Activated – 2 - Blocked
<i>facilityCk</i> [IN]	<ul style="list-style-type: none"> • Facility depersonalization control key, string in ASCII text (maximum 8 bytes)

8.372.2 Field Documentation

8.372.2.1 uint8_t pack_dms_UIMSetControlKeyProtection_t::facility

8.372.2.2 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]

8.372.2.3 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState

8.373 pack_dms_UIMSetPINProtection_t Struct Reference

Data Fields

- uint8_t [id](#)
- uint8_t [bEnable](#)
- uint8_t [value](#) [255]

8.373.1 Detailed Description

This structure contains PIN Protection Information

Parameters

<i>id</i> [IN]	<ul style="list-style-type: none"> • PIN ID <ul style="list-style-type: none"> – 1 (PIN1 / CHV1) – 2 (PIN2 / CHV2)
<i>bEnable</i> [IN]	<ul style="list-style-type: none"> • Enable/disable PIN protection, 0 = Disable
<i>pValue</i> [IN]	<ul style="list-style-type: none"> • PIN value of the PIN to be enabled/disabled

8.373.2 Field Documentation

8.373.2.1 uint8_t pack_dms_UIMSetPINProtection_t::bEnable

8.373.2.2 uint8_t pack_dms_UIMSetPINProtection_t::id

8.373.2.3 uint8_t pack_dms_UIMSetPINProtection_t::value[255]

8.374 pack_dms_UIMUnblockControlKey_t Struct Reference

Data Fields

- uint8_t [facility](#)
- uint8_t [facilityCk](#) [8]

8.374.1 Detailed Description

This structure contains UIM unblock Control Key information

Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> • MT or network facility <ul style="list-style-type: none"> – 0 - Network Personalization (PN) – 1 - Network Subset Personalization (PU) – 2 - Service Provider Personalization (PP) – 3 - Corporate Personalization (PC) – 4 - UIM Personalization (PF)
<i>facilityCk</i> [IN]	<ul style="list-style-type: none"> • Facility depersonalization control key, string in ASCII text (maximum 8 bytes)

8.374.2 Field Documentation

8.374.2.1 uint8_t pack_dms_UIMUnblockControlKey_t::facility

8.374.2.2 uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]

8.375 pack_dms_UIMUnblockPIN_t Struct Reference

Data Fields

- uint8_t [id](#)
- uint8_t [pukValue](#) [255]
- uint8_t [newPin](#) [255]

8.375.1 Detailed Description

This structure contains UIM Unblock PIN Information

Parameters

<i>id</i> [IN]	<ul style="list-style-type: none"> PIN ID <ul style="list-style-type: none"> 1 (PIN1 / CHV1) 2 (PIN2 / CHV2)
<i>pukValue</i> [IN]	<ul style="list-style-type: none"> PUK value of PIN to be unblocked
<i>newPin</i> [IN]	<ul style="list-style-type: none"> New PIN value for the PIN to be unblocked

8.375.2 Field Documentation

8.375.2.1 uint8_t pack_dms_UIMUnblockPIN_t::id

8.375.2.2 uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]

8.375.2.3 uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]

8.376 pack_dms_UIMVerifyPIN_t Struct Reference

Data Fields

- uint8_t [id](#)
- uint8_t [value](#) [255]

8.376.1 Detailed Description

This structure contains PIN Value Information

Parameters

<i>id</i> [IN]	<ul style="list-style-type: none"> PIN ID <ul style="list-style-type: none"> 1 (PIN1 / CHV1) 2 (PIN2 / CHV2)
<i>value</i> [IN]	<ul style="list-style-type: none"> PIN value of the PIN to be enabled/disabled

8.376.2 Field Documentation

8.376.2.1 uint8_t pack_dms_UIMVerifyPIN_t::id

8.376.2.2 uint8_t pack_dms_UIMVerifyPIN_t::value[255]

8.377 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.377.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result
------------------	---

8.377.2 Field Documentation

8.377.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

8.378 pack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.378.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result
------------------	---

8.378.2 Field Documentation

8.378.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

8.379 pack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t [imageListSize](#)
- [FMSPrefImageList](#) * [pImageList](#)
- uint32_t [bForceDownload](#)
- uint8_t [modemindex](#)
- uint16_t [Tlvresult](#)

8.379.1 Detailed Description

This structure contains the Set Images Preference pack

Parameters

<i>imageListSize</i>	<ul style="list-style-type: none"> Image List Size
<i>plmageList</i>	<ul style="list-style-type: none"> Image List See FMSPrefImageList
<i>bForceDownload</i>	<ul style="list-style-type: none"> 0 - Not Force Download. 1 - Focrce Download.
<i>modemindex</i>	<ul style="list-style-type: none"> Modem Index.
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack result

8.379.2 Field Documentation

8.379.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

8.379.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

8.379.2.3 uint8_t pack_fms_SetImagesPreference_t::modemindex

8.379.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::plmageList

8.379.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

8.380 pack_ims_SLQSImsConfigIndicationRegister_t Struct Reference

Data Fields

- uint8_t * [pSIPConfigEvents](#)
- uint8_t * [pRegMgrConfigEvents](#)
- uint8_t * [pSMSCConfigEvents](#)
- uint8_t * [pUserConfigEvents](#)
- uint8_t * [pVoIPConfigEvents](#)

8.380.1 Detailed Description

This structure contains parameters of IMS Config Indication Register

Parameters

<i>pSIPConfig-Events(optional)</i>	<ul style="list-style-type: none"> Registration Indication For SIP Configuration Events. 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"> 0x00 - Disable 0x01 - Enable
------------------------------------	--

<i>pRegMgrConfig-Events(optional)</i>	<ul style="list-style-type: none"> Registration Indication For Registration Manager Configuration Events. 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"> 0x00 - Disable 0x01 - Enable
<i>pSMSConfig-Events(optional)</i>	<ul style="list-style-type: none"> Registration Indication For SMS Configuration Events. 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"> 0x00 - Disable 0x01 - Enable
<i>pUserConfig-Events(optional)</i>	<ul style="list-style-type: none"> Registration Indication For User Configuration Events. 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"> 0x00 - Disable 0x01 - Enable
<i>pVoIPConfig-Events(optional)</i>	<ul style="list-style-type: none"> Registration Indication For VoIP Configuration Events. 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"> 0x00 - Disable 0x01 - Enable

Note

One of the optional parameter is mandatory to be present in the request.

8.380.2 Field Documentation

8.380.2.1 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pRegMgrConfigEvents

8.380.2.2 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pSIPConfigEvents

8.380.2.3 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pSMSConfigEvents

8.380.2.4 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pUserConfigEvents

8.380.2.5 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pVoIPConfigEvents

8.381 pack_ims_SLQSSetIMSSMSConfig_t Struct Reference**Data Fields**

- uint8_t * pSMSFormat
- uint8_t * pSMSOverIPNwInd
- uint8_t * pPhoneCtxtURLLen
- uint8_t * pPhoneCtxtURI

8.381.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig pack parameters.

Parameters

<i>pSMSFormat</i>	<ul style="list-style-type: none"> SMS format <ul style="list-style-type: none"> 0 - 3GPP 1 - 3GPP2
<i>pSMSOverIPNwInd</i>	<ul style="list-style-type: none"> SMS over IP Network Indication Flag <ul style="list-style-type: none"> TRUE - Turn on mobile-originated SMS FALSE - Turn off mobile-originated SMS
<i>pPhoneCtxtURLen</i>	<ul style="list-style-type: none"> Length of Phone context Universal Resource Identifier to follow
<i>pPhoneCtxtURI</i>	<ul style="list-style-type: none"> Phone context universal resource identifier Length of this string must be specified in pPhoneCtxtURLen parameter

8.381.2 Field Documentation

8.381.2.1 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pPhoneCtxtURI`

8.381.2.2 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pPhoneCtxtURLen`

8.381.2.3 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pSMSFormat`

8.381.2.4 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pSMSOverIPNwInd`

8.382 pack_ims_SLQSSetIMSUserConfig_t Struct Reference

Data Fields

- `uint8_t * pIMSDomainLen`
- `uint8_t * pIMSDomain`

8.382.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig pack parameters.

Parameters

<i>pIMSDomainLen</i>	<ul style="list-style-type: none"> Length of IMS Domain Name to follow
<i>pIMSDomain</i>	<ul style="list-style-type: none"> IMS domain name

8.382.2 Field Documentation

8.382.2.1 uint8_t* pack_ims_SLQSSetIMSUserConfig_t::pIMSDomain

8.382.2.2 uint8_t* pack_ims_SLQSSetIMSUserConfig_t::pIMSDomainLen

8.383 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

8.383.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig pack parameters.

Parameters

<i>pSessionExpiryTimer</i>	<ul style="list-style-type: none"> • Session duration, in seconds
<i>pMinSessionExpiryTimer</i>	<ul style="list-style-type: none"> • Minimum allowed value for session expiry timer, in seconds
<i>pAmrWbEnable</i>	<ul style="list-style-type: none"> • Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio • Values: <ul style="list-style-type: none"> – True - Enable – False - Disable
<i>pScrAmrEnable</i>	<ul style="list-style-type: none"> • Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB) • Values: <ul style="list-style-type: none"> – True - Enable – False - Disable
<i>pScrAmrWbEnable</i>	<ul style="list-style-type: none"> • Flag to enable/disable SCR for AMR WB Audio • Values: <ul style="list-style-type: none"> – True - Enable – False - Disable

<i>pAmrMode</i>	<ul style="list-style-type: none"> • BitMask for AMR NB modes allowed • Values: <ul style="list-style-type: none"> – 0x1 - 4.75 kbps – 0x2 - 5.15 kbps – 0x4 - 5.9 kbps – 0x8 - 6.17 kbps – 0x10 - 7.4 kbps – 0x20 - 7.95 kbps – 0x40 - 10.2 kbps – 0x80 - 12.2 kbps
<i>pAmrWBMode</i>	<ul style="list-style-type: none"> • BitMask for AMR WB modes allowed • Values: <ul style="list-style-type: none"> – 0x1 - 6.60 kbps – 0x2 - 8.85 kbps – 0x4 - 12.65 kbps – 0x8 - 14.25 kbps – 0x10 - 15.85 kbps – 0x20 - 18.25 kbps – 0x40 - 19.85 kbps – 0x80 - 23.05 kbps – 0x100 - 23.85 kbps
<i>pAmrOctet-Aligned</i>	<ul style="list-style-type: none"> • Flag to indicate if the octet is aligned for AMR NB Audio • Values: <ul style="list-style-type: none"> – True - Aligned – False - Not aligned, Bandwidth Efficient mode
<i>pAmrWBOctet-Aligned</i>	<ul style="list-style-type: none"> • Flag to indicate if the octet is aligned for AMR WB Audio • Values: <ul style="list-style-type: none"> – True - Aligned – False - Not aligned, Bandwidth Efficient mode
<i>pRingingTimer</i>	<ul style="list-style-type: none"> • 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.
<i>pRingBackTimer</i>	<ul style="list-style-type: none"> • 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.
<i>pRTPRTCP-InactTimer</i>	<ul style="list-style-type: none"> • 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.

8.383.2 Field Documentation

- 8.383.2.1 uint8_t* pack_ims_SLQSSetIMSVolIPConfig_t::pAmrMode
- 8.383.2.2 uint8_t* pack_ims_SLQSSetIMSVolIPConfig_t::pAmrOctetAligned
- 8.383.2.3 uint8_t* pack_ims_SLQSSetIMSVolIPConfig_t::pAmrWbEnable
- 8.383.2.4 uint16_t* pack_ims_SLQSSetIMSVolIPConfig_t::pAmrWBMode
- 8.383.2.5 uint8_t* pack_ims_SLQSSetIMSVolIPConfig_t::pAmrWBOctetAligned
- 8.383.2.6 uint16_t* pack_ims_SLQSSetIMSVolIPConfig_t::pMinSessionExpiryTimer
- 8.383.2.7 uint16_t* pack_ims_SLQSSetIMSVolIPConfig_t::pRingBackTimer
- 8.383.2.8 uint16_t* pack_ims_SLQSSetIMSVolIPConfig_t::pRingingTimer
- 8.383.2.9 uint16_t* pack_ims_SLQSSetIMSVolIPConfig_t::pRTPRTCPlnactTimer
- 8.383.2.10 uint8_t* pack_ims_SLQSSetIMSVolIPConfig_t::pScrAmrEnable
- 8.383.2.11 uint8_t* pack_ims_SLQSSetIMSVolIPConfig_t::pScrAmrWbEnable
- 8.383.2.12 uint16_t* pack_ims_SLQSSetIMSVolIPConfig_t::pSessionExpiryTimer

8.384 pack_ims_SLQSSetRegMgrConfig_t Struct Reference

Data Fields

- uint16_t * pPriCSCFPort
- uint8_t * pCSCFPortNameLen
- uint8_t * pCSCFPortName
- uint8_t * pIMSTestMode

8.384.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig pack parameters.

Parameters

<i>pPriCSCFPort</i>	<ul style="list-style-type: none"> • Primary call session control function port
<i>pCSCFPort-NameLen</i>	<ul style="list-style-type: none"> • Length of the CSCF Port name parameter to follow
<i>pCSCFPort-Name</i>	<ul style="list-style-type: none"> • Call Session control port, fully qualified domain name • Length of this string must be specified in pCSCFPortNameLen parameter
<i>pIMSTestMode</i>	<ul style="list-style-type: none"> • IMS Test mode Enabled. <ul style="list-style-type: none"> – TRUE - Enable, no IMS registration – FALSE - Disable, IMS registration is initiated

8.384.2 Field Documentation

8.384.2.1 uint8_t* pack_ims_SLQSSetRegMgrConfig_t::pCSCFPortName

8.384.2.2 uint8_t* pack_ims_SLQSSetRegMgrConfig_t::pCSCFPortNameLen

8.384.2.3 uint8_t* pack_ims_SLQSSetRegMgrConfig_t::pIMSTestMode

8.384.2.4 uint16_t* pack_ims_SLQSSetRegMgrConfig_t::pPriCSCFPort

8.385 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](#)

8.385.1 Detailed Description

This structure contains the SLQSSetSIPConfig pack parameters.

Parameters

<i>pSIPLocalPort</i>	<ul style="list-style-type: none"> • Primary call session control function SIP port number
<i>pTimerSIPReg</i>	<ul style="list-style-type: none"> • Initial SIP registration duration from the User equipment, in seconds
<i>pSubscribeTimer</i>	<ul style="list-style-type: none"> • Duration of the subscription by the UE for IMS registration notifications, in seconds
<i>pTimerT1</i>	<ul style="list-style-type: none"> • RTT estimate, in milliseconds
<i>pTimerT2</i>	<ul style="list-style-type: none"> • The maximum retransmit interval for non-invite requests and invite responses, in milliseconds
<i>pTimerTf</i>	<ul style="list-style-type: none"> • Non-invite transaction timeout timer, in milliseconds
<i>pSigComp-Enabled</i>	<ul style="list-style-type: none"> • Sig Comp Status <ul style="list-style-type: none"> – TRUE - Enable – FALSE - Disable

8.385.2 Field Documentation

8.385.2.1 uint8_t* pack_ims_SLQSSetSIPConfig_t::pSigCompEnabled

8.385.2.2 uint16_t* pack_ims_SLQSSetSIPConfig_t::pSIPLocalPort

8.385.2.3 uint32_t* pack_ims_SLQSSetSIPConfig_t::pSubscribeTimer

8.385.2.4 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerSIPReg

8.385.2.5 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerT1

8.385.2.6 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerT2

8.385.2.7 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerTf

8.386 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](#)

8.386.1 Detailed Description

This structure contains parameters of IMSA Config Indication Register

Parameters

<i>RegStatus-Config(optional)</i>	<ul style="list-style-type: none"> • Register Indication For Registration status. • 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"> – 0x00 - Disable – 0x01 - Enable
<i>ServiceStatus-Config(optional)</i>	<ul style="list-style-type: none"> • Register Indication For Service status Events. • 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"> – 0x00 - Disable – 0x01 - Enable
<i>RatHandover-Status-Config(optional)</i>	<ul style="list-style-type: none"> • Registration Indication For RAT handover status. • 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"> – 0x00 - Disable – 0x01 - Enable

<i>PdpStatus-Config(optional)</i>	<ul style="list-style-type: none"> • PDP Status Configuration. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
-----------------------------------	--

Note

One of the optional parameter is mandatory to be present in the request.

8.386.2 Field Documentation

8.386.2.1 `int pack_imsa_SLQSRegisterIMSAIndication_t::has_PdpStatusConfig`

8.386.2.2 `int pack_imsa_SLQSRegisterIMSAIndication_t::has_RatHandoverStatusConfig`

8.386.2.3 `int pack_imsa_SLQSRegisterIMSAIndication_t::has_RegStatusConfig`

8.386.2.4 `int pack_imsa_SLQSRegisterIMSAIndication_t::has_ServiceStatusConfig`

8.386.2.5 `uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::PdpStatusConfig`

8.386.2.6 `uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::RatHandoverStatusConfig`

8.386.2.7 `uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::RegStatusConfig`

8.386.2.8 `uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::ServiceStatusConfig`

8.387 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](#)

8.387.1 Detailed Description

This structure contains LOC delete assist data pack

Parameters

<i>pSVInfo</i>	<ul style="list-style-type: none"> • Pointer to struct loc_SVInfo. See loc_SVInfo for more information
<i>pGnssData</i>	<ul style="list-style-type: none"> • Pointer to struct loc_GnssData. See loc_GnssData for more information
<i>pCellDb</i>	<ul style="list-style-type: none"> • Pointer to struct loc_CellDb. See loc_CellDb for more information

<i>pClkInfo</i>	<ul style="list-style-type: none"> • Pointer to struct loc_ClkInfo. See loc_ClkInfo for more information
<i>pBdsSVInfo</i>	<ul style="list-style-type: none"> • Pointer to struct loc_BdsSVInfo. See loc_BdsSVInfo for more information
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Pack delete assist data request result.

8.387.2 Field Documentation

8.387.2.1 `loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo`

8.387.2.2 `loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb`

8.387.2.3 `loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo`

8.387.2.4 `loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData`

8.387.2.5 `loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo`

8.387.2.6 `uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult`

8.388 pack_loc_EventRegister_t Struct Reference

Data Fields

- `uint64_t` [eventRegister](#)
- `uint16_t` [Tlvresult](#)

8.388.1 Detailed Description

This structure contains the Parameter for RegisterEvents

Parameters

<i>eventRegister</i>	<ul style="list-style-type: none"> • Specifies the events that the control point is interested in receiving. -Values <ul style="list-style-type: none"> – 0x00000001 - to receive position report event indications – 0x00000002 - to receive satellite report event indications. These reports are sent at a 1 Hz rate. – 0x00000004 - to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate. – 0x00000008 - to receive NI Notify/Verify request event indications – 0x00000010 - to receive time injection request event indications. – 0x00000020 - to receive predicted orbits request event indications. – 0x00000040 - to receive position injection request event indications. – 0x00000080 - to receive engine state report event indications. – 0x00000100 - to receive fix session status report event indications. – 0x00000200 - to receive Wi-Fi position request event indications. – 0x00000400 - to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.). – 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. – 0x00001000 - to receive Stationary Position Indicator (SPI) streaming report indications. – 0x00002000 - to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server. – 0x00004000 - to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited. – 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. – 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. – 0x00020000 - to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports. – 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. – 0x00080000 - to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session. – 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. – 0x00200000 - to receive Wi-Fi Access Point (AP) data inject request event indications. – 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. – 0x00800000 - to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.). – 0x01000000 - to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). – 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.
----------------------	--

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"> • Pack result.
------------------	--

8.388.2 Field Documentation

8.388.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

8.388.2.2 uint16_t pack_loc_EventRegister_t::Tlvresult

8.389 pack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint32_t [extPowerState](#)
- uint16_t [Tlvresult](#)

8.389.1 Detailed Description

This structure contains the Parameter External Power Source State pack.

Parameters

<i>extPowerState</i>	<ul style="list-style-type: none"> • Specifies the Power state; injected by the control point. • Values <ul style="list-style-type: none"> – 0 - Device is not connected to an external power source – 1 - Device is connected to an external power source – 2 - Unknown external power state
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Pack result.

8.389.2 Field Documentation

8.389.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

8.389.2.2 uint16_t pack_loc_SetExtPowerState_t::Tlvresult

8.390 pack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint32_t [mode](#)
- uint16_t [Tlvresult](#)

8.390.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

<i>mode</i>	<ul style="list-style-type: none"> Valid values: <ul style="list-style-type: none"> eQMI_LOC_OPER_MODE_DEFAULT (1) - Use the default engine mode eQMI_LOC_OPER_MODE_MSB (2) - Use the MS-based mode eQMI_LOC_OPER_MODE_MSA (3) - Use the MS-assisted mode eQMI_LOC_OPER_MODE_STANDALONE (4) - Use Standalone mode eQMI_LOC_OPER_MODE_CELL_ID (5) - Use cell ID; this mode is only valid for GSM/UMTS networks 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
<i>Tlvresult</i>	<ul style="list-style-type: none"> Pack result.

8.390.2 Field Documentation

8.390.2.1 `uint32_t pack_loc_SetOperationMode_t::mode`

8.390.2.2 `uint16_t pack_loc_SetOperationMode_t::Tlvresult`

8.391 `pack_loc_SLQSLOCGetBestAvailPos_t` Struct Reference

Data Fields

- `uint32_t xid`
- `uint16_t Tlvresult`

8.391.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

<i>xid</i>	<ul style="list-style-type: none"> Identifies the transaction. The transaction ID is returned in the Get Best Available Position indication.
<i>Tlvresult</i>	<ul style="list-style-type: none"> Pack result.

8.391.2 Field Documentation

8.391.2.1 `uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult`

8.391.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

8.392 pack_loc_SLQSLOCGetServer_t Struct Reference

Data Fields

- uint32_t [serverType](#)
- int [has_serverAddrTypeMask](#)
- uint8_t [serverAddrTypeMask](#)

8.392.1 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"> • Type of server. • Valid values <ul style="list-style-type: none"> – eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE – eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC – eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP – eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4)- Server type is custom PDE
<i>serverAddrTypeMask</i>	<ul style="list-style-type: none"> • Optional parameter • Type of address the client wants. If unspecified, the indication will contain all the types of addresses it has for the specified server type. • Valid bitmasks <ul style="list-style-type: none"> – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - URL

8.392.2 Field Documentation

8.392.2.1 int pack_loc_SLQSLOCGetServer_t::has_serverAddrTypeMask

8.392.2.2 uint8_t pack_loc_SLQSLOCGetServer_t::serverAddrTypeMask

8.392.2.3 uint32_t pack_loc_SLQSLOCGetServer_t::serverType

8.393 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](#)

8.393.1 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"> • Optional parameter • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude • Note - This field must be specified together with pLongitude and pHorUncCircular.
<i>longitude</i>	<ul style="list-style-type: none"> • Optional parameter • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude • Note - This field must be specified together with pLatitude and pHorUncCircular.

<i>horUncCircular</i>	<ul style="list-style-type: none"> • Optional parameter • Horizontal position uncertainty. • Units - Meters • Note - This field must be specified together with pLatitude and pLongitude.
<i>horConfidence</i>	<ul style="list-style-type: none"> • Optional parameter • Horizontal confidence. • Units - Percent • Values <ul style="list-style-type: none"> – Valid Values - 1 to 99 – Invalid Values - 0, 101 to 255 – If 100 is received, reinterpret to 99 • Note - This field must be specified together with horizontal uncertainty. If not specified when pHorUncCircular is set, the default value is 50.
<i>horReliability</i>	<ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – Values <ul style="list-style-type: none"> * 0 - Location reliability is not set. * 1 - Location reliability is very low; use it at your own risk * 2 - Location reliability is low; little or no cross-checking is possible. * 3 - Location reliability is medium; limited cross-check passed * 4 - Location reliability is high; strong cross-check passed
<i>altitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> • Optional parameter • Altitude With Respect to Ellipsoid. • Units - Meters • Values <ul style="list-style-type: none"> – Positive - height – Negative = depth
<i>altitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> • Optional parameter • Altitude With Respect to Sea Level. • Units - Meters
<i>vertUnc</i>	<ul style="list-style-type: none"> • Optional parameter • Vertical uncertainty. • Units - Meters • Note - This is mandatory if either pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified.

<i>vertConfidence</i>	<ul style="list-style-type: none"> • Optional parameter • Vertical confidence. • Units - Percentage • Values <ul style="list-style-type: none"> – Valid Values - 0 to 99 – Invalid Values - 0, 100-256 – If 100 is received, reinterpret to 99 • Note - This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50.
<i>vertReliability</i>	<ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – Values <ul style="list-style-type: none"> * 0 - Location reliability is not set. * 1 - Location reliability is very low; use it at your own risk. * 2 - Location reliability is low; little or no cross-checking is possible * 3 - Location reliability is medium; limited cross-check passed * 4 - Location reliability is high; strong cross-check passed
<i>altitudeSrcInfo</i>	<ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – Pointer to struct altitudeSrcInfo. See altitudeSrcInfo for more information
<i>timestampUtc</i>	<ul style="list-style-type: none"> • Optional parameter • UTC timestamp • Units - Milliseconds since Jan. 1, 1970
<i>timestampAge</i>	<ul style="list-style-type: none"> • Optional parameter • Position age, which is an estimate of how long ago this fix was made. • Units - Milliseconds
<i>positionSrc</i>	<ul style="list-style-type: none"> • Optional parameter • Source from which this position was obtained • Valid values <ul style="list-style-type: none"> – 0 - Position source is GNSS – 1 - Position source is Cell ID – 2 - Position source is Enhanced Cell ID – 3 - Position source is Wi-Fi – 4 - Position source is Terrestrial – 5 - Position source is GNSS Terrestrial Hybrid – 6 - Other sources • 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"> – If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source.

<i>rawHorUnc-Circular</i>	<ul style="list-style-type: none"> • Optional parameter • Horizontal position uncertainty (circular) without any optimization. • Units - Meters
<i>rawHor-Confidence</i>	<ul style="list-style-type: none"> • Optional parameter • Horizontal confidence associated with raw horizontal uncertainty • Units: Percent • Values <ul style="list-style-type: none"> – Valid values - 1 to 99 – Invalid values - 0, 101 to 255 – If 100 is received, reinterpret to 99 • Note - This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50.

8.393.2 Field Documentation

8.393.2.1 altSrcInfo_t pack_loc_SLQSLOCInjectPosition_t::altitudeSrcInfo

8.393.2.2 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtEllipsoid

8.393.2.3 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtMeanSeaLevel

8.393.2.4 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeSrcInfo

8.393.2.5 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtEllipsoid

8.393.2.6 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtMeanSeaLevel

8.393.2.7 int pack_loc_SLQSLOCInjectPosition_t::has_horConfidence

8.393.2.8 int pack_loc_SLQSLOCInjectPosition_t::has_horReliability

8.393.2.9 int pack_loc_SLQSLOCInjectPosition_t::has_horUncCircular

8.393.2.10 int pack_loc_SLQSLOCInjectPosition_t::has_latitude

8.393.2.11 int pack_loc_SLQSLOCInjectPosition_t::has_longitude

8.393.2.12 int pack_loc_SLQSLOCInjectPosition_t::has_positionSrc

8.393.2.13 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorConfidence

8.393.2.14 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorUncCircular

8.393.2.15 int pack_loc_SLQSLOCInjectPosition_t::has_timestampAge

8.393.2.16 int pack_loc_SLQSLOCInjectPosition_t::has_timestampUtc

8.393.2.17 int pack_loc_SLQSLOCInjectPosition_t::has_vertConfidence

8.393.2.18 int pack_loc_SLQSLOCInjectPosition_t::has_vertRelicability

8.393.2.19 int pack_loc_SLQSLOCInjectPosition_t::has_vertUnc

8.393.2.20 uint8_t pack_loc_SLQSLOCInjectPosition_t::horConfidence

8.393.2.21 uint32_t pack_loc_SLQSLOCInjectPosition_t::horReliability

8.393.2.22 float pack_loc_SLQSLOCInjectPosition_t::horUncCircular

8.393.2.23 double pack_loc_SLQSLOCInjectPosition_t::latitude

8.393.2.24 double pack_loc_SLQSLOCInjectPosition_t::longitude

8.393.2.25 uint32_t pack_loc_SLQSLOCInjectPosition_t::positionSrc

8.393.2.26 uint8_t pack_loc_SLQSLOCInjectPosition_t::rawHorConfidence

8.393.2.27 float pack_loc_SLQSLOCInjectPosition_t::rawHorUncCircular

8.393.2.28 uint32_t pack_loc_SLQSLOCInjectPosition_t::timestampAge

8.393.2.29 uint64_t pack_loc_SLQSLOCInjectPosition_t::timestampUtc

8.393.2.30 uint8_t pack_loc_SLQSLOCInjectPosition_t::vertConfidence

8.393.2.31 uint32_t pack_loc_SLQSLOCInjectPosition_t::vertReliability

8.393.2.32 float pack_loc_SLQSLOCInjectPosition_t::vertUnc

8.394 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](#)

8.394.1 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"> • Opaque Identifier (Optional parameter) • 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.
<i>acceleroData</i>	<ul style="list-style-type: none"> • 3-Axis Accelerometer Data (Optional parameter) • Pointer to struct sensorData. See sensorData_t for more information
<i>gyroData</i>	<ul style="list-style-type: none"> • 3-Axis Gyroscope Data (Optional parameter) • Pointer to struct sensorData. See sensorData_t for more information
<i>acceleroTimeSrc</i>	<ul style="list-style-type: none"> • 3-Axis Accelerometer Data Time Source (Optional parameter) • The location service uses this field to identify the time reference used in the accelerometer data time stamps. • If not specified, the location service assumes that the time source for the accelerometer data is unknown. • Valid values <ul style="list-style-type: none"> – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
<i>gyroTimeSrc</i>	<ul style="list-style-type: none"> • 3-Axis Gyroscope Data Time Source (Optional) • The location service uses this field to identify the time reference used in the gyroscope data time stamps. • If not specified, the location service assumes that the time source for the gyroscope data is unknown. • Valid values <ul style="list-style-type: none"> – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
<i>accelTemp</i>	<ul style="list-style-type: none"> • Accelerometer Temperature Data (Optional parameter) • Pointer to struct tempratureData. See tempData_t for more information
<i>gyroTemp</i>	<ul style="list-style-type: none"> • Gyroscope Temperature Data (Optional parameter) • Pointer to struct tempratureData. See tempData_t for more information

8.394.2 Field Documentation

8.394.2.1 `sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData`8.394.2.2 `uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc`8.394.2.3 `tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp`8.394.2.4 `sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData`

8.394.2.5 `tempData_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroTemp`

8.394.2.6 `uint32_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc`

8.394.2.7 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc`

8.394.2.8 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp`

8.394.2.9 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_accleroData`

8.394.2.10 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroData`

8.394.2.11 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp`

8.394.2.12 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc`

8.394.2.13 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_opaqueld`

8.394.2.14 `uint32_t` `pack_loc_SLQSLOCInjectSensorData_t::opaqueld`

8.395 `pack_loc_SLQSLOCInjectUTCTime_t` Struct Reference

Data Fields

- `uint64_t` [timeMsec](#)
- `uint32_t` [timeUncMsec](#)

8.395.1 Detailed Description

This structure contains inject UTC time parameter.

Parameters

<i>timeMsec</i>	<ul style="list-style-type: none"> • The UTC time since Jan. 1, 1970
<i>timeUncMsec</i>	<ul style="list-style-type: none"> • The time Uncertainty

8.395.2 Field Documentation

8.395.2.1 `uint64_t` `pack_loc_SLQSLOCInjectUTCTime_t::timeMsec`

8.395.2.2 `uint32_t` `pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec`

8.396 `pack_loc_SLQSLOCSetCradleMountConfig_t` Struct Reference

Data Fields

- `uint32_t` [state](#)
- `int` [has_confidence](#)
- `uint8_t` [confidence](#)

8.396.1 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"> • Cradle Mount State • Valid values: <ul style="list-style-type: none"> – 0 - Device is mounted on the cradle – 1 - Device is not mounted on the cradle – 2 - Unknown cradle mount state
<i>confidence</i>	<ul style="list-style-type: none"> • Cradle Mount Confidence (Optional) • Confidence in the Cradle Mount state expressed as a percentage. • Range - 0 to 100

8.396.2 Field Documentation

8.396.2.1 uint8_t pack_loc_SLQSLOCSetCradleMountConfig_t::confidence

8.396.2.2 int pack_loc_SLQSLOCSetCradleMountConfig_t::has_confidence

8.396.2.3 uint32_t pack_loc_SLQSLOCSetCradleMountConfig_t::state

8.397 pack_loc_SLQSLOCSetServer_t Struct Reference

Data Fields

- uint32_t [serverType](#)
- [loc_IPv4Config](#) * [pIPv4Config](#)
- [loc_IPv6Config](#) * [pIPv6Config](#)
- [loc_URLAddrInfo](#) * [pURLAddr](#)

8.397.1 Detailed Description

This structure contains parameters to set A-GPS Server

Parameters

<i>serverType</i>	<ul style="list-style-type: none"> • Type of server • Valid values: <ul style="list-style-type: none"> – 1 - Server type is CDMA PDE – 2 - Server type is CDMA MPC – 3 - Server type is UMTS SLP – 4 - Server type is custom PDE
-------------------	---

<i>pIPv4Config</i>	<ul style="list-style-type: none"> IPv4 address and port (Optional) See loc_IPv4Config for more information
<i>pIPv6Config</i>	<ul style="list-style-type: none"> IPv6 address and port (Optional) See loc_IPv6Config for more information
<i>pURLAddr</i>	<ul style="list-style-type: none"> Uniform Resource Locator (Optional) See loc_URLAddrInfo for more information

8.397.2 Field Documentation

8.397.2.1 `loc_IPv4Config*` `pack_loc_SLQSLOCSetServer_t::pIPv4Config`

8.397.2.2 `loc_IPv6Config*` `pack_loc_SLQSLOCSetServer_t::pIPv6Config`

8.397.2.3 `loc_URLAddrInfo*` `pack_loc_SLQSLOCSetServer_t::pURLAddr`

8.397.2.4 `uint32_t` `pack_loc_SLQSLOCSetServer_t::serverType`

8.398 `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](#)

8.398.1 Detailed Description

This structure contains the LOC Start pack

Parameters

<i>SessionId</i>	<ul style="list-style-type: none"> ID of the session as identified by the control point. Range: 0 to 255
<i>pRecurrence-Type</i>	<ul style="list-style-type: none"> Optional Parameter Specifies the type of session in which the control point is interested. Defaults to SINGLE. -Values <ul style="list-style-type: none"> 1 - Request periodic position fixes 2 - Request a single position fix

<i>pHorizontal-AccuracyLvl</i>	<ul style="list-style-type: none"> • Optional Parameter • Specifies the horizontal accuracy level required by the control point. • Defaults to LOW • Values <ul style="list-style-type: none"> – 1 - Low accuracy – 2 - Medium accuracy – 3 - High accuracy
<i>pIntermediate-ReportState</i>	<ul style="list-style-type: none"> • Optional Parameter • Specifies if the control point is interested in receiving intermediate reports. • ON by default. • Values <ul style="list-style-type: none"> – 1 - Intermediate reports are turned on – 2 - Intermediate reports are turned off
<i>pMinInterval-Time</i>	<ul style="list-style-type: none"> • Optional Parameter • Minimum time interval, specified by the control point, that must elapse between position reports. • Units - Milliseconds • Default - 1000 ms
<i>LocApplication-Info</i>	<ul style="list-style-type: none"> • Optional Parameter • LOC Application Parameters • See loc_LocApplicationInfo for more information
<i>pConfigAltitude-Assumed</i>	<ul style="list-style-type: none"> • Optional Parameter • Configuration for Altitude Assumed Info in GNSS SV Info Event • Defaults to ENABLED. • Values <ul style="list-style-type: none"> – 1 - Enable Altitude Assumed information in GNSS SV Info Event – 2 - Disable Altitude Assumed information in GNSS SV Info Event
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result.

8.398.2 Field Documentation

8.398.2.1 `loc_LocApplicationInfo*` `pack_loc_Start_t::pApplicationInfo`

8.398.2.2 `uint32_t*` `pack_loc_Start_t::pConfigAltitudeAssumed`

8.398.2.3 `uint32_t*` `pack_loc_Start_t::pHorizontalAccuracyLvl`

8.398.2.4 `uint32_t*` `pack_loc_Start_t::pIntermediateReportState`

8.398.2.5 `uint32_t*` `pack_loc_Start_t::pMinIntervalTime`

8.398.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

8.398.2.7 uint8_t pack_loc_Start_t::SessionId

8.398.2.8 uint16_t pack_loc_Start_t::Tlvresult

8.399 pack_loc_Stop_t Struct Reference

Data Fields

- uint8_t [SessionId](#)
- uint16_t [Tlvresult](#)

8.399.1 Detailed Description

This structure contains Stop LOC pack

Parameters

<i>sessionId</i>	<ul style="list-style-type: none"> • ID of the session as identified by the control point. • Range: 0 to 255
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result.

8.399.2 Field Documentation

8.399.2.1 uint8_t pack_loc_Stop_t::SessionId

8.399.2.2 uint16_t pack_loc_Stop_t::Tlvresult

8.400 pack_nas_InitiateDomainAttach_t Struct Reference

Data Fields

- uint32_t [action](#)

8.400.1 Detailed Description

Pack structure for Initiates a domain attach/detach of the device.

Parameters

<i>action</i>	<ul style="list-style-type: none"> • Domain action to attempt <ul style="list-style-type: none"> 1 - Attach 2 - Detach
---------------	--

8.400.2 Field Documentation

8.400.2.1 uint32_t pack_nas_InitiateDomainAttach_t::action

8.401 pack_nas_PerformNetworkScanPCI_t Struct Reference

Data Fields

- uint8_t * [pNetworkType](#)
- uint32_t * [pScanType](#)
- uint64_t * [pCiotOpModePref](#)
- uint64_t * [pLteM1BandPref](#)
- uint64_t * [pLteNB1BandPref](#)

8.401.1 Field Documentation

8.401.1.1 uint64_t* pack_nas_PerformNetworkScanPCI_t::pCiotOpModePref

8.401.1.2 uint64_t* pack_nas_PerformNetworkScanPCI_t::pLteM1BandPref

8.401.1.3 uint64_t* pack_nas_PerformNetworkScanPCI_t::pLteNB1BandPref

8.401.1.4 uint8_t* pack_nas_PerformNetworkScanPCI_t::pNetworkType

8.401.1.5 uint32_t* pack_nas_PerformNetworkScanPCI_t::pScanType

8.402 pack_nas_SetACCOLC_t Struct Reference

Data Fields

- int8_t [spc](#) [6]
- uint8_t [accolc](#)

8.402.1 Detailed Description

Sets the access overload class (ACCOLC)

Parameters

<i>spc</i>	<ul style="list-style-type: none"> • service programming code NULL-terminated string of six digit
<i>accolc</i>	<ul style="list-style-type: none"> • ACCOLC : Valid range is 0 to 15

8.402.2 Field Documentation

8.402.2.1 uint8_t pack_nas_SetACCOLC_t::accolc

8.402.2.2 int8_t pack_nas_SetACCOLC_t::spc[6]

8.403 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](#)

8.403.1 Detailed Description

Pack structure to set CDMA Network parameters.

Parameters

<i>pSPC</i>	<ul style="list-style-type: none"> • Six digit service programming code (not necessary when only the roaming field is being set)
<i>pForceRev0</i>	<ul style="list-style-type: none"> • (Optional)Force CDMA 1x-EV-DO Rev. 0 mode <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled Note: Enabled can only be specified if pCustomSCP state is set to Disabled
<i>pCustomSCP</i>	<ul style="list-style-type: none"> • (Optional)Use a custom config for CDMA 1x-EV-DO SCP <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled Note: Enabled can only be specified if pForceRev0 is set to Disabled
<i>pProtocol</i>	<ul style="list-style-type: none"> • Protocol mask for custom SCP config <ul style="list-style-type: none"> – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port – 0xFFFFFFFF - Unknown
<i>pBroadcast</i>	<ul style="list-style-type: none"> • Broadcast mask for custom SCP config <ul style="list-style-type: none"> – 0x00000001 - Generic broadcast enabled – 0xFFFFFFFF - Unknown
<i>pApplication</i>	<ul style="list-style-type: none"> • Application mask for custom SCP config <ul style="list-style-type: none"> – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application – 0xFFFFFFFF - Unknown

<i>pRoaming</i>	<ul style="list-style-type: none"> • Roaming preference <ul style="list-style-type: none"> – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming – 0xFFFFFFFF - Unknown
-----------------	--

8.403.2 Field Documentation

8.403.2.1 uint32_t* pack_nas_SetCDMANetworkParameters_t::pApplication

8.403.2.2 uint32_t* pack_nas_SetCDMANetworkParameters_t::pBroadcast

8.403.2.3 uint8_t* pack_nas_SetCDMANetworkParameters_t::pCustomSCP

8.403.2.4 uint8_t* pack_nas_SetCDMANetworkParameters_t::pForceRev0

8.403.2.5 uint32_t* pack_nas_SetCDMANetworkParameters_t::pProtocol

8.403.2.6 uint32_t* pack_nas_SetCDMANetworkParameters_t::pRoaming

8.403.2.7 char* pack_nas_SetCDMANetworkParameters_t::pSPC

8.404 pack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint32_t [TechnologyPref](#)
- uint32_t [Duration](#)
- uint16_t [Tlvresult](#)

8.404.1 Detailed Description

This structure contains pack set network preference parameters.

Parameters

<i>TechnologyPref[IN]</i>	<ul style="list-style-type: none"> • Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: <ul style="list-style-type: none"> – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: <ul style="list-style-type: none"> – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
---------------------------	--

<i>Duration[IN]</i>	<ul style="list-style-type: none"> • Duration of active preference <ul style="list-style-type: none"> – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
<i>Tlvresult</i>	<ul style="list-style-type: none"> • pack result

8.404.2 Field Documentation

8.404.2.1 `uint32_t pack_nas_SetNetworkPreference_t::Duration`

8.404.2.2 `uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref`

8.404.2.3 `uint16_t pack_nas_SetNetworkPreference_t::Tlvresult`

8.405 `pack_nas_SLQSCfgSigInfo_t` Struct Reference

Data Fields

- `nas_RSSIthresh` * `pRSSIthresh`
- `nas_ECIOthresh` * `pECIOthresh`
- `nas_HDRSINRthresh` * `pHDRSINRthresh`
- `nas_LTESNRthresh` * `pLTESNRthresh`
- `nas_IOTthresh` * `pIOTthresh`
- `nas_RSRQthresh` * `pRSRQthresh`
- `nas_RSRPthresh` * `pRSRPthresh`
- `nas_LTESigRptCfg` * `pLTESigRptCfg`
- `nas_TDSCDMASINRCONFTthresh` * `pTDSCDMASINRCONFTthresh`

8.405.1 Detailed Description

Structure for ConfigSigInfo pack.

Parameters

<i>pRSSIthresh</i>	<ul style="list-style-type: none"> • RSSI threshold List • See nas_RSSIthresh for more details
<i>pECIOthresh</i>	<ul style="list-style-type: none"> • ECIO Threshold List • See nas_ECIOthresh for more details
<i>pHDRSINR- Thresh</i>	<ul style="list-style-type: none"> • HDR SINR Threshold List • See nas_HDRSINRthresh for more details

<i>pLTESNR- Thresh</i>	<ul style="list-style-type: none"> • LTE SNR Threshold List • See nas_LTESNRThresh for more details
<i>pIOTthresh</i>	<ul style="list-style-type: none"> • IO Threshold List • See nas_IOTthresh for more details
<i>pRSRQThresh</i>	<ul style="list-style-type: none"> • RSRQ Threshold List • See nas_RSRQThresh for more details
<i>pRSRPThresh</i>	<ul style="list-style-type: none"> • RSRP Threshold List • See nas_RSRPThresh for more details
<i>pLTESigRptCfg</i>	<ul style="list-style-type: none"> • LTE signal report config • See nas_LTESigRptCfg for more details
<i>pTDSCDMASIN- RCONFThresh</i>	<ul style="list-style-type: none"> • TD-SCDMA SINR Threshold List • See nas_TDSCDMASINRCONFThresh for more details

8.405.2 Field Documentation

8.405.2.1 **nas_ECIOTthresh*** pack_nas_SLQSConfigSigInfo_t::pECIOTthresh

8.405.2.2 **nas_HDRSINRThresh*** pack_nas_SLQSConfigSigInfo_t::pHDRSINRThresh

8.405.2.3 **nas_IOTthresh*** pack_nas_SLQSConfigSigInfo_t::pIOTthresh

8.405.2.4 **nas_LTESigRptCfg*** pack_nas_SLQSConfigSigInfo_t::pLTESigRptCfg

8.405.2.5 **nas_LTESNRThresh*** pack_nas_SLQSConfigSigInfo_t::pLTESNRThresh

8.405.2.6 **nas_RSRPThresh*** pack_nas_SLQSConfigSigInfo_t::pRSRPThresh

8.405.2.7 **nas_RSRQThresh*** pack_nas_SLQSConfigSigInfo_t::pRSRQThresh

8.405.2.8 **nas_RSSIThresh*** pack_nas_SLQSConfigSigInfo_t::pRSSIThresh

8.405.2.9 **nas_TDSCDMASINRCONFThresh*** pack_nas_SLQSConfigSigInfo_t::pTDSCDMASINRCONFThresh

8.406 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint16_t [mcc](#)
- uint16_t [mnc](#)
- uint8_t * [pMncPcsStatus](#)

8.406.1 Detailed Description

This structure contains pack get operator name for specified network parameters.

Parameters

<i>mcc</i>	<ul style="list-style-type: none"> A 16-bit integer representation of MCC. Range: 0 to 999
<i>mnc</i>	<ul style="list-style-type: none"> A 16-bit integer representation of MNC. Range: 0 to 999
<i>pMncPcsStatus</i>	<ul style="list-style-type: none"> MNC PCS Digit Include Status Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01). Values <ul style="list-style-type: none"> TRUE - MNC is a three-digit value. e.g. a reported value of 90 corresponds to an MNC value of 090 FALSE - MNC is a two-digit value. e.g. a reported value of 90 corresponds to an MNC value of 90

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.

8.406.2 Field Documentation

8.406.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.406.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.406.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.407 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

Data Fields

- uint32_t [regAction](#)
- [nas_MNRInfo](#) * [pMNRInfo](#)
- uint32_t * [pChangeDuration](#)
- uint8_t * [pMncPcsDigitStatus](#)

8.407.1 Detailed Description

This structure contains Initiate Network Registration request parameters

Parameters

<i>regAction</i>	<ul style="list-style-type: none"> Specifies one of the following register actions : <ul style="list-style-type: none"> AUTO_REGISTER - Device registers according to its provisioning and optional parameters supplied with the command are ignored. 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.
<i>pMNRInfo</i>	[Optional] <ul style="list-style-type: none"> Pointer to structure MNRInfo <ul style="list-style-type: none"> See nas_MNRInfo for more information
<i>pChange-Duration</i>	[Optional] <ul style="list-style-type: none"> Duration of the change. <ul style="list-style-type: none"> 0x00 - Power cycle - Remains active until the next device power cycle 0x01 - Permanent - Remains active through power cycles until changed by the client
<i>pMncPcsDigit-Status</i>	[Optional] <ul style="list-style-type: none"> MNC PCS Digit Include Status <ul style="list-style-type: none"> True - MNC is a 3-digit value. False - MNC is a 2-digit value.

8.407.2 Field Documentation

8.407.2.1 uint32_t* pack_nas_SLQSIInitiateNetworkRegistration_t::pChangeDuration

8.407.2.2 uint8_t* pack_nas_SLQSIInitiateNetworkRegistration_t::pMncPcsDigitStatus

8.407.2.3 nas_MNRInfo* pack_nas_SLQSIInitiateNetworkRegistration_t::pMNRInfo

8.407.2.4 uint32_t pack_nas_SLQSIInitiateNetworkRegistration_t::regAction

8.408 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 * pLTESNRThresh
- uint16_t * pLTESNRDelta
- nas_LTERSRQThresh * pLTERSRQThresh
- uint16_t * pLTERSRQDelta
- nas_LTERSRPThresh * pLTERSRPThresh
- uint16_t * pLTERSRPDelta
- nas_LTESigRptConfig * pLTESigRptConfig
- nas_TDSCDMARSCPThresh * pTDSCDMARSCPThresh
- uint16_t * pTDSCDMARSCPDelta
- nas_TDSCDMARSSIThresh * pTDSCDMARSSIThresh
- float * pTDSCDMARSSIDelta
- nas_TDSCDMAECIOThresh * pTDSCDMAECIOThresh
- float * pTDSCDMAECIODelta
- nas_TDSCDMASINRThresh * pTDSCDMASINRThresh
- float * pTDSCDMASINRDelta

8.408.1 Detailed Description

This structure contains pack set the signal strength reporting thresholds parameters.

Parameters

<i>pCDMARSSI- Thresh</i>	<ul style="list-style-type: none"> • CDMA RSSI threshold List
<i>pCDMARSSI- Delta</i>	<ul style="list-style-type: none"> • RSSI delta (in units of 0.1 dBm). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pCDMAECIO- Thresh</i>	<ul style="list-style-type: none"> • CDMA ECIO Threshold List
<i>pCDMAECIO- Delta</i>	<ul style="list-style-type: none"> • ECIO delta (in units of 0.1 dB). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pHDDRSSI- Thresh</i>	<ul style="list-style-type: none"> • HDR RSSI Threshold List
<i>pHDDRSSIDelta</i>	<ul style="list-style-type: none"> • RSSI delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pHDRECIO- Thresh</i>	<ul style="list-style-type: none"> • HDR ECIO Threshold List
<i>pHDRECIODelta</i>	<ul style="list-style-type: none"> • ECIO delta (in units of 0.1 dB) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.

<i>pHDRSINR- Thresh</i>	<ul style="list-style-type: none"> HDR SINR Threshold List
<i>pHDRSINRDelta</i>	<ul style="list-style-type: none"> SINR delta (in units of 1 SINR level) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pHDRIOTresh</i>	<ul style="list-style-type: none"> HDR IO Threshold List
<i>pHDRIODelta</i>	<ul style="list-style-type: none"> IO delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pGSMRSSI- Thresh</i>	<ul style="list-style-type: none"> GSM RSSI Threshold List See nas_GSMRSSIThresh for more details
<i>pGSMRSSIDelta</i>	<ul style="list-style-type: none"> RSSI delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pWCDMARSSI- Thresh</i>	<ul style="list-style-type: none"> WCDMA RSSI Threshold List See nas_WCDMARSSIThresh for more details
<i>pWCDMARSSI- Delta</i>	<ul style="list-style-type: none"> RSSI delta (in units of 0.1 dBm). A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pWCDMAECIO- Thresh</i>	<ul style="list-style-type: none"> WCDMA ECIO Threshold List
<i>pWCDMAECIO- Delta</i>	<ul style="list-style-type: none"> ECIO delta (in units of 0.1 dB) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSSI- Thresh</i>	<ul style="list-style-type: none"> LTE RSSI Threshold List
<i>pLTERSSIDelta</i>	<ul style="list-style-type: none"> RSSI delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTESNR- Thresh</i>	<ul style="list-style-type: none"> LTE SNR Threshold List
<i>pLTESNRDelta</i>	<ul style="list-style-type: none"> SNR delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.

<i>pLTERSRQ- Thresh</i>	<ul style="list-style-type: none"> • LTE RSRQ Threshold List
<i>pLTERSRQ- Delta</i>	<ul style="list-style-type: none"> • RSRQ delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSRP- Thresh</i>	<ul style="list-style-type: none"> • LTE RSRP Threshold List
<i>pLTERSRPDelta</i>	<ul style="list-style-type: none"> • RSRP delta (in units of 0.1 dBm). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSigRpt- Config</i>	<ul style="list-style-type: none"> • LTE Signal Report Config
<i>pTDSCDMARS- CPTthresh</i>	<ul style="list-style-type: none"> • TDSCDMA RSCP Threshold List
<i>pTDSCDMARS- CPDelta</i>	<ul style="list-style-type: none"> • RSCP delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pTDSCDMARS- SIThresh</i>	<ul style="list-style-type: none"> • TDSCDMA RSSI Threshold List
<i>pTDSCDMARS- SIDelta</i>	<ul style="list-style-type: none"> • RSSI delta (in dBm) used by TD-SCDMA.
<i>pTDSCDMAECI- OTthresh</i>	<ul style="list-style-type: none"> • TDSCDMA ECIO Threshold List
<i>pTDSCDMAECI- ODelta</i>	<ul style="list-style-type: none"> • ECIO delta (in dB) used by TD-SCDMA
<i>pTDSCDMASIN- RThresh</i>	<ul style="list-style-type: none"> • TDSCDMA SINR Threshold List
<i>pTDSCDMASIN- RDelta</i>	<ul style="list-style-type: none"> • SINR delta (in dB) used by TD-SCDMA.

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.

8.408.2 Field Documentation

8.408.2.1 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta`

8.408.2.2 `nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh`

- 8.408.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta
- 8.408.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh
- 8.408.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta
- 8.408.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh
- 8.408.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta
- 8.408.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh
- 8.408.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta
- 8.408.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh
- 8.408.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIDelta
- 8.408.2.12 nas_HDRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIThresh
- 8.408.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRDelta
- 8.408.2.14 nas_HDRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRThreshold
- 8.408.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta
- 8.408.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh
- 8.408.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta
- 8.408.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh
- 8.408.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta
- 8.408.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh
- 8.408.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig
- 8.408.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta
- 8.408.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThreshold
- 8.408.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta
- 8.408.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh
- 8.408.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta
- 8.408.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh
- 8.408.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta
- 8.408.2.29 nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh
- 8.408.2.30 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

8.408.2.31 `nas_TDSCDMASINRThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh`

8.408.2.32 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIODelta`

8.408.2.33 `nas_WCDMAECIOThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIOThresh`

8.408.2.34 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta`

8.408.2.35 `nas_WCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh`

8.409 `pack_nas_SLQSNasGet3GPP2Subscription_t` Struct Reference

Data Fields

- `uint8_t` [namID](#)

8.409.1 Detailed Description

Structure for 3GPP2 Subscription pack.

Parameters

<i>namID</i>	[Mandatory] <ul style="list-style-type: none"> • NAM ID of the information to be retrieved. The index starts from 0. A <code>nam_id</code> of 0xFF is used to retrieve information of current NAM.
--------------	---

8.409.2 Field Documentation

8.409.2.1 `uint8_t` `pack_nas_SLQSNasGet3GPP2Subscription_t::namID`

8.410 `pack_nas_SLQSNASGeteDRXParamsExt_t` Struct Reference

Data Fields

- `uint8_t *` [pEdrxRAT](#)
- `uint32_t *` [pLteOpMode](#)

8.410.1 Detailed Description

This structure contains the SLQSNASGeteDRXParamsExt request parameters.

Parameters

<i>pEdrxRAT</i>	[Optional] <ul style="list-style-type: none"> eDRX Radio Access Technology Values <ul style="list-style-type: none"> NAS_RADIO_IF_NO_SVC (0x00) - None (no service) NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X NAS_RADIO_IF_CDMA_1XEVD0 (0x02) - cdma2000 @ HRPD (1xEV-DO) NAS_RADIO_IF_AMPS (0x03) - AMPS NAS_RADIO_IF_GSM (0x04) - GSM NAS_RADIO_IF_UMTS (0x05) - UMTS NAS_RADIO_IF_WLAN (0x06) - WLAN NAS_RADIO_IF_GPS (0x07) - GPS NAS_RADIO_IF_LTE (0x08) - LTE NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1 NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1 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.
<i>pLteOpMode</i>	[Optional] <ul style="list-style-type: none"> LTE Operational Mode Values <ul style="list-style-type: none"> NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1 NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1 Note: This TLV should be set when edrx_rat_type is LTE.

8.410.2 Field Documentation

8.410.2.1 uint8_t* pack_nas_SLQSNASGetDRXParamsExt_t::pEdrxRAT

8.410.2.2 uint32_t* pack_nas_SLQSNASGetDRXParamsExt_t::pLteOpMode

8.411 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- uint8_t [radio_if](#)

8.411.1 Detailed Description

Structure for TxRx Info pack

Parameters

<i>radio_if</i>	[Mandatory] <ul style="list-style-type: none"> Radio interface technology of the signal being measured Valid Values <ul style="list-style-type: none"> 0x01 - NAS_RADIO_IF_CDMA_1X - CDMA 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - HDR 0x04 - NAS_RADIO_IF_GSM - GSM 0x05 - NAS_RADIO_IF_UMTS - UMTS 0x08 - NAS_RADIO_IF_LTE - LTE
-----------------	---

8.411.2 Field Documentation

8.411.2.1 uint8_t pack_nas_SLQSNasGetTxRxInfo_t::radio_if

8.412 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](#)

8.412.1 Detailed Description

This structure contains pack indication register parameters.

Parameters

<i>pSystem-SelectionInd</i>	[Optional] <ul style="list-style-type: none"> System Selection Preference indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable
<i>pDDTMInd</i>	[Optional] <ul style="list-style-type: none"> DDTM (Data Dedicated Transmission Mode) indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable

<i>pServing-SystemInd</i>	[Optional] <ul style="list-style-type: none"> Serving System indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pDualStandBy-PrefInd</i>	[Optional] <ul style="list-style-type: none"> Dual Standby Preference indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pSubscription-InfoInd</i>	[Optional] <ul style="list-style-type: none"> Subscription Information indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pNetworkTime-Ind</i>	[Optional] <ul style="list-style-type: none"> Network Time indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pSysInfoInd</i>	[Optional] <ul style="list-style-type: none"> System Information indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pSignalStrength-Ind</i>	[Optional] <ul style="list-style-type: none"> Signal Strength indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pErrorRateInd</i>	[Optional] <ul style="list-style-type: none"> Error Rate indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pHDRNewUATI-AssInd</i>	[Optional] <ul style="list-style-type: none"> HDR New UATI Assigned indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable

<i>pHDRSessionCloseInd</i>	[Optional] <ul style="list-style-type: none"> HDR Session Closed indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable
<i>pManagedRoamingInd</i>	[Optional] <ul style="list-style-type: none"> Managed Roaming indication registration. The callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable
<i>pNetworkRejectInd</i>	[Optional] <ul style="list-style-type: none"> Network reject information indication registration. The following callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable
<i>pSuppressSysInfoInd</i>	[Optional] <ul style="list-style-type: none"> Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values: <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable- 0x01 - Enable
<i>pLTECphyCa</i>	[Optional] <ul style="list-style-type: none"> LTE Physical Carrier Aggregation Information. The following callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable (default value) 0x01 - Enable
<i>pEdrxChangeInfoInd</i>	[Optional] <ul style="list-style-type: none"> EDRX Change Info indication registration. The following callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable

8.412.2 Field Documentation

8.412.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.412.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.412.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pEdrxChangeInfoInd

8.412.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.412.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.412.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.412.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.412.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.412.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd

8.412.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.412.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.412.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.412.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.412.2.14 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd

8.412.2.15 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.412.2.16 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.413 pack_nas_SLQSNASSeteDRXParams_t Struct Reference

Data Fields

- uint8_t * [pEdrxEnable](#)
- uint8_t * [pCycleLen](#)
- uint8_t * [pPagingTimeWindow](#)
- uint8_t * [pEdrxRatType](#)
- uint32_t * [pEdrxCiotLteMode](#)

8.413.1 Detailed Description

This structure contains the SLQSNASSeteDRXParams request parameters.

Parameters

<i>pEdrxEnable</i>	[Optional] <ul style="list-style-type: none"> • eDRX Enable • Values <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pCycleLen</i>	[Optional] <ul style="list-style-type: none"> • eDRX cycle length • Value to be set per 3GPP TS 24.008 Rel-13 Section 10.5.5.32
<i>pPagingTime-Window</i>	[Optional] <ul style="list-style-type: none"> • eDRX paging time window • Value to be set per 3GPP TS 24.008 Rel-13 Section 10.5.5.32

<i>pEdrxRatType</i>	[Optional] <ul style="list-style-type: none"> • Radio access technology • Values <ul style="list-style-type: none"> – NAS_RADIO_IF_NO_SVC (0x00) - None (no service) – NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X – NAS_RADIO_IF_CDMA_1XEVD0 (0x02) cdma2000 @ HRPD (1xEV-DO) – NAS_RADIO_IF_AMPS (0x03) - AMPS – NAS_RADIO_IF_GSM (0x04) - GSM – NAS_RADIO_IF_UMTS (0x05) - UMTS – NAS_RADIO_IF_WLAN (0x06) - WLAN – NAS_RADIO_IF_GPS (0x07) - GPS – NAS_RADIO_IF_LTE (0x08) - LTE – NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA – NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1 – NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1 – 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.
<i>pEdrxCiotLteMode</i>	[Optional] <ul style="list-style-type: none"> • CIOT LTE mode • Values <ul style="list-style-type: none"> – NAS_CIoT_SYS_MODE_NO_SRV(0x00) - No service – NAS_CIoT_SYS_MODE_LTE_WB(0x01) - Camped on LTE wideband – NAS_CIoT_SYS_MODE_LTE_M1(0x02) - Camped on LTE M1 – NAS_CIoT_SYS_MODE_LTE_NB1(0x03) - Camped on LTE NB1 Note: This TLV should be set when pEdrxRatType is LTE.

8.413.2 Field Documentation

8.413.2.1 `uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pCycleLen`

8.413.2.2 `uint32_t* pack_nas_SLQSNASSeteDRXParams_t::pEdrxCiotLteMode`

8.413.2.3 `uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pEdrxEnable`

8.413.2.4 `uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pEdrxRatType`

8.413.2.5 `uint8_t* pack_nas_SLQSNASSeteDRXParams_t::pPagingTimeWindow`

8.414 `pack_nas_SLQSNasSwilIndicationRegister_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](#)

8.414.1 Detailed Description

This structure contains the OTA message indication.

Parameters

<i>lteEsmUI</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report LTE ESM uplink messages
<i>lteEsmDI</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report LTE ESM downlink messages
<i>lteEmmUI</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report LTE EMM uplink messages
<i>lteEmmDI</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report GSM/UMTS uplink messages
<i>gsmUmtsUI</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report GSM/UMTS uplink messages
<i>gsmUmtsDI</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report GSM/UMTS downlink messages
<i>pRankIndicator-Ind</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report Rank Indicator messages
<i>pTimer</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report Timer Indicator messages

8.414.2 Field Documentation

8.414.2.1 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDI

8.414.2.2 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUI

8.414.2.3 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDI

8.414.2.4 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUI

8.414.2.5 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDI

8.414.2.6 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUI`

8.414.2.7 `uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd`

8.414.2.8 `uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pTimer`

8.415 `pack_nas_SLQSNASSwiSetChannelLock_t` Struct Reference

Data Fields

- `nas_wcdmaUARFCN` * `pWcdmaUARFCN`
- `nas_lteEARFCN` * `pLteEARFCN`
- `nas_ltePCI` * `pLtePCI`

8.415.1 Detailed Description

Structure for ChannelLock pack.

Parameters

<code>pWcdmaUARFCN</code>	[Optional] • See nas_wcdmaUARFCN for more information
<code>pLteEARFCN</code>	[Optional] • See nas_lteEARFCN for more information
<code>pLtePCI</code>	[Optional] • See nas_ltePCI for more information

8.415.2 Field Documentation

8.415.2.1 `nas_lteEARFCN*` `pack_nas_SLQSNASSwiSetChannelLock_t::pLteEARFCN`

8.415.2.2 `nas_ltePCI*` `pack_nas_SLQSNASSwiSetChannelLock_t::pLtePCI`

8.415.2.3 `nas_wcdmaUARFCN*` `pack_nas_SLQSNASSwiSetChannelLock_t::pWcdmaUARFCN`

8.416 `pack_nas_SLQSSetBandPreference_t` Struct Reference

Data Fields

- `uint64_t` `bandPref`

8.416.1 Detailed Description

Provides information about the band preference.

Parameters

<i>bandpreference</i>	<ul style="list-style-type: none"> • Bit mask representing the band preference to be set. • Bit position meanings: <ul style="list-style-type: none"> – 0 - BC0_A - Band Class 0, A-System – 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band – 2 - BC1 - Band Class 1, all blocks – 3 - BC2 - Band Class 2 place holder – 4 - BC3 - Band Class 3, A-System – 5 - BC4 - Band Class 4, all blocks – 6 - BC5 - Band Class 5, all blocks – 7 - GSM_DCS_1800 - GSM DCS band – 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band – 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band – 10 - BC6 - Band Class 6 – 11 - BC7 - Band Class 7 – 12 - BC8 - Band Class 8 – 13 - BC9 - Band Class 9 – 14 - BC10 - Band Class 10 – 15 - BC11 - Band Class 11 – 16 - GSM_450 - GSM 450 band – 17 - GSM_480 - GSM 480 band – 18 - GSM_750 - GSM 750 band – 19 - GSM_850 - GSM 850 band – 20 - GSM_RGSM_900 - GSM Railways GSM Band – 21 - GSM_PCS_1900 - GSM PCS band – 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band – 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band – 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band – 25 - WCDMA_IV_1700 - WCDMA US 1700 band – 26 - WCDMA_V_850 - WCDMA US 850 band – 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band – 28 - BC12 - Band Class 12 – 29 - BC14 - Band Class 14 – 30 - RESERVED_2 - Reserved 2 – 31 - BC15 - Band Class 15 – 32 - 47 - Reserved – 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band – 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band – 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band – 51 to 55 - Reserved – 56 - BBC16 - Band Class 16 – 57 - BC17 - Band Class 17 – 58 - BC18 - Band Class 18 – 59 - BC19 - Band Class 19 – 60 to 64 - Reserved
-----------------------	---

8.416.2 Field Documentation

8.416.2.1 uint64_t pack_nas_SLQSSetBandPreference_t::bandPref

8.417 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

Data Fields

- uint8_t bEnable
- nas_SLQSSignalStrengthsIndReq * pSigIndReq

8.417.1 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

8.417.2 Field Documentation

8.417.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable

8.417.2.2 nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq

8.418 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

8.418.1 Detailed Description

Contain the system selection preferences.

Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none">• Optional parameter specifying the emergency Mode• Values:<ul style="list-style-type: none">– 0 - OFF (normal)– 1 - ON (Emergency)
<i>pModePref</i>	<ul style="list-style-type: none">• Optional parameter• Bit Mask indicating the radio technology mode preference• Bit values:<ul style="list-style-type: none">– Bit 0 - cdma2000 1x– Bit 1 - cdma2000 HRPD(1xEV-DO)– Bit 2 - GSM– Bit 3 - UMTS– Bit 4 - LTE

<i>pBandPref</i>	<ul style="list-style-type: none"> • Optional parameter • Bit mask representing the band preference • Bit values: <ul style="list-style-type: none"> – Bit 0 - Band Class 0, A-System – Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band – Bit 2 - Band Class 1, all blocks – Bit 3 - Band Class 2 place holder – Bit 4 - Band Class 3, A-System – Bit 5 - Band Class 4, all blocks – Bit 6 - Band Class 5, all blocks – Bit 7 - GSM_DCS_1800 band – Bit 8 - GSM Extended GSM (E-GSM) 900 band – Bit 9 - GSM Primary GSM (P-GSM) 900 band – Bit 10 - Band Class 6 – Bit 11 - Band Class 7 – Bit 12 - Band Class 8 – Bit 13 - Band Class 9 – Bit 14 - Band Class 10 – Bit 15 - Band Class 11 – Bit 16 - GSM 450 band – Bit 17 - GSM 480 band – Bit 18 - GSM 750 band – Bit 19 - GSM 850 band – Bit 20 - GSM Railways GSM 900 Band – Bit 21 - GSM PCS 1900 band – Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band – Bit 23 - WCDMA U.S. PCS 1900 band – Bit 24 - WCDMA Europe and China DCS 1800 band – Bit 25 - WCDMA U.S. 1700 band – Bit 26 - WCDMA U.S. 850 band – Bit 27 - WCDMA Japan 800 band – Bit 28 - Band Class 12 – Bit 29 - Band Class 14 – Bit 30 - Reserved – Bit 31 - Band Class 15 – Bit 32 to 47 - Reserved – Bit 48 - WCDMA Europe 2600 band – Bit 49 - WCDMA Europe and Japan 900 band – Bit 50 - WCDMA Japan 1700 band – Bit 51 to 55 - Reserved – Bit 56 - Band Class 16 – Bit 57 - Band Class 17 – Bit 58 - Band Class 18 – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved
------------------	--

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

<i>pNetSelPref</i>	<ul style="list-style-type: none"> - netSelectionPref • Optional parameter for specifying Network Selection Preference • Modem selects networks based on this parameter(if present). • see nas_netSelectionPref for more information
<i>pChgDuration</i>	<ul style="list-style-type: none"> • Optional parameter specifying the duration of the change • Values: <ul style="list-style-type: none"> – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by client – Device will use "0x01 - permanent" as default if this parameter is omitted
<i>pMNCIncPCS-DigStat</i>	<ul style="list-style-type: none"> • Optional parameter indicating if MNC includes PCS digit • Values: <ul style="list-style-type: none"> – TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating Service domain preference • Values: <ul style="list-style-type: none"> – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> • Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: <ul style="list-style-type: none"> – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM
<i>pTdscdmaBand-Pref</i>	<ul style="list-style-type: none"> • Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set. • Values: <ul style="list-style-type: none"> – 0x01 - TD-SCDMA Band A – 0x02 - TD-SCDMA Band B – 0x04 - TD-SCDMA Band C – 0x08 - TD-SCDMA Band D – 0x10 - TD-SCDMA Band E – 0x20 - TD-SCDMA Band F – All other bits are reserved
<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> - acqOrderPref • Optional parameter for specifying Acquisition Order Preference • see nas_acqOrderPref for more information

<i>pSrvReg-Restriction</i>	<ul style="list-style-type: none"> Optional parameter indicating Network Selection Registration Restriction Preference Values: <ul style="list-style-type: none"> 0x00 - Device follows the normal registration process 0x01 - Device camps on the network according to its provisioning, but does not register 0x02 - Device selects the network for limited service All other values are reserved.
<i>pCSGID</i>	<ul style="list-style-type: none"> - CSGID Optional parameter for specifying CSG ID Either of pNetSelPref or pCSGID can be set. see nas_CSGID for more information
<i>pRAT</i>	<ul style="list-style-type: none"> Optional parameter Radio Access Technology order Preference Values: <ul style="list-style-type: none"> 0x04 - GSM 0x05 - UMTS 0x08 - LTE 0x09 - TDSCDMA

8.418.2 Field Documentation

8.418.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.418.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

8.418.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

8.418.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

8.418.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

8.418.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

8.418.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref

8.418.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat

8.418.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref

8.418.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref

8.418.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref

8.418.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT

8.418.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref

8.418.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref

8.418.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction

8.418.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdsdmaBandPref

8.419 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

8.419.1 Detailed Description

Contain the system selection preferences.

Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none"> • Optional parameter specifying the emergency Mode • Values: <ul style="list-style-type: none"> – 0 - OFF (normal) – 1 - ON (Emergency)
------------------	---

<i>pModePref</i>	<ul style="list-style-type: none">• Optional parameter• Bit Mask indicating the radio technology mode preference• Bit values:<ul style="list-style-type: none">– Bit 0 - cdma2000 1x– Bit 1 - cdma2000 HRPD(1xEV-DO)– Bit 2 - GSM– Bit 3 - UMTS– Bit 4 - LTE– Bit 5 - TD-SCDMA– Bit 6 - NR5G
------------------	--

<i>pBandPref</i>	<ul style="list-style-type: none"> • Optional parameter • Bit mask representing the band preference • Bit values: <ul style="list-style-type: none"> – Bit 0 - Band Class 0, A-System – Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band – Bit 2 - Band Class 1, all blocks – Bit 3 - Band Class 2 place holder – Bit 4 - Band Class 3, A-System – Bit 5 - Band Class 4, all blocks – Bit 6 - Band Class 5, all blocks – Bit 7 - GSM_DCS_1800 band – Bit 8 - GSM Extended GSM (E-GSM) 900 band – Bit 9 - GSM Primary GSM (P-GSM) 900 band – Bit 10 - Band Class 6 – Bit 11 - Band Class 7 – Bit 12 - Band Class 8 – Bit 13 - Band Class 9 – Bit 14 - Band Class 10 – Bit 15 - Band Class 11 – Bit 16 - GSM 450 band – Bit 17 - GSM 480 band – Bit 18 - GSM 750 band – Bit 19 - GSM 850 band – Bit 20 - GSM Railways GSM 900 Band – Bit 21 - GSM PCS 1900 band – Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band – Bit 23 - WCDMA U.S. PCS 1900 band – Bit 24 - WCDMA Europe and China DCS 1800 band – Bit 25 - WCDMA U.S. 1700 band – Bit 26 - WCDMA U.S. 850 band – Bit 27 - WCDMA Japan 800 band – Bit 28 - Band Class 12 – Bit 29 - Band Class 14 – Bit 30 - Reserved – Bit 31 - Band Class 15 – Bit 32 to 47 - Reserved – Bit 48 - WCDMA Europe 2600 band – Bit 49 - WCDMA Europe and Japan 900 band – Bit 50 - WCDMA Japan 1700 band – Bit 51 to 55 - Reserved – Bit 56 - Band Class 16 – Bit 57 - Band Class 17 – Bit 58 - Band Class 18 – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved
------------------	--

<i>pPRLPref</i>	<ul style="list-style-type: none">• Optional parameter indicating the CDMA PRL Preference• Values:<ul style="list-style-type: none">– 0x0001 - Acquire available system only on the A side– 0x0002 - Acquire available system only on the B side– 0x3FFF - Acquire any available systems
<i>pRoamPref</i>	<ul style="list-style-type: none">• Optional parameter indicating the roaming Preference• Values:<ul style="list-style-type: none">– 0x01 - Acquire only systems for which the roaming indicator is off– 0x02 - Acquire a system as long as its roaming indicator is not off– 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only– 0xFF - Acquire systems, regardless of their roaming indicator

<i>pLTEBandPref</i>	<ul style="list-style-type: none"> • Optional parameter • Bit mask representing the LTE band preference • Bit Values <ul style="list-style-type: none"> – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – All other bits are reserved
---------------------	--

<i>pNetSelPref</i>	<ul style="list-style-type: none"> - netSelectionPref • Optional parameter for specifying Network Selection Preference • Modem selects networks based on this parameter(if present). • Either of pNetSelPref or pCSGID can be set. • see nas_netSelectionPref for more information
<i>pChgDuration</i>	<ul style="list-style-type: none"> • Optional parameter specifying the duration of the change • At least one system selection setting to be set if pChgDuration is populated. • Values: <ul style="list-style-type: none"> – 0x00 - Power cycle - Remains active until the next device power cycle – 0x01 - Permanent - Remains active through power cycles until changed by client – Device will use "0x01 - permanent" as default if this parameter is omitted
<i>pMNCIncPCS-DigStat</i>	<ul style="list-style-type: none"> • Optional parameter indicating if MNC includes PCS digit • pNetSelPref is expected if MNC includes PCS digit is set to 1. • Values: <ul style="list-style-type: none"> – TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090 – FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating Service domain preference • Values: <ul style="list-style-type: none"> – 0x00 - Circuit switched only – 0x01 - Packet switched only – 0x02 - Circuit switched and packet switched – 0x03 - Packet switched attach – 0x04 - Packet switched detach
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> • Optional parameter indicating GSM/WCDMA Acquisition order Preference • Values: <ul style="list-style-type: none"> – 0x00 - Automatic – 0x01 - GSM then WCDMA – 0x02 - WCDMA then GSM
<i>pTdsdmaBand-Pref</i>	<ul style="list-style-type: none"> • Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set. • Values: <ul style="list-style-type: none"> – 0x01 - TD-SCDMA Band A – 0x02 - TD-SCDMA Band B – 0x04 - TD-SCDMA Band C – 0x08 - TD-SCDMA Band D – 0x10 - TD-SCDMA Band E – 0x20 - TD-SCDMA Band F – All other bits are reserved

<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> - acqOrderPref • Optional parameter for specifying Acquisition Order Preference • see nas_acqOrderPref for more information
<i>pSrvReg-Restriction</i>	<ul style="list-style-type: none"> • Optional parameter indicating Network Selection Registration Restriction Preference • Values: <ul style="list-style-type: none"> – 0x00 - Device follows the normal registration process – 0x01 - Device camps on the network according to its provisioning, but does not register – 0x02 - Device selects the network for limited service – All other values are reserved.
<i>pCSGID</i>	<ul style="list-style-type: none"> - CSGID • Optional parameter for specifying CSG ID • Either of pNetSelPref or pCSGID can be set. • see nas_CSGID for more information
<i>pRAT</i>	<ul style="list-style-type: none"> • Optional parameter Radio Access Technology order Preference • Values: <ul style="list-style-type: none"> – 0x04 - GSM – 0x05 - UMTS – 0x08 - LTE – 0x09 - TDSCDMA – 0x0A - LTE-M1 – 0x0B - LTE-NB1 – 0x0C - NR5G
<i>pLTEBandPref-Ext[IN]</i>	<ul style="list-style-type: none"> • LTE Band Preference Extended • see nas_lteBandPrefExt for more information
<i>pCiotLteOp-ModePref[IN]</i>	<ul style="list-style-type: none"> • CIOT LTE Operational Mode Preference • Bitmask representing the ClotT LTE operational mode preference to be set. • Values : <ul style="list-style-type: none"> – Bit 0 - LTE wideband – Bit 1 - LTE M1 – 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.

<p><i>pLteM1Band-Pref</i>[IN]</p>	<ul style="list-style-type: none"> • Bitmask representing the LTE M1 band preference to be set. • Bit Values <ul style="list-style-type: none"> – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – Bit 61 - E-UTRA Operating Band 126 – Bit 62 - E-UTRA Operating Band 127
-----------------------------------	--

- 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"> • Bitmask representing the LTE NB1 band preference to be set. • Bit Values <ul style="list-style-type: none"> – Bit 0 - E-UTRA Operating Band 1 – Bit 1 - E-UTRA Operating Band 2 – Bit 2 - E-UTRA Operating Band 3 – Bit 3 - E-UTRA Operating Band 4 – Bit 4 - E-UTRA Operating Band 5 – Bit 5 - E-UTRA Operating Band 6 – Bit 6 - E-UTRA Operating Band 7 – Bit 7 - E-UTRA Operating Band 8 – Bit 8 - E-UTRA Operating Band 9 – Bit 9 - E-UTRA Operating Band 10 – Bit 10 - E-UTRA Operating Band 11 – Bit 11 - E-UTRA Operating Band 12 – Bit 12 - E-UTRA Operating Band 13 – Bit 13 - E-UTRA Operating Band 14 – Bit 16 - E-UTRA Operating Band 17 – Bit 17 - E-UTRA Operating Band 18 – Bit 18 - E-UTRA Operating Band 19 – Bit 19 - E-UTRA Operating Band 20 – Bit 20 - E-UTRA Operating Band 21 – Bit 22 - E-UTRA Operating Band 23 – Bit 23 - E-UTRA Operating Band 24 – Bit 24 - E-UTRA Operating Band 25 – Bit 25 - E-UTRA Operating Band 26 – Bit 27 - E-UTRA Operating Band 28 – Bit 28 - E-UTRA Operating Band 29 – Bit 29 - E-UTRA Operating Band 32 – Bit 32 - E-UTRA Operating Band 33 – Bit 33 - E-UTRA Operating Band 34 – Bit 34 - E-UTRA Operating Band 35 – Bit 35 - E-UTRA Operating Band 36 – Bit 36 - E-UTRA Operating Band 37 – Bit 37 - E-UTRA Operating Band 38 – Bit 38 - E-UTRA Operating Band 39 – Bit 39 - E-UTRA Operating Band 40 – Bit 40 - E-UTRA Operating Band 41 – Bit 41 - E-UTRA Operating Band 42 – Bit 42 - E-UTRA Operating Band 43 – Bit 60 - E-UTRA Operating Band 125 – Bit 61 - E-UTRA Operating Band 126 – Bit 62 - E-UTRA Operating Band 127 • If this field is not present, the modem will use the NB1 band preference value read from the NV.
------------------------------------	---

<i>pCiotLteOpMode</i>	<ul style="list-style-type: none"> • This TLV is used when the net_sel_pref is MANUAL and RAT TLV is LTE. • This TLV indicates the the operational mode to be used during the manual LTE PLMN selection. • Values : <ul style="list-style-type: none"> – 0x00 - No service – 0x01 - Camped on LTE wideband – 0x02 - Camped on LTE M1 – 0x03 - Camped on LTE NB1
<i>pCiotAcqOrderPref</i> [IN]	<ul style="list-style-type: none"> • CIOT Acquisition Order Preference • see nas_ciotAcqOrderPref for more information
<i>pNr5gBandPref</i> [IN]	<ul style="list-style-type: none"> • NR5G Band Preference • see nas_nr5gBandPref for more information

8.419.2 Field Documentation

- 8.419.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pAcqOrderPref
- 8.419.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pBandPref
- 8.419.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pChgDuration
- 8.419.2.4 struct nas_ciotAcqOrderPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotAcqOrderPref
- 8.419.2.5 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotLteOpMode
- 8.419.2.6 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotLteOpModePref
- 8.419.2.7 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPrefExt_t::pCSGID
- 8.419.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pEmerMode
- 8.419.2.9 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pGWAcqOrderPref
- 8.419.2.10 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLTEBandPref
- 8.419.2.11 struct nas_lteBandPrefExt* pack_nas_SLQSSetSysSelectionPrefExt_t::pLTEBandPrefExt
- 8.419.2.12 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLteM1BandPref
- 8.419.2.13 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLteNb1BandPref
- 8.419.2.14 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pMNCIncPCSDigStat
- 8.419.2.15 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pModePref
- 8.419.2.16 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pNetSelPref
- 8.419.2.17 struct nas_nr5gBandPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pNr5gBandPref

8.419.2.18 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pPRLPref

8.419.2.19 unsigned char* pack_nas_SLQSSetSysSelectionPrefExt_t::pRAT

8.419.2.20 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pRoamPref

8.419.2.21 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pSrvDomainPref

8.419.2.22 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pSrvRegRestriction

8.419.2.23 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pTdsdmaBandPref

8.420 pack_nas_SLQSSwiPSDetach_t Struct Reference

Data Fields

- uint8_t * [pDetachAction](#)

8.420.1 Detailed Description

Structure for PS connection detach pack.

Parameters

<i>pDetachAction</i> [1- N]	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 2- Initiates an immediate packet domain detach.
--------------------------------	---

8.420.2 Field Documentation

8.420.2.1 uint8_t* pack_nas_SLQSSwiPSDetach_t::pDetachAction

8.421 pack_pds_PDSInjectTimeReference_t Struct Reference

Data Fields

- uint64_t [systemTime](#)
- uint16_t [systemDiscontinuities](#)

8.421.1 Detailed Description

Structure contain the parameter for PDS Inject time.

Parameters

<i>systemTime</i>	<ul style="list-style-type: none"> • System time(milliseconds)
<i>system-Discontinuities</i>	<ul style="list-style-type: none"> • Number of system time discontinuities

8.421.2 Field Documentation

8.421.2.1 uint16_t pack_pds_PDSInjectTimeReference_t::systemDiscontinuities

8.421.2.2 uint64_t pack_pds_PDSInjectTimeReference_t::systemTime

8.422 pack_pds_ResetPDSData_t Struct Reference

Data Fields

- uint32_t * [pGPSDataMask](#)
- uint32_t * [pCellDataMask](#)

8.422.1 Detailed Description

Structure contain the parameter to Resets the specified PDS data.

Parameters

<i>pGPSDataMask</i>	<ul style="list-style-type: none"> • Bitmask of GPS data to clear (optional) <ul style="list-style-type: none"> – 0x00000001 - EPH – 0x00000002 - ALM – 0x00000004 - POS – 0x00000008 - TIME – 0x00000010 - IONO – 0x00000020 - UTC – 0x00000040 - HEALTH – 0x00000080 - SVDIR – 0x00000100 - SVSTEER – 0x00000200 - SADATA – 0x00000400 - RTI – 0x00000800 - ALM_CORR – 0x00001000 - FREQ_BIAS_EST
<i>pCellDataMask</i>	<ul style="list-style-type: none"> • Bitmask of cell data to clear (optional) <ul style="list-style-type: none"> – 0x00000001 - POS – 0x00000002 - LATEST_GPS_POS – 0x00000004 - OTA_POS – 0x00000008 - EXT_REF_POS – 0x00000010 - TIMETAG – 0x00000020 - CELLID – 0x00000040 - CACHED_CELLID – 0x00000080 - LAST_SRV_CELL – 0x00000100 - CUR_SRV_CELL – 0x00000200 - NEIGHBOR_INFO

8.422.2 Field Documentation

8.422.2.1 uint32_t* pack_pds_ResetPDSData_t::pCellDataMask

8.422.2.2 uint32_t* pack_pds_ResetPDSData_t::pGPSDataMask

8.423 pack_pds_SetEventReportCallback_t Struct Reference

Data Fields

- uint8_t [posDataNmea](#)
- uint8_t [rptPosData](#)

8.423.1 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"> • Report new position data in NMEA format. • Values <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>rptPosData</i>	<ul style="list-style-type: none"> • Report new position data in raw format. • Values <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable

8.423.2 Field Documentation

8.423.2.1 uint8_t pack_pds_SetEventReportCallback_t::posDataNmea

8.423.2.2 uint8_t pack_pds_SetEventReportCallback_t::rptPosData

8.424 pack_pds_SetPDSDefaults_t Struct Reference

Data Fields

- uint32_t [operation](#)
- uint8_t [timeout](#)
- uint32_t [interval](#)
- uint32_t [accuracy](#)

8.424.1 Detailed Description

Structure containing session configuration parameters.

Parameters

<i>operation</i>	<ul style="list-style-type: none"> Current session operating mode <ul style="list-style-type: none"> 0 - Standalone 1 - MS based 2 - MS assisted
<i>timeout</i>	<ul style="list-style-type: none"> Maximum amount of time (seconds) to work on each fix, maximum is 255
<i>interval</i>	<ul style="list-style-type: none"> Interval (seconds) between fix requests
<i>accuracy</i>	<ul style="list-style-type: none"> Preferred accuracy threshold (meters)

8.424.2 Field Documentation

8.424.2.1 uint32_t pack_pds_SetPDSDefaults_t::accuracy

8.424.2.2 uint32_t pack_pds_SetPDSDefaults_t::interval

8.424.2.3 uint32_t pack_pds_SetPDSDefaults_t::operation

8.424.2.4 uint8_t pack_pds_SetPDSDefaults_t::timeout

8.425 pack_pds_SetPDSSState_t Struct Reference

Data Fields

- uint32_t [enable](#)

8.425.1 Detailed Description

Structure contain PDS state parameters.

Parameters

<i>enable</i>	<ul style="list-style-type: none"> Desired PDS state <ul style="list-style-type: none"> Zero - disable Non-Zero - enable
---------------	--

8.425.2 Field Documentation

8.425.2.1 uint32_t pack_pds_SetPDSSState_t::enable

8.426 pack_pds_SetPortAutomaticTracking_t Struct Reference

Data Fields

- uint32_t [bAuto](#)

8.426.1 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"> • Enable automatic tracking for NMEA COM port <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled
--------------	--

8.426.2 Field Documentation

8.426.2.1 uint32_t pack_pds_SetPortAutomaticTracking_t::bAuto

8.427 pack_pds_SetServiceAutomaticTracking_t Struct Reference

Data Fields

- uint32_t [bAuto](#)

8.427.1 Detailed Description

Structure contain the parameter to Sets the automatic tracking state.

Parameters

<i>bAuto</i>	<ul style="list-style-type: none"> • Automatic tracking session started for service <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled
--------------	---

8.427.2 Field Documentation

8.427.2.1 uint32_t pack_pds_SetServiceAutomaticTracking_t::bAuto

8.428 pack_pds_SetXTRAAutomaticDownload_t Struct Reference

Data Fields

- uint32_t [bEnabled](#)
- uint16_t [interval](#)

8.428.1 Detailed Description

Structure contain the parameter to set the XTRA automatic database download configuration.

Parameters

<i>bEnabled</i>	<ul style="list-style-type: none"> Automatic XTRA download status <ul style="list-style-type: none"> 0 - Disabled 1 - Enabled
<i>interval</i>	<ul style="list-style-type: none"> Interval (hours) between XTRA downloads

8.428.2 Field Documentation

8.428.2.1 uint32_t pack_pds_SetXTRAAutomaticDownload_t::bEnabled

8.428.2.2 uint16_t pack_pds_SetXTRAAutomaticDownload_t::interval

8.429 pack_pds_SetXTRANetwork_t Struct Reference

Data Fields

- uint32_t [preference](#)

8.429.1 Detailed Description

Structure contain the parameter to Sets the XTRA WWAN network preference

Parameters

<i>preference</i>	<ul style="list-style-type: none"> XTRA WWAN network preference <ul style="list-style-type: none"> 0x00 - None (any available network) 0x01 - Home-only, only when on home systems 0x02 - Roam-only, only when on non-home systems
-------------------	---

8.429.2 Field Documentation

8.429.2.1 uint32_t pack_pds_SetXTRANetwork_t::preference

8.430 pack_pds_SLQSGetAGPSConfig_t Struct Reference

Data Fields

- uint8_t * [pNetworkMode](#)

8.430.1 Detailed Description

Structure contain the parameter to PDS AGPS (MS-based) configuration pack.

Parameters

<i>pNetworkMode</i>	<ul style="list-style-type: none"> • Network Mode of AGPS Server [optional - should be present in Multimode Systems] <ul style="list-style-type: none"> – 0x00 - UMTS – 0x01 - CDMA
---------------------	---

8.430.2 Field Documentation

8.430.2.1 `uint8_t* pack_pds_SLQSGetAGPSConfig_t::pNetworkMode`

8.431 `pack_pds_SLQSPDSInjectAbsoluteTimeReference_t` Struct Reference

Data Fields

- `uint64_t timeMsec`
- `uint32_t timeUncMsec`
- `uint8_t timeBase`
- `uint8_t forceFlag`

8.431.1 Detailed Description

Structure contain the parameter to Injects a absolute time reference.

Parameters

<i>timeMsec</i>	<ul style="list-style-type: none"> • 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
<i>timeUncMsec</i>	<ul style="list-style-type: none"> • Time uncertainty in milliseconds
<i>timeBase</i>	<ul style="list-style-type: none"> • Time base <ul style="list-style-type: none"> – 0x00 - GPS (midnight, Jan 6, 1980) – 0x01 - UTC (midnight, Jan 1, 1970)
<i>forceFlag</i>	<ul style="list-style-type: none"> • Force acceptance of data

8.431.2 Field Documentation

8.431.2.1 `uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::forceFlag`

8.431.2.2 `uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeBase`

8.431.2.3 uint64_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeMsec

8.431.2.4 uint32_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeUncMsec

8.432 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](#)

8.432.1 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"> • 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
<i>pLatitude</i>	<ul style="list-style-type: none"> • 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)
<i>pLongitude</i>	<ul style="list-style-type: none"> • 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.
<i>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> • 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)
<i>pAltitudeWrt-Sealevel</i>	<ul style="list-style-type: none"> • 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)
<i>pHorizontalUnc-Circular</i>	<ul style="list-style-type: none"> • 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)

<i>pVerticalUnc</i>	<ul style="list-style-type: none"> 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)
<i>pHorizontal-Confidence</i>	<ul style="list-style-type: none"> 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.
<i>pVertical-Confidence</i>	<ul style="list-style-type: none"> 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.
<i>pPositionSource</i>	<ul style="list-style-type: none"> Source of injected position: <ul style="list-style-type: none"> 0x00 - Unknown 0x01 - GPS 0x02 - Cell ID 0x03 - Enhanced cell ID 0x04 - WiFi 0x05 - Terrestrial 0x06 - Terrestrial hybrid 0x07 - Other
<i>pTimeType</i>	<ul style="list-style-type: none"> Defines the time value set in the pTimeStamp parameter. <ul style="list-style-type: none"> 0x00 - UTC Time: starting Jan 1, 1970 0x01 - GPS Time: starting Jan 6, 1980 0x02 - Age: Age of position information

8.432.2 Field Documentation

8.432.2.1 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtEllipsoid

8.432.2.2 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtSealevel

8.432.2.3 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalConfidence

8.432.2.4 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalUncCircular

8.432.2.5 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLatitude

8.432.2.6 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLongitude

8.432.2.7 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pPositionSource

8.432.2.8 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeStamp

8.432.2.9 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeType

8.432.2.10 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalConfidence

8.432.2.11 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalUnc

8.433 pack_pds_SLQSSetAGPSConfig_t Struct Reference

Data Fields

- uint32_t * [pServerAddress](#)
- uint32_t * [pServerPort](#)
- uint8_t * [pServerURL](#)
- uint8_t * [pServerURLLength](#)
- uint8_t * [pNetworkMode](#)

8.433.1 Detailed Description

Structure contain the parameter to Sets the PDS AGPS (MS-based) configuration.

Parameters

<i>pServerAddress</i>	<ul style="list-style-type: none"> • IPv4 address of AGPS server [optional]
<i>pServerPort</i>	<ul style="list-style-type: none"> • Port number of AGPS server [optional - should be present when pServerAddress is present]
<i>pServerURL</i>	<ul style="list-style-type: none"> • URL of the AGPS server [optional]
<i>pServerURL- Length</i>	<ul style="list-style-type: none"> • URL length of AGPS server [optional - should be present when pServerURL is present]
<i>pNetworkMode</i>	<ul style="list-style-type: none"> • Network Mode of AGPS Server [optional - should be present in Multimode Systems] <ul style="list-style-type: none"> – 0x00 - UMTS – 0x01 - CDMA

8.433.2 Field Documentation

8.433.2.1 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pNetworkMode

8.433.2.2 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerAddress

8.433.2.3 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerPort

8.433.2.4 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURL

8.433.2.5 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURLLength

8.434 pack_pds_SLQSSetPositionMethodState_t Struct Reference

Data Fields

- uint8_t * [pXtraTimeState](#)

- uint8_t * [pXtraDataState](#)
- uint8_t * [pWifiState](#)

8.434.1 Detailed Description

Parameters to Set state of positioning method for a device.

Parameters

<i>pXtraTimeState</i>	<ul style="list-style-type: none"> • XTRA Time Position Method State. • Values: <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pXtraDataState</i>	<ul style="list-style-type: none"> • XTRA Data Position Method State. • Values: <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>Latitude</i>	<ul style="list-style-type: none"> • WiFi Position Method State • Values: <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable

8.434.2 Field Documentation

8.434.2.1 uint8_t* [pack_pds_SLQSSetPositionMethodState_t::pWifiState](#)

8.434.2.2 uint8_t* [pack_pds_SLQSSetPositionMethodState_t::pXtraDataState](#)

8.434.2.3 uint8_t* [pack_pds_SLQSSetPositionMethodState_t::pXtraTimeState](#)

8.435 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](#)

8.435.1 Detailed Description

Structure contain the parameter for PDS tracking session.

Parameters

<i>sessionControl</i>	<ul style="list-style-type: none"> Control method: <ul style="list-style-type: none"> 0x0 - Manual
<i>sessionType</i>	<ul style="list-style-type: none"> Type: <ul style="list-style-type: none"> 0x0 - New
<i>session-Operation</i>	<ul style="list-style-type: none"> Operating mode: <ul style="list-style-type: none"> 0x00 - Standalone 0x01 - MS-based
<i>sessionServer-Option</i>	<ul style="list-style-type: none"> Location server option: <ul style="list-style-type: none"> 0x0 - Default
<i>fixTimeout</i>	<ul style="list-style-type: none"> Maximum time to work on each fix (in seconds, max 255)
<i>fixCount</i>	<ul style="list-style-type: none"> Count of position fix requests for this session (must be at least 1)
<i>fixInterval</i>	<ul style="list-style-type: none"> interval between position fix requests (in seconds)
<i>fixAccuracy</i>	<ul style="list-style-type: none"> Preferred accuracy threshold(in meters)

8.435.2 Field Documentation

8.435.2.1 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixAccuracy

8.435.2.2 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixCount

8.435.2.3 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixInterval

8.435.2.4 uint8_t pack_pds_StartPDSTrackingSessionExt_t::fixTimeout

8.435.2.5 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionControl

8.435.2.6 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionOperation

8.435.2.7 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionServerOption

8.435.2.8 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionType

8.436 pack_qmi_t Struct Reference

Data Fields

- uint16_t [xid](#)
- int [timeout](#)
- uint16_t [msgid](#)
- uint8_t [svc](#)

8.436.1 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

8.436.2 Field Documentation

8.436.2.1 uint16_t pack_qmi_t::msgid

8.436.2.2 uint8_t pack_qmi_t::svc

8.436.2.3 int pack_qmi_t::timeout

8.436.2.4 uint16_t pack_qmi_t::xid

8.437 pack_qos_BindDataPort_t Struct Reference

Data Fields

- qos_BindDataPortPeripheralEndPointID_t * pPeripheralEndPointID
- qos_BindDataPortMuxID_t * pMuxID
- qos_BindDataPortSIODDataPort_t * pSIODDataPort

8.437.1 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"> • See qos_BindDataPortPeripheralEndPointID_t for more information.
<i>pMuxID</i>	<ul style="list-style-type: none"> • See qos_BindDataPortMuxID_t for more information.
<i>pSIODDataPort</i>	<ul style="list-style-type: none"> • See qos_BindDataPortSIODDataPort_t for more information.

8.437.2 Field Documentation

8.437.2.1 qos_BindDataPortMuxID_t* pack_qos_BindDataPort_t::pMuxID

8.437.2.2 qos_BindDataPortPeripheralEndPointID_t* pack_qos_BindDataPort_t::pPeripheralEndPointID

8.437.2.3 qos_BindDataPortSIODataPort_t* pack_qos_BindDataPort_t::pSIODataPort

8.438 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.438.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

Parameters

<i>apnId</i> [IN]	<ul style="list-style-type: none">• APN id
-------------------	--

8.438.2 Field Documentation

8.438.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.439 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.439.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

<i>apnId</i> [IN]	<ul style="list-style-type: none">• APN id
-------------------	--

8.439.2 Field Documentation

8.439.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.440 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t [enable](#)

8.440.1 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"> • 1 - Enable QoS event reporting • 0 - Disable QoS event reporting
-----------	---------------	---

8.440.2 Field Documentation

8.440.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.441 pack_rms_SetSMSWake_t Struct Reference

Data Fields

- uint32_t [enabled](#)
- uint32_t [wake_mask](#)

8.441.1 Detailed Description

Pack set SMS wake parameters

Parameters

<i>enabled</i>	<ul style="list-style-type: none"> • SMS wake functionality enabled <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled
<i>wake_mask</i>	<ul style="list-style-type: none"> • SMS wake mask to search for incoming messages (only relevant when enabled)

8.441.2 Field Documentation

8.441.2.1 uint32_t pack_rms_SetSMSWake_t::enabled

8.441.2.2 uint32_t pack_rms_SetSMSWake_t::wake_mask

8.442 pack_sar_SLQSSetRfSarState_t Struct Reference

Data Fields

- uint32_t [RfSarState](#)

8.442.1 Detailed Description

This sturcture contain sets the specified RF SAR state parameter.

Parameters

<i>RfSarState</i>	<ul style="list-style-type: none"> SAR RF State <ul style="list-style-type: none"> QMI_SAR_RF_STATE_DEFAULT QMI_SAR_RF_STATE_1 QMI_SAR_RF_STATE_2 QMI_SAR_RF_STATE_3 QMI_SAR_RF_STATE_4 QMI_SAR_RF_STATE_5 QMI_SAR_RF_STATE_6 QMI_SAR_RF_STATE_7 QMI_SAR_RF_STATE_8
-------------------	--

8.442.2 Field Documentation

8.442.2.1 uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState

8.443 pack_sms_SaveSMS_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t [messageFormat](#)
- uint32_t [messageSize](#)
- uint8_t * [pMessage](#)

8.443.1 Detailed Description

Sutstructure containing parameters to Save an SMS message to device memory

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> SMS message storage type <ul style="list-style-type: none"> 0 - UIM - Invalid in case of CDMA device that does not require SIM 1 - NV
<i>messageFormat</i>	<ul style="list-style-type: none"> Message format <ul style="list-style-type: none"> 0 - CDMA (IS-637B) 1 - 5 (Reserved) 6 - GSM/WCDMA PP
<i>messageSize</i>	<ul style="list-style-type: none"> The length of the message contents in bytes

<i>pMessage</i>	<ul style="list-style-type: none"> • The message contents
-----------------	--

8.443.2 Field Documentation

8.443.2.1 uint32_t pack_sms_SaveSMS_t::messageFormat

8.443.2.2 uint32_t pack_sms_SaveSMS_t::messageSize

8.443.2.3 uint8_t* pack_sms_SaveSMS_t::pMessage

8.443.2.4 uint32_t pack_sms_SaveSMS_t::storageType

8.444 pack_sms_SendSMS_t Struct Reference

Data Fields

- uint32_t [messageFormat](#)
- uint32_t [messageSize](#)
- uint8_t * [pMessage](#)
- uint8_t * [pLinktimer](#)

8.444.1 Detailed Description

This structure contains pack send SMS parameters.

Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> • Message format <ul style="list-style-type: none"> – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 6 - GSM/WCDMA PP
<i>messageSize</i>	<ul style="list-style-type: none"> • The length of the message contents in bytes
<i>pLinktimer</i>	<ul style="list-style-type: none"> • GW SMS link open for the specified number of second
<i>pMessage</i>	<ul style="list-style-type: none"> • The message contents in PDU format contains SMS header and payload message

8.444.2 Field Documentation

8.444.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.444.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.444.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.444.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.445 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum [eqmiCbkJSetStatus status](#)

8.445.1 Detailed Description

This strucure contains pack set new SMS callback parameters.

Parameters

<i>status</i>	callback parameter
---------------	--------------------

8.445.2 Field Documentation

8.445.2.1 enum [eqmiCbkJSetStatus](#) pack_sms_SetNewSMSCallback_t::status

8.446 pack_sms_SetSMSCAddress_t Struct Reference

Data Fields

- uint8_t * [pSMSCAddress](#)
- uint8_t * [pSMSCType](#)

8.446.1 Detailed Description

Structure to Set the SMS center address.

Parameters

<i>SMSCAddress</i>	<ul style="list-style-type: none"> The SMS center address represented as a NULL terminated string
<i>SMSCType</i>	<ul style="list-style-type: none"> The SMS center address type represented as a NULL terminated string (optional).

8.446.2 Field Documentation

8.446.2.1 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCAddress

8.446.2.2 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCType

8.447 pack_sms_SLQSDDeleteSMS_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t * [pMessageIndex](#)
- uint32_t * [pMessageTag](#)
- uint8_t * [pMessageMode](#)

8.447.1 Detailed Description

This structure contains pack delete SMS parameters.

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> • SMS message storage type <ul style="list-style-type: none"> – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
<i>pMessageIndex</i>	<ul style="list-style-type: none"> • (Optional) message index
<i>pMessageTag</i>	<ul style="list-style-type: none"> • (Optional) message tag <ul style="list-style-type: none"> – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
<i>pMessageMode</i>	<ul style="list-style-type: none"> • (Optional) message mode • this must be included if the device is capable of supporting more than one protocol • e.g. CDMA and GW <ul style="list-style-type: none"> – 0x00 - CDMA, LTE (if network type is CDMA) – 0x01 - GW, LTE (if network type is UMTS)

8.447.2 Field Documentation

8.447.2.1 uint32_t* pack_sms_SLQSDelSMS_t::pMessageIndex

8.447.2.2 uint8_t* pack_sms_SLQSDelSMS_t::pMessageMode

8.447.2.3 uint32_t* pack_sms_SLQSDelSMS_t::pMessageTag

8.447.2.4 uint32_t pack_sms_SLQSDelSMS_t::storageType

8.448 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t [messageIndex](#)
- uint8_t * [pMessageMode](#)

8.448.1 Detailed Description

This sturcture contains pack get SMS parameters.

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> SMS message storage type <ul style="list-style-type: none"> 0 - UIM - Invalid in case of CDMA device that does not require SIM 1 - NV
<i>messageIndex</i>	<ul style="list-style-type: none"> Message index
<i>pMessageMode</i>	<ul style="list-style-type: none"> 0x00 - CDMA, LTE (if network type is CDMA) 0x01 - GW, LTE (if network type is UMTS)

8.448.2 Field Documentation

8.448.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.448.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

8.448.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

8.449 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

Data Fields

- uint8_t [mode](#)

8.449.1 Detailed Description

This structure contains get SMS boardcast configure parameter.

Parameters

<i>mode</i>	<ul style="list-style-type: none"> Mode <ul style="list-style-type: none"> 0x00 - CDMA, LTE (if network type is CDMA) 0x01 - GW, LTE (if network type is UMTS)
-------------	--

8.449.2 Field Documentation

8.449.2.1 uint8_t pack_sms_SLQSGetSmsBroadcastConfig_t::mode

8.450 pack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t * [pRequestedTag](#)
- uint8_t * [pMessageMode](#)

8.450.1 Detailed Description

This structure contains pack get SMS list parameters.

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> • SMS message storage type <ul style="list-style-type: none"> – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
<i>requestedTag</i>	<ul style="list-style-type: none"> • (Optional) Message tag <ul style="list-style-type: none"> – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
<i>messageMode</i>	<ul style="list-style-type: none"> • 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

8.450.2 Field Documentation

8.450.2.1 uint8_t* [pack_sms_SLQSGetSMSList_t::pMessageMode](#)

8.450.2.2 uint32_t* [pack_sms_SLQSGetSMSList_t::pRequestedTag](#)

8.450.2.3 uint32_t [pack_sms_SLQSGetSMSList_t::storageType](#)

8.451 [pack_sms_SLQSMModifySMSStatus_t](#) Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t [messageIndex](#)
- uint32_t [messageTag](#)
- uint8_t * [pMessageMode](#)

8.451.1 Detailed Description

This structure contains pack modify SMS status parameters.

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> SMS message storage type <ul style="list-style-type: none"> 0 - UIM - Invalid in case of CDMA device that does not require SIM 1 - NV
<i>messageIndex</i>	<ul style="list-style-type: none"> Message index
<i>messageTag</i>	<ul style="list-style-type: none"> Message tag <ul style="list-style-type: none"> 0 - Read 1 - Not read
<i>pMessageMode</i>	<ul style="list-style-type: none"> 0x00 - CDMA, LTE (if network type is CDMA) 0x01 - GW, LTE (if network type is UMTS)

8.451.2 Field Documentation

8.451.2.1 uint32_t pack_sms_SLQSMModifySMSStatus_t::messageIndex

8.451.2.2 uint32_t pack_sms_SLQSMModifySMSStatus_t::messageTag

8.451.2.3 uint8_t* pack_sms_SLQSMModifySMSStatus_t::pMessageMode

8.451.2.4 uint32_t pack_sms_SLQSMModifySMSStatus_t::storageType

8.452 pack_sms_SLQSSendAsyncSMS_t Struct Reference

Data Fields

- [sms_sendAsynsmsParams](#) * [pSendSmsParams](#)

8.452.1 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"> structure containing the SMS parameters. Refer sms_sendasynsmsparams
------------------------	--

8.452.2 Field Documentation

8.452.2.1 [sms_sendAsynsmsParams](#)* pack_sms_SLQSSendAsyncSMS_t::pSendSmsParams

8.453 pack_sms_SLQSSetIndicationRegister_t Struct Reference

Data Fields

- [sms_setIndicationReg](#) * [pSetIndicationRegReq](#)

8.453.1 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"> • Pointer to structure of indicationRegReqParams <ul style="list-style-type: none"> – See sms_setIndicationReg for more information
--------------------------------	---

8.453.2 Field Documentation

8.453.2.1 [sms_setIndicationReg](#)* [pack_sms_SLQSSetIndicationRegister_t::pSetIndicationRegReq](#)

8.454 [pack_sms_SLQSSetSmsBroadcastActivation_t](#) Struct Reference

Data Fields

- [uint8_t](#) [mode](#)
- [uint8_t](#) [broadcastActivate](#)

8.454.1 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"> • Mode • 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)
<i>broadcast- Activate</i>	<ul style="list-style-type: none"> • 0x00 - Disable broadcast • 0x01 - Activate broadcast

8.454.2 Field Documentation

8.454.2.1 [uint8_t](#) [pack_sms_SLQSSetSmsBroadcastActivation_t::broadcastActivate](#)

8.454.2.2 [uint8_t](#) [pack_sms_SLQSSetSmsBroadcastActivation_t::mode](#)

8.455 [pack_sms_SLQSSetSmsBroadcastConfig_t](#) Struct Reference

Data Fields

- [uint8_t mode](#)
- [sms_qaQmi3GPPBroadcastCfgInfo](#) * [pBroadcastConfig](#)
- [sms_qaQmi3GPP2BroadcastCfgInfo](#) * [pCDMABroadcastConfig](#)

8.455.1 Detailed Description

Structure containing parameters that provides the information about the SMS BroadcastConfiguration

Parameters

<i>mode</i>	<ul style="list-style-type: none"> • Mode <ul style="list-style-type: none"> – 0x00 - CDMA, LTE (if network type is CDMA) – 0x01 - GW, LTE (if network type is UMTS)
<i>pBroadcast-Config</i>	<ul style="list-style-type: none"> • The data for 3GPP Broadcast Information(Optional).
<i>pCDMA-BroadcastConfig</i>	<ul style="list-style-type: none"> • The data for 3GPP2 Broadcast Information(Optional).

8.455.2 Field Documentation

8.455.2.1 [uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode](#)

8.455.2.2 [sms_qaQmi3GPPBroadcastCfgInfo](#)* [pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig](#)

8.455.2.3 [sms_qaQmi3GPP2BroadcastCfgInfo](#)* [pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig](#)

8.456 pack_sms_SLQSSetSmsStorage_t Struct Reference

Data Fields

- [uint8_t smsStorage](#)

8.456.1 Detailed Description

Structure contain Parameter to set the SMS Storage on the device

Parameters

<i>smsStorage</i>	<ul style="list-style-type: none"> • SMS Storage <ul style="list-style-type: none"> – 0x01 - device's permanent memory – 0x02 - UICC
-------------------	--

8.456.2 Field Documentation

8.456.2.1 `uint8_t pack_sms_SLQSSetSmsStorage_t::smsStorage`

8.457 `pack_sms_SLQSSmsGetMaxStorageSize_t` Struct Reference

Data Fields

- [sms_maxStorageSizeReq](#) * [pMaxStorageSizeReq](#)

8.457.1 Detailed Description

This structure contains get maximum storage size.

Parameters

<i>pMaxStorage-SizeReq</i>	<ul style="list-style-type: none"> • Request parameters for <code>SmsSLQSGetMaxStorageSize</code> <ul style="list-style-type: none"> – See sms_maxStorageSizeReq for more information
----------------------------	--

8.457.2 Field Documentation

8.457.2.1 `sms_maxStorageSizeReq* pack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeReq`

8.458 `pack_sms_SLQSSmsSetRoutes_t` Struct Reference

Data Fields

- [sms_setRoutesReq](#) * [pSetRoutesReq](#)

8.458.1 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"> • Pointer to structure of <code>smsSetRoutesReq</code> <ul style="list-style-type: none"> – See sms_setRoutesReq for more information
----------------------	---

8.458.2 Field Documentation

8.458.2.1 `sms_setRoutesReq* pack_sms_SLQSSmsSetRoutes_t::pSetRoutesReq`

8.459 `pack_swiaudio_SLQSGetM2MAudioProfile_t` Struct Reference

Data Fields

- `uint8_t` * [pGenerator](#)

8.459.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile pack parameters.

Parameters

<i>p-Generator[optional]</i>	<ul style="list-style-type: none">• Generator<ul style="list-style-type: none">– 0 - Voice
------------------------------	--

8.459.2 Field Documentation

8.459.2.1 uint8_t* pack_swiaudio_SLQSGetM2MAudioProfile_t::pGenerator

8.460 pack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [Generator](#)

8.460.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none">• Audio Profile<ul style="list-style-type: none">– 0-5
<i>Generator</i>	<ul style="list-style-type: none">• Generator<ul style="list-style-type: none">– 0 - Voice

8.460.2 Field Documentation

8.460.2.1 uint8_t pack_swiaudio_SLQSGetM2MAudioVolume_t::Generator

8.460.2.2 uint8_t pack_swiaudio_SLQSGetM2MAudioVolume_t::Profile

8.461 pack_swiaudio_SLQSGetM2MAVMute_t Struct Reference

Data Fields

- uint8_t [Profile](#)

8.461.1 Detailed Description

This structure contains the SLQSGetM2MAVMute pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile Number <ul style="list-style-type: none"> – 0-5
----------------	---

8.461.2 Field Documentation

8.461.2.1 uint8_t pack_swiaudio_SLQSGetM2MAVMute_t::Profile

8.462 pack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference

Data Fields

- uint8_t [Profile](#)

8.462.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile Number <ul style="list-style-type: none"> – 0-5
----------------	---

8.462.2 Field Documentation

8.462.2.1 uint8_t pack_swiaudio_SLQSGetM2MSpkrGain_t::Profile

8.463 pack_swiaudio_SLQSSetM2MAudioAVCFG_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [Device](#)
- uint8_t [PIFACEId](#)
- [swiaudio_PCMparams](#) * [pPCMPParams](#)

8.463.1 Detailed Description

This structure contains the SLQSSetM2MAudioAVCFG pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile <ul style="list-style-type: none"> – 0-5
----------------	--

<i>Device</i>	<ul style="list-style-type: none"> • ACDB Device • See qaGobiApiTableSwiAudio.h for more information on ACDB Device
<i>PIFACEId</i>	<ul style="list-style-type: none"> • Physical Interface • See qaGobiApiTableSwiAudio.h for more information on physical interface
<i>pPCMPParams</i>	<ul style="list-style-type: none"> • PCM parameters • See swiaudio_PCMparams for more information

8.463.2 Field Documentation

8.463.2.1 uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::Device

8.463.2.2 uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::PIFACEId

8.463.2.3 swiaudio_PCMparams* pack_swiaudio_SLQSSetM2MAudioAVCFG_t::pPCMPParams

8.463.2.4 uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::Profile

8.464 pack_swiaudio_SLQSSetM2MAudioLPBK_t Struct Reference

Data Fields

- uint8_t [Enable](#)

8.464.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK pack parameters.

Parameters

<i>Enable</i>	<ul style="list-style-type: none"> • Operation to be performed <ul style="list-style-type: none"> – 0 - stop – 1 - VOCODER loop – 2 - internal codec loop
---------------	--

8.464.2 Field Documentation

8.464.2.1 uint8_t pack_swiaudio_SLQSSetM2MAudioLPBK_t::Enable

8.465 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](#)

8.465.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile Number <ul style="list-style-type: none"> – 0-5
<i>pEarMute</i>	<ul style="list-style-type: none"> • Ear Mute <ul style="list-style-type: none"> – 0 - mute – 1 - unmute
<i>pMicMute</i>	<ul style="list-style-type: none"> • Mic Mute <ul style="list-style-type: none"> – 0 - mute – 1 - unmute
<i>pGenerator</i>	<ul style="list-style-type: none"> • Generator <ul style="list-style-type: none"> – 0 - voice
<i>pVolume</i>	<ul style="list-style-type: none"> • Set RX Volume level <ul style="list-style-type: none"> – 0-5
<i>pCwtMute</i>	<ul style="list-style-type: none"> • Call Waiting Tone Mute <ul style="list-style-type: none"> – 0 - Mute – 1 - UnMute

8.465.2 Field Documentation

- 8.465.2.1 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pCwtMute
- 8.465.2.2 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pEarMute
- 8.465.2.3 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pGenerator
- 8.465.2.4 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pMicMute
- 8.465.2.5 uint8_t pack_swiaudio_SLQSSetM2MAudioProfile_t::Profile
- 8.465.2.6 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pVolume

8.466 pack_swiaudio_SLQSSetM2MAudioVolume_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [Generator](#)
- uint8_t [Level](#)

8.466.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none">• Audio Profile Number<ul style="list-style-type: none">– 0-5
<i>Generator</i>	<ul style="list-style-type: none">• Generator<ul style="list-style-type: none">– 0 - voice
<i>Level</i>	<ul style="list-style-type: none">• Audio volume level<ul style="list-style-type: none">– 0-5

8.466.2 Field Documentation

8.466.2.1 uint8_t pack_swiaudio_SLQSSetM2MAudioVolume_t::Generator

8.466.2.2 uint8_t pack_swiaudio_SLQSSetM2MAudioVolume_t::Level

8.466.2.3 uint8_t pack_swiaudio_SLQSSetM2MAudioVolume_t::Profile

8.467 pack_swiaudio_SLQSSetM2MAVMute_t Struct Reference

Data Fields

- uint8_t [Profile](#)
- uint8_t [EarMute](#)
- uint8_t [MicMute](#)
- uint8_t * [pCwtMute](#)

8.467.1 Detailed Description

This structure contains the SLQSSetM2MAVMute pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none">• Audio Profile Number<ul style="list-style-type: none">– 0-5
----------------	---

<i>EarMute</i>	<ul style="list-style-type: none"> • Ear Mute <ul style="list-style-type: none"> – 0-1
<i>MicMute</i>	<ul style="list-style-type: none"> • Mic Mute <ul style="list-style-type: none"> – 0-1
<i>pCwtMute</i>	[Optional] <ul style="list-style-type: none"> • Call Waiting Tone Mute <ul style="list-style-type: none"> – 0-1

8.467.2 Field Documentation

8.467.2.1 `uint8_t pack_swiaudio_SLQSSetM2MAVMute_t::EarMute`

8.467.2.2 `uint8_t pack_swiaudio_SLQSSetM2MAVMute_t::MicMute`

8.467.2.3 `uint8_t* pack_swiaudio_SLQSSetM2MAVMute_t::pCwtMute`

8.467.2.4 `uint8_t pack_swiaudio_SLQSSetM2MAVMute_t::Profile`

8.468 `pack_swiaudio_SLQSSetM2MSpkrGain_t` Struct Reference

Data Fields

- `uint8_t` [Profile](#)
- `uint16_t` [Value](#)

8.468.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain pack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile Number <ul style="list-style-type: none"> – 0-5
<i>Value</i>	<ul style="list-style-type: none"> • RX speakerphone gain <ul style="list-style-type: none"> – 0x0 - 0x7fff

8.468.2 Field Documentation

8.468.2.1 `uint8_t pack_swiaudio_SLQSSetM2MSpkrGain_t::Profile`

8.468.2.2 `uint16_t pack_swiaudio_SLQSSetM2MSpkrGain_t::Value`

8.469 pack_swivms_SLQSAVMSSendSelection_t Struct Reference

Data Fields

- uint8_t [selection](#)
- uint32_t * [pDeferTime](#)
- uint8_t * [pClientPerformOperationFlag](#)
- uint8_t * [pPackageID](#)
- uint16_t * [pRejectReason](#)

8.469.1 Detailed Description

Structure containing the AVMS selection

Parameters

<i>selection</i> [IN]	<ul style="list-style-type: none"> • User Selection <ul style="list-style-type: none"> – 0x01 - Accept – 0x02 - Reject – 0x03 - Defer
<i>pDeferTime</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> • Defer time in minutes. A value of 0 will cause the prompt to be resent immediately.
<i>pClientPerformOperationFlag</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> • Client operation flag after accept. <ul style="list-style-type: none"> – 0: if modem performs the operation (download or update) – 1: if client performs the operation (download or update)
<i>pPackageID</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> • Package ID.
<i>pRejectReason</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> • Reject Reason.

8.469.2 Field Documentation

8.469.2.1 uint8_t* pack_swivms_SLQSAVMSSendSelection_t::pClientPerformOperationFlag

8.469.2.2 uint32_t* pack_swivms_SLQSAVMSSendSelection_t::pDeferTime

8.469.2.3 uint8_t* pack_swivms_SLQSAVMSSendSelection_t::pPackageID

8.469.2.4 uint16_t* pack_swivms_SLQSAVMSSendSelection_t::pRejectReason

8.469.2.5 uint8_t pack_swivms_SLQSAVMSSendSelection_t::selection

8.470 pack_swivms_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](#)

8.470.1 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"> • Auto Connect to AirVantage server <ul style="list-style-type: none"> – 0x00 - FALSE – 0x01 - TRUE
<i>AutoReboot</i> [IN]	<ul style="list-style-type: none"> • Automatic device reboot when the request is received from AirVantage server <ul style="list-style-type: none"> – 0x00 - FALSE – 0x01 - TRUE
<i>PromptFw-Download</i> [IN]	<ul style="list-style-type: none"> • Firmware Auto Download <ul style="list-style-type: none"> – 0x00 - FALSE – 0x01 - TRUE
<i>PromptFw-Update</i> [IN]	<ul style="list-style-type: none"> • 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> – 0x00 - Firmware autoupdate FALSE – 0x01 - Firmware autoupdate TRUE
<i>pFwAutoSDM</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • OMA Automatic UI Alert Response <ul style="list-style-type: none"> – 0x00 - DISABLED – 0x01 - ENABLED ACCEPT – 0x02 - ENABLED REJECT
<i>pPollingTimer</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • Polling timer to connect to AVMS server <ul style="list-style-type: none"> – 0-525600 (min) – 0:disabled
<i>pConnection-RetryTimers</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsConnectionRetryTimers for more information

<i>pAPNInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsAPNInfo for more information
<i>pNotifStore</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsAPNInfo for more information
<i>pPeriodInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsAPNInfo for more information

8.470.2 Field Documentation

8.470.2.1 `uint8_t pack_swiaavms_SLQSAVMSSetSettings_t::AutoConnect`

8.470.2.2 `uint8_t pack_swiaavms_SLQSAVMSSetSettings_t::AutoReboot`

8.470.2.3 `PackSwiAvmsSetSettingsAPNInfo* pack_swiaavms_SLQSAVMSSetSettings_t::pAPNInfo`

8.470.2.4 `PackSwiAvmsSetSettingsConnectionRetryTimers* pack_swiaavms_SLQSAVMSSetSettings_t::pConnectionRetryTimers`

8.470.2.5 `uint8_t* pack_swiaavms_SLQSAVMSSetSettings_t::pFwAutoSDM`

8.470.2.6 `uint8_t* pack_swiaavms_SLQSAVMSSetSettings_t::pNotifStore`

8.470.2.7 `PackSwiAvmsSetSettingsPeriodInfo* pack_swiaavms_SLQSAVMSSetSettings_t::pPeriodInfo`

8.470.2.8 `uint32_t* pack_swiaavms_SLQSAVMSSetSettings_t::pPollingTimer`

8.470.2.9 `uint8_t pack_swiaavms_SLQSAVMSSetSettings_t::PromptFwDownload`

8.470.2.10 `uint8_t pack_swiaavms_SLQSAVMSSetSettings_t::PromptFwUpdate`

8.471 pack_swiaavms_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`

8.471.1 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"> • Auto Connect to AirVantage server <ul style="list-style-type: none"> – 0x00 - FALSE – 0x01 - TRUE
<i>PromptFw-Download</i> [IN]	<ul style="list-style-type: none"> • Firmware Auto Download <ul style="list-style-type: none"> – 0x00 - FALSE – 0x01 - TRUE
<i>PromptFw-Update</i> [IN]	<ul style="list-style-type: none"> • 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> – 0x00 - Firmware autoupdate FALSE – 0x01 - Firmware autoupdate TRUE
<i>pFwAutoSDM</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • OMA Automatic UI Alert Response <ul style="list-style-type: none"> – 0x00 - DISABLED – 0x01 - ENABLED ACCEPT – 0x02 - ENABLED REJECT
<i>pPollingTimer</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • Polling timer to connect to AVMS server <ul style="list-style-type: none"> – 0-525600 (min) – 0:disabled
<i>pConnection-RetryTimers</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsConnectionRetryTimers for more information
<i>pAPNInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsAPNInfo for more information
<i>pNotifStore</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsAPNInfo for more information
<i>pPeriodInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • See PackSwiAvmsSetSettingsAPNInfo for more information
<i>pAutoReboot</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> • Automatic device reboot when the request is received from AirVantage server <ul style="list-style-type: none"> – 0x00 - FALSE – 0x01 - TRUE

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.

8.471.2 Field Documentation

- 8.471.2.1 uint8_t pack_swiaavms_SLQSAVMSSetSettings_v2_t::AutoConnect
- 8.471.2.2 PackSwiAvmsSetSettingsAPNInfo* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pAPNInfo
- 8.471.2.3 uint8_t* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pAutoReboot
- 8.471.2.4 PackSwiAvmsSetSettingsConnectionRetryTimers* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pConnectionRetryTimers
- 8.471.2.5 uint8_t* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pFwAutoSDM
- 8.471.2.6 uint8_t* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pNotifStore
- 8.471.2.7 PackSwiAvmsSetSettingsPeriodInfo* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pPeriodInfo
- 8.471.2.8 uint32_t* pack_swiaavms_SLQSAVMSSetSettings_v2_t::pPollingTimer
- 8.471.2.9 uint8_t pack_swiaavms_SLQSAVMSSetSettings_v2_t::PromptFwDownload
- 8.471.2.10 uint8_t pack_swiaavms_SLQSAVMSSetSettings_v2_t::PromptFwUpdate

8.472 pack_swiaavms_SLQSAVMSStartSession_t Struct Reference

Data Fields

- uint8_t [sessionType](#)

8.472.1 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">• Session type<ul style="list-style-type: none">– 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.
------------------------	--

8.472.2 Field Documentation

- 8.472.2.1 uint8_t pack_swiaavms_SLQSAVMSStartSession_t::sessionType

8.473 pack_swiaavms_SLQSAVMSStopSession_t Struct Reference

Data Fields

- uint8_t [sessionType](#)

8.473.1 Detailed Description

Structure that contains the session type for AVMS Stop session command

Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> Session type <ul style="list-style-type: none"> 0x01 - FOTA, to suspend FOTA session. 0xFF - Suspend ongoing FOTA session or stop any other active AVMS session.
<i>resultCode</i>	<ul style="list-style-type: none"> Tlv Result Code

8.473.2 Field Documentation

8.473.2.1 `uint8_t pack_swiaavms_SLQSAVMSStopSession_t::sessionType`8.474 `pack_swidms_SLQSSwiDmsSetHWWatchdog_t` Struct Reference

Data Fields

- `uint32_t` [timeout](#)
- `uint32_t` [resetDelay](#)
- `uint8_t` [enable](#)

8.474.1 Detailed Description

This structure contains the set hardware watchdog settings request parameter.

Parameters

<i>timeout</i>	<ul style="list-style-type: none"> timeout value for HW watchdog (unit in second)
<i>resetDelay</i>	<ul style="list-style-type: none"> delay before reset after watchdog timeout (unit in second)
<i>enable</i>	<ul style="list-style-type: none"> 0 to disable watchdog; 1 to enable watchdog

8.474.2 Field Documentation

8.474.2.1 `uint8_t pack_swidms_SLQSSwiDmsSetHWWatchdog_t::enable`8.474.2.2 `uint32_t pack_swidms_SLQSSwiDmsSetHWWatchdog_t::resetDelay`8.474.2.3 `uint32_t pack_swidms_SLQSSwiDmsSetHWWatchdog_t::timeout`8.475 `pack_swidms_SLQSSwiDmsSetMTU_t` Struct Reference

Data Fields

- `uint16_t` [MTUSize](#)

8.475.1 Detailed Description

This structure contains the Set MTU request parameter.

Parameters

<i>MTUSize</i>	<ul style="list-style-type: none">• Set MTU size• MTU Values<ul style="list-style-type: none">– 0 - use default value– 576 to 2000 - other values required by carrier.– Set the same MTU for all RAT/interfaces.– New MTU size effective after modem reboot.
----------------	--

8.475.2 Field Documentation

8.475.2.1 `uint16_t pack_swidms_SLQSSwiDmsSetMTU_t::MTUSize`

8.476 pack_swidms_SLQSSwiDmsSetUsbComp_t Struct Reference

Data Fields

- `uint32_t CfgValue`

8.476.1 Detailed Description

This structure contains the Set Usb Interface Config request parameter.

Parameters

<i>CfgValue</i>	<ul style="list-style-type: none"> • Host composition bit mask, see supported USB interface bitmasks, not supported by QMI object setting for 9x50 modules <ul style="list-style-type: none"> – 0x00000001 - DIAG interface – 0x00000002 - ADB interface – 0x00000004 - NMEA interface – 0x00000008 - MODEM interface – 0x00000010 - RESERVED5 – 0x00000020 - RESERVED6 – 0x00000040 - RESERVED7 – 0x00000080 - RESERVED8 – 0x00000100 - RMENT0 interface – 0x00000200 - RESERVED10 – 0x00000400 - RMENT1 interface – 0x00000800 - RESERVED12 – 0x00001000 - MBIM interface – 0x00002000 - RESERVED14 – 0x00004000 - RNDIS interface – 0x00008000 - RESERVED16 – 0x00010000 - AUDIO interface – 0x00020000 - RESERVED18 – 0x00080000 - ECM interface – 0x00100000 - RESERVED21 – 0x00200000 - RESERVED22 – 0xFFC00000 - RESERVED
-----------------	--

8.476.2 Field Documentation

8.476.2.1 `uint32_t pack_swidms_SLQSSwiDmsSetUsbComp_t::CfgValue`

8.477 `pack_swidms_SLQSSwiDmsSetUsbNetNum_t` Struct Reference

Data Fields

- `uint8_t` [nUsbNetNum](#)

8.477.1 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"> • value of usb net numbers to set for the device • range is 0-8
-------------------	--

8.477.2 Field Documentation

8.477.2.1 uint8_t pack_swidms_SLQSSwiDmsSetUsbNetNum_t::nUsbNetNum

8.478 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](#)

8.478.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

Parameters

<i>function</i>	<ul style="list-style-type: none"> • Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> – 0 - disabled – 1 - At bootup – 2 - When NMEA port is opened
<i>set_function</i>	<ul style="list-style-type: none"> • 0 - do not set to modem • 1 - set to modem
<i>fix_type</i>	<ul style="list-style-type: none"> • Type of GNSS fix: <ul style="list-style-type: none"> – 1 - Default Engine mode – 2 - MS-Based – 3 - MS-Assisted – 4 - Standalone
<i>set_fix_type</i>	<ul style="list-style-type: none"> • 0 - do not set to modem • 1 - set to modem
<i>max_time</i>	<ul style="list-style-type: none"> • Maximum time allowed for the receiver to get a fix in seconds • Valid range: 1-255
<i>set_max_time</i>	<ul style="list-style-type: none"> • 0 - do not set to modem • 1 - set to modem

<i>max_dist</i>	<ul style="list-style-type: none"> • Maximum uncertainty of a fix measured by distance in meters • Valid range: 1 - 4294967280
<i>set_max_dist</i>	<ul style="list-style-type: none"> • 0 - do not set to modem • 1 - set to modem
<i>fix_rate</i>	<ul style="list-style-type: none"> • Time between fixes in seconds • Valid range: 1–65535
<i>set_fix_rate</i>	<ul style="list-style-type: none"> • 0 - do not set to modem • 1 - set to modem

8.478.2 Field Documentation

8.478.2.1 `uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate`

8.478.2.2 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type`

8.478.2.3 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::function`

8.478.2.4 `uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist`

8.478.2.5 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time`

8.478.2.6 `int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate`

8.478.2.7 `int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type`

8.478.2.8 `int pack_swiloc_SwiLocSetAutoStart_t::set_function`

8.478.2.9 `int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist`

8.478.2.10 `int pack_swiloc_SwiLocSetAutoStart_t::set_max_time`

8.479 `pack_swiooma_SLQSOMADMCcancelSession_t` Struct Reference

Data Fields

- `uint32_t` [sessionType](#)

8.479.1 Detailed Description

Structure that contains the session type for OMA cancel session command

Parameters

<i>sessionType</i>	<div><div>[in]</div><ul style="list-style-type: none">• Session type<ul style="list-style-type: none">– 0x01 - FOTA, to check availability of FW Update– 0xFF - Cancel any active OMADM session</div>
--------------------	--

8.479.2 Field Documentation

8.479.2.1 uint32_t pack_swima_SLQSOMADMCancelSession_t::sessionType

8.480 pack_swima_SLQSOMADMCancelSessionExt_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.480.1 Detailed Description

Structure that contains the session type for OMA cancel session command

Parameters

<i>sessionType</i>	<div><div>[IN]</div><ul style="list-style-type: none">• Session type<ul style="list-style-type: none">– 0x01 - FOTA, to check availability of FW Update– 0xFF - Cancel any active OMADM session</div>
--------------------	--

8.480.2 Field Documentation

8.480.2.1 uint32_t pack_swima_SLQSOMADMCancelSessionExt_t::sessionType

8.481 pack_swima_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t [SessionType](#)

8.481.1 Detailed Description

Structure that contains the session type for OMA get session info command

Parameters

<i>SessionType</i>	<div><div>[in]</div><ul style="list-style-type: none">• Session type<ul style="list-style-type: none">– 0x01 - FOTA– 0xFF - Any active OMADM session. If no active sessions are available, then previous OMADM session info is returned</div>
--------------------	--

8.481.2 Field Documentation

8.481.2.1 uint32_t pack_swima_SLQSOMADMGetSessionInfo_t::SessionType

8.482 pack_swima_SLQSOMADMSendSelection_t Struct Reference

Data Fields

- uint32_t [selection](#)
- uint32_t * [pDeferTime](#)
- uint32_t * [pRejectReason](#)

8.482.1 Detailed Description

Structure containing the OMA DM selection

Parameters

<i>selection</i>	[in] <ul style="list-style-type: none"> • OMA-DM NIA Selection <ul style="list-style-type: none"> – 0x01 - Accept – 0x02 - Reject – 0x03 - Defer
<i>pDeferTime</i>	[in] <ul style="list-style-type: none"> • Defer time in minutes. A value of 0 will cause the prompt to be resent immediately. • This TLV is mandatory if selection is set to 0x03.
<i>pRejectReason</i>	[in] <ul style="list-style-type: none"> • Reject Reason • This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default.

8.482.2 Field Documentation

8.482.2.1 uint32_t* pack_swima_SLQSOMADMSendSelection_t::pDeferTime

8.482.2.2 uint32_t* pack_swima_SLQSOMADMSendSelection_t::pRejectReason

8.482.2.3 uint32_t pack_swima_SLQSOMADMSendSelection_t::selection

8.483 pack_swima_SLQSOMADMSendSelectionExt_t Struct Reference

Data Fields

- uint32_t [selection](#)

8.483.1 Detailed Description

Structure containing the OMA DM selection

Parameters

<i>selection</i>	[IN] <ul style="list-style-type: none"> OMA-DM NIA Selection <ul style="list-style-type: none"> 0x01 - Accept 0x02 - Reject 0x03 - Defer
------------------	---

8.483.2 Field Documentation

8.483.2.1 uint32_t pack_swioama_SLQSOMADMSendSelectionExt_t::selection

8.484 pack_swioama_SLQSOMADMSetSettings_t Struct Reference

Data Fields

- uint8_t [FOTAdownload](#)
- uint8_t [FOTAUpdate](#)
- uint8_t * [pAutosdm](#)
- uint8_t * [pFwAutoCheck](#)

8.484.1 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"> 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> 0x00 - Firmware autodownload FALSE 0x01 - Firmware autodownload TRUE
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> 0x00 - Firmware autoupdate FALSE 0x01 - Firmware autoupdate TRUE
<i>pAutosdm</i>	<ul style="list-style-type: none"> Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> 0x00 - Disabled 0x01 - Enabled Accept 0x02 - Enabled Reject
<i>pFwAutoCheck</i>	<ul style="list-style-type: none"> Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> 0x00 - Disabled 0x01 - Enabled

8.484.2 Field Documentation

8.484.2.1 `uint8_t pack_swisma_SLQSOMADMSetSettings_t::FOTAdownload`

8.484.2.2 `uint8_t pack_swisma_SLQSOMADMSetSettings_t::FOTAUpdate`

8.484.2.3 `uint8_t* pack_swisma_SLQSOMADMSetSettings_t::pAutosdm`

8.484.2.4 `uint8_t* pack_swisma_SLQSOMADMSetSettings_t::pFwAutoCheck`

8.485 `pack_swisma_SLQSOMADMSetSettingsExt_t` Struct Reference

Data Fields

- `uint8_t FOTAdownload`
- `uint8_t FOTAUpdate`
- `uint8_t OMADMEEnable`
- `uint8_t OMADMLogEnable`
- `uint8_t FUMOEnable`
- `uint8_t PRLEnable`

8.485.1 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"> • 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> – 0x00 - Firmware autodownload FALSE – 0x01 - Firmware autodownload TRUE
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> • 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> – 0x00 - Firmware autoupdate FALSE – 0x01 - Firmware autoupdate TRUE
<i>OMADMEEnable</i>	[IN] <ul style="list-style-type: none"> • 1 byte parameter indicating OMA DM <ul style="list-style-type: none"> – 0x00 - OMA DM Disabled – 0x01 - OMA DM Enabled
<i>OMADMLog-Enable</i>	[IN] <ul style="list-style-type: none"> • 1 byte parameter indicating OMA DM Logs <ul style="list-style-type: none"> – 0x00 - OMA DM Logs Disabled – 0x01 - OMA DM Logs Enabled
<i>FUMOEnable</i>	[IN] <ul style="list-style-type: none"> • 1 byte parameter indicating FUMO enabled <ul style="list-style-type: none"> – 0x00 - FUMO Disabled – 0x01 - FUMO enabled

<i>PRLEnable</i>	<div><div>[IN]</div><ul style="list-style-type: none">• 1 byte parameter indicating PRL Enabled<ul style="list-style-type: none">– 0x00 - PRL Disabled– 0x01 - PRL Enabled– 0x02 - Launch a CI PRL– 0x03 - PRL Enabled and PRL automatic update every 45 days– 0x04 - PRL Enabled and PRL update ever 90 days</div>
------------------	---

8.485.2 Field Documentation

8.485.2.1 uint8_t pack_swisma_SLQSOMADMSetSettingsExt_t::FOTAdownload

8.485.2.2 uint8_t pack_swisma_SLQSOMADMSetSettingsExt_t::FOTAUpdate

8.485.2.3 uint8_t pack_swisma_SLQSOMADMSetSettingsExt_t::FUMOEnable

8.485.2.4 uint8_t pack_swisma_SLQSOMADMSetSettingsExt_t::OMADMEEnable

8.485.2.5 uint8_t pack_swisma_SLQSOMADMSetSettingsExt_t::OMADMLogEnable

8.485.2.6 uint8_t pack_swisma_SLQSOMADMSetSettingsExt_t::PRLEnable

8.486 pack_swisma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.486.1 Detailed Description

Structure that contains the session type for OMA start session command

Parameters

<i>sessionType</i> [IN]	<div><ul style="list-style-type: none">• Session type<ul style="list-style-type: none">– 0x01 - FOTA, to check availability of FW Update– 0x02 - DM, to check availability of DM Update– 0x03 - PRL, to check availability of PRL Update</div>
-------------------------	--

8.486.2 Field Documentation

8.486.2.1 uint32_t pack_swisma_SLQSOMADMStartSession_t::sessionType

8.487 pack_swisma_SLQSOMADMStartSessionExt_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.487.1 Detailed Description

Structure that contains the session type for OMA start session command

Parameters

<i>sessionType</i>	[IN] <ul style="list-style-type: none"> Session type <ul style="list-style-type: none"> 0x01 - FOTA, to check availability of FW Update 0x02 - DM, to check availability of DM Update 0x03 - PRL, to check availability of PRL Update
--------------------	--

8.487.2 Field Documentation

8.487.2.1 `uint32_t pack_swioa_SLQSOMADMStartSessionExt_t::sessionType`

8.488 `pack_tmd_SLQSTmdDeRegNotMitigationLvl_t` Struct Reference

Data Fields

- `uint8_t mitigationDevIDLen`
- `char mitigationDevID` [255]

8.488.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

<i>mitigationDevIDLen</i>	<ul style="list-style-type: none"> Number of sets of the following elements <ul style="list-style-type: none"> mitigation_dev_id
<i>mitigationDevID</i>	<ul style="list-style-type: none"> Mitigation device ID

8.488.2 Field Documentation

8.488.2.1 `char pack_tmd_SLQSTmdDeRegNotMitigationLvl_t::mitigationDevID`[255]

8.488.2.2 `uint8_t pack_tmd_SLQSTmdDeRegNotMitigationLvl_t::mitigationDevIDLen`

8.489 `pack_tmd_SLQSTmdGetMitigationLvl_t` Struct Reference

Data Fields

- `uint8_t mitigationDevIDLen`
- `char mitigationDevID` [255]

8.489.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

<i>mitigationDevID-Len</i>	<ul style="list-style-type: none"> Number of sets of the following elements <ul style="list-style-type: none"> mitigation_dev_id
<i>mitigationDevID</i>	<ul style="list-style-type: none"> Mitigation device ID

8.489.2 Field Documentation

8.489.2.1 char pack_tmd_SLQSTmdGetMitigationLvl_t::mitigationDevID[255]

8.489.2.2 uint8_t pack_tmd_SLQSTmdGetMitigationLvl_t::mitigationDevIDLen

8.490 pack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference

Data Fields

- uint8_t [mitigationDevIDLen](#)
- char [mitigationDevID](#) [255]

8.490.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

<i>mitigationDevID-Len</i>	<ul style="list-style-type: none"> Number of sets of the following elements <ul style="list-style-type: none"> mitigation_dev_id
<i>mitigationDevID</i>	<ul style="list-style-type: none"> Mitigation device ID

8.490.2 Field Documentation

8.490.2.1 char pack_tmd_SLQSTmdRegNotMitigationLvl_t::mitigationDevID[255]

8.490.2.2 uint8_t pack_tmd_SLQSTmdRegNotMitigationLvl_t::mitigationDevIDLen

8.491 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](#)

8.491.1 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"> • See uim_sessionInformation for more information.
<i>changePIN</i>	<ul style="list-style-type: none"> • See uim_changeUIMPIN for more information.
<i>pKeyReferenceID(optional)</i>	<ul style="list-style-type: none"> • Indicates the PIN key reference ID. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values <ul style="list-style-type: none"> – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.491.2 Field Documentation

8.491.2.1 [uim_changeUIMPIN](#) [pack_uim_ChangePin_t::changePIN](#)

8.491.2.2 [uim_encryptedPIN1](#) [pack_uim_ChangePin_t::EncryptedPIN1](#)

8.491.2.3 [uint32_t*](#) [pack_uim_ChangePin_t::pIndicationToken](#)

8.491.2.4 [uint8_t*](#) [pack_uim_ChangePin_t::pKeyReferenceID](#)

8.491.2.5 [uim_sessionInformation](#) [pack_uim_ChangePin_t::sessionInfo](#)

8.491.2.6 [uint16_t](#) [pack_uim_ChangePin_t::Tlvresult](#)

8.492 [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

8.492.1 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"> • See uim_sessionInformation for more information.
<i>fileIndex</i>	<ul style="list-style-type: none"> • See uim_fileInfo for more information.
<i>readTransparent</i>	<ul style="list-style-type: none"> • See uim_readTransparentInfo for more information.
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values <ul style="list-style-type: none"> – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function
<i>pEncrypt-Data(optional)</i>	<ul style="list-style-type: none"> • Encrypt Data. • Indicates whether the data read from the card is to be encrypted.

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.492.2 Field Documentation

8.492.2.1 [uim_fileInfo](#) pack_uim_ReadTransparent_t::fileIndex

8.492.2.2 [uint8_t*](#) pack_uim_ReadTransparent_t::pEncryptData

8.492.2.3 [uint32_t*](#) pack_uim_ReadTransparent_t::pIndicationToken

8.492.2.4 [uim_readTransparentInfo](#) pack_uim_ReadTransparent_t::readTransparent

8.492.2.5 [uim_sessionInformation](#) pack_uim_ReadTransparent_t::sessionInfo

8.492.2.6 [uint16_t](#) pack_uim_ReadTransparent_t::Tlvresult

8.493 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](#)

8.493.1 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"> • See uim_sessionInformation for more information.
<i>pinProtection</i>	<ul style="list-style-type: none"> • See uim_setPINProtection for more information.
<i>pKeyReferenceID(optional)</i>	<ul style="list-style-type: none"> • Indicates the PIN key reference ID. • Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. • This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Valid Values <ul style="list-style-type: none"> – 0 - Result of operation in response. Indication will not be generated by the modem – Any other positive number - Result of operation in indication. Indication will have same token value set by this function

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.493.2 Field Documentation

8.493.2.1 [uim_encryptedPIN1](#) [pack_uim_SetPinProtection_t::EncryptedPIN1](#)

8.493.2.2 [uint32_t*](#) [pack_uim_SetPinProtection_t::pIndicationToken](#)

8.493.2.3 [uim_setPINProtection](#) [pack_uim_SetPinProtection_t::pinProtection](#)

8.493.2.4 [uint8_t*](#) [pack_uim_SetPinProtection_t::pKeyReferenceID](#)

8.493.2.5 [uim_sessionInformation](#) [pack_uim_SetPinProtection_t::sessionInfo](#)

8.493.2.6 uint16_t pack_uim_SetPinProtection_t::Tlvresult

8.494 pack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- [uim_sessionInformation sessionInfo](#)
- [uim_authenticationData authData](#)
- uint32_t * [pIndicationToken](#)

8.494.1 Detailed Description

This structure contains information of the request parameters associated with a SLQSUIMAuthenticate.

Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none">• See uim_sessionInformation for more information.
<i>authData</i>	<ul style="list-style-type: none">• See uim_authenticationData for more information.
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none">• Response in Indication.• When this TLV is present, it indicates that the result must be provided in a subsequent indication.

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.494.2 Field Documentation

8.494.2.1 [uim_authenticationData](#) pack_uim_SLQSUIMAuthenticate_t::authData

8.494.2.2 [uint32_t*](#) pack_uim_SLQSUIMAuthenticate_t::pIndicationToken

8.494.2.3 [uim_sessionInformation](#) pack_uim_SLQSUIMAuthenticate_t::sessionInfo

8.495 pack_uim_SLQSUIDepersonalization_t Struct Reference

Data Fields

- [uim_depersionalizationInformation depersonilisationInfo](#)

8.495.1 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">• See uim_depersionalizationInformation for more information.
-------------------------------	---

8.495.2 Field Documentation

8.495.2.1 uim_depersonalizationInformation pack_uim_SLQSUIMDepersonalization_t::depersonalisationInfo

8.496 pack_uim_SLQSUIEventRegister_t Struct Reference

Data Fields

- uint32_t [eventMask](#)

8.496.1 Detailed Description

This structure contains pack event register parameter.

Parameters

<i>eventMask</i>	<ul style="list-style-type: none"> - bit 1 - card status • bit 4 - physical slot status
------------------	---

8.496.2 Field Documentation

8.496.2.1 uint32_t pack_uim_SLQSUIEventRegister_t::eventMask

8.497 pack_uim_SLQSUIGetConfiguration_t Struct Reference

Data Fields

- uint32_t * [pConfigurationMask](#)

8.497.1 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>pConfiguration-Mask(optional)</i>	<ul style="list-style-type: none"> • Requested configurations <ul style="list-style-type: none"> – Bit 0 - Automatic selection – Bit 1 - Personalization status – Bit 2 - Halt subscription – All other bits are reserved for future use
--------------------------------------	--

Note

- if the TLV is missing, the service returns all configuration items in the response.

8.497.2 Field Documentation

8.497.2.1 uint32_t* pack_uim_SLQSUIGetConfiguration_t::pConfigurationMask

8.498 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference

Data Fields

- [uim_sessionInformation](#) sessionInfo
- [uim_fileInfo](#) fileIndex
- [uint32_t](#) * [pIndicationToken](#)

8.498.1 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">• See uim_sessionInformation for more information.
<i>fileIndex</i>	<ul style="list-style-type: none">• See uim_fileInfo for more information.
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none">• Response in Indication.• When this TLV is present, it indicates that the result must be provided in a subsequent indication.

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.498.2 Field Documentation

8.498.2.1 [uim_fileInfo](#) [pack_uim_SLQSUIMGetFileAttributes_t::fileIndex](#)

8.498.2.2 [uint32_t*](#) [pack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken](#)

8.498.2.3 [uim_sessionInformation](#) [pack_uim_SLQSUIMGetFileAttributes_t::sessionInfo](#)

8.499 pack_uim_SLQSUIMPowerDown_t Struct Reference

Data Fields

- [uint8_t](#) slot

8.499.1 Detailed Description

This structure contains information of the request parameters associated with a Power Down.

Parameters

<i>slot</i>	<ul style="list-style-type: none">• Indicates the slot to be used.<ul style="list-style-type: none">– 1 - Slot 1– 2 - Slot 2
-------------	---

8.499.2 Field Documentation

8.499.2.1 `uint8_t pack_uim_SLQSUIMPowerDown_t::slot`

8.500 `pack_uim_SLQSUIMPowerUp_t` Struct Reference

Data Fields

- `uint8_t slot`
- `uint8_t * plgnoreHotSwapSwitch`

8.500.1 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"> • Indicates the slot to be used. <ul style="list-style-type: none"> – 1 - Slot 1 – 2 - Slot 2
<i>plgnoreHot-Swap-Switch(optional)</i>	<ul style="list-style-type: none"> • Hot-swap switch status. <ul style="list-style-type: none"> – 0 - Checks the hot-swap switch status – 1 - Ignores the hot-swap switch status

8.500.2 Field Documentation

8.500.2.1 `uint8_t* pack_uim_SLQSUIMPowerUp_t::plgnoreHotSwapSwitch`

8.500.2.2 `uint8_t pack_uim_SLQSUIMPowerUp_t::slot`

8.501 `pack_uim_SLQSUIMRefreshComplete_t` Struct Reference

Data Fields

- `uim_sessionInformation sessionInfo`
- `uint8_t refreshComplete`

8.501.1 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"> • See <code>uim_sessionInformation</code> for more information.
--------------------------------	---

<i>refresh-Complete(-Mandatory)</i>	<ul style="list-style-type: none"> Indicates whether the refresh was successful. Valid values: <ul style="list-style-type: none"> 0 - Refresh was not completed successfully 1 - Refresh was completed successfully
-------------------------------------	---

8.501.2 Field Documentation

8.501.2.1 uint8_t pack_uim_SLQSUIRefreshComplete_t::refreshComplete

8.501.2.2 uim_sessionInformation pack_uim_SLQSUIRefreshComplete_t::sessionInfo

8.502 pack_uim_SLQSUIRefreshGetLastEvent_t Struct Reference

Data Fields

- [uim_sessionInformation sessionInfo](#)

8.502.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Get Last Event.

Parameters

<i>sessionInfo(-Mandatory)</i>	<ul style="list-style-type: none"> See uim_sessionInformation for more information.
--------------------------------	--

8.502.2 Field Documentation

8.502.2.1 uim_sessionInformation pack_uim_SLQSUIRefreshGetLastEvent_t::sessionInfo

8.503 pack_uim_SLQSUIRefreshOK_t Struct Reference

Data Fields

- [uim_sessionInformation sessionInfo](#)
- uint8_t OKtoRefresh

8.503.1 Detailed Description

This structure contains Parameters of the Session Information

Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> Session Information See uim_sessionInformation for more information
--------------------	--

<i>OKtoRefresh</i>	<ul style="list-style-type: none"> Indicates whether a refresh is OK. Valid values: <ul style="list-style-type: none"> 0 - Not OK to refresh 1 - OK to refresh
--------------------	--

8.503.2 Field Documentation

8.503.2.1 `uint8_t pack_uim_SLQSUIRefreshOK_t::OKtoRefresh`

8.503.2.2 `uim_sessionInformation pack_uim_SLQSUIRefreshOK_t::sessionInfo`

8.504 `pack_uim_SLQSUIRefreshRegister_t` Struct Reference

Data Fields

- [uim_sessionInformation sessionInfo](#)
- [uim_registerRefresh regRefresh](#)

8.504.1 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"> Session Information params See uim_sessionInformation for more information
<i>regRefresh</i>	<ul style="list-style-type: none"> Register Refresh parameters See uim_registerRefresh for more information

8.504.2 Field Documentation

8.504.2.1 `uim_registerRefresh pack_uim_SLQSUIRefreshRegister_t::regRefresh`

8.504.2.2 `uim_sessionInformation pack_uim_SLQSUIRefreshRegister_t::sessionInfo`

8.505 `pack_uim_SLQSUISwitchSlot_t` Struct Reference

Data Fields

- `uint8_t bLogicalSlot`
- `uint32_t ulPhysicalSlot`

8.505.1 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"> Indicates the slot to be used. <ul style="list-style-type: none"> 1 - Slot 1 2 - Slot 2 3 - Slot 3 4 - Slot 4 5 - Slot 5
<i>ulPhysicalSlot</i>	<ul style="list-style-type: none"> 1 - Slot 1 2 - Slot 2 3 - Slot 3 4 - Slot 4 5 - Slot 5

8.505.2 Field Documentation

8.505.2.1 uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot

8.505.2.2 uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot

8.506 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](#)

8.506.1 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"> See uim_encryptedPIN1 for more information.
<i>sessionInfo</i>	<ul style="list-style-type: none"> See uim_sessionInformation for more information.
<i>pinProtection</i>	<ul style="list-style-type: none"> See uim_unblockUIMPIN for more information.

<i>pKeyReferenceID(optional)</i>	<ul style="list-style-type: none"> Indicates the PIN key reference ID. Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> Response in Indication. When this TLV is present, it indicates that the result must be provided in a subsequent indication. Valid Values <ul style="list-style-type: none"> 0 - Result of operation in response. Indication will not be generated by the modem Any other positive number - Result of operation in indication. Indication will have same token value set by this function

8.506.2 Field Documentation

8.506.2.1 `uim_encryptedPIN1` `pack_uim_UnblockPin_t::EncryptedPIN1`

8.506.2.2 `uint32_t*` `pack_uim_UnblockPin_t::pIndicationToken`

8.506.2.3 `uim_unblockUIMPIN` `pack_uim_UnblockPin_t::pinProtection`

8.506.2.4 `uint8_t*` `pack_uim_UnblockPin_t::pKeyReferenceID`

8.506.2.5 `uim_sessionInformation` `pack_uim_UnblockPin_t::sessionInfo`

8.506.2.6 `uint16_t` `pack_uim_UnblockPin_t::Tlvresult`

8.507 `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`

8.507.1 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"> See uim_sessionInformation for more information.
<i>verifyPIN</i>	<ul style="list-style-type: none"> See uim_verifyUIMPIN for more information.

<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> See uim_encryptedPIN1 for more information.
<i>pKeyReferenceID(optional)</i>	<ul style="list-style-type: none"> Indicates the PIN key reference ID. Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8. This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> Response in Indication. When this TLV is present, it indicates that the result must be provided in a subsequent indication. Valid Values <ul style="list-style-type: none"> 0 - Result of operation in response. Indication will not be generated by the modem Any other positive number - Result of operation in indication. Indication will have same token value set by this function

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.507.2 Field Documentation

8.507.2.1 `uim_encryptedPIN1*` `pack_uim_VerifyPin_t::pEncryptedPIN1`

8.507.2.2 `uint32_t*` `pack_uim_VerifyPin_t::pIndicationToken`

8.507.2.3 `uint8_t*` `pack_uim_VerifyPin_t::pKeyReferenceID`

8.507.2.4 `uim_sessionInformation` `pack_uim_VerifyPin_t::sessionInfo`

8.507.2.5 `uint16_t` `pack_uim_VerifyPin_t::Tlvresult`

8.507.2.6 `uim_verifyUIMPIN` `pack_uim_VerifyPin_t::verifyPIN`

8.508 pack_voice_AnswerUSSD_t Struct Reference**Data Fields**

- `uint8_t *` [pInfo](#)

8.508.1 Detailed Description

This structure contains pack anser USSD parameter.

Parameters

<i>pInfo</i>	<ul style="list-style-type: none"> USS information See voice_USSInfo for more details
--------------	---

8.508.2 Field Documentation

8.508.2.1 `uint8_t* pack_voice_AnswerUSSD_t::pInfo`

8.509 pack_voice_OriginateUSSD_t Struct Reference

Data Fields

- `uint8_t *` [pInfo](#)

8.509.1 Detailed Description

This structure contains pack originate USSD parameter.

Parameters

<i>pInfo</i>	<ul style="list-style-type: none"> • USS information • See voice_USSInfo for more details
--------------	---

8.509.2 Field Documentation

8.509.2.1 `uint8_t* pack_voice_OriginateUSSD_t::pInfo`

8.510 pack_voice_SLQSOrginateUSSD_t Struct Reference

Data Fields

- `uint8_t` [ussDCS](#)
- `uint8_t` [ussLen](#)
- `uint8_t` [ussData](#) [182]

8.510.1 Detailed Description

This structure contains USS Information

Parameters

<i>ussDCS</i>	<ul style="list-style-type: none"> • 1 - ASCII coding scheme • 2 - 8-BIT coding scheme • 3 - UCS2
<i>ussLen</i>	<ul style="list-style-type: none"> • Range 1 to 182
<i>ussData</i>	<ul style="list-style-type: none"> • Data encoded as per the DCS

8.510.2 Field Documentation

8.510.2.1 uint8_t pack_voice_SLQSOrginateUSSD_t::ussData[182]

8.510.2.2 uint8_t pack_voice_SLQSOrginateUSSD_t::ussDCS

8.510.2.3 uint8_t pack_voice_SLQSOrginateUSSD_t::ussLen

8.511 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference

Data Fields

- uint8_t [lineValue](#)

8.511.1 Detailed Description

This structure contains ALS Select Line Information Parameters.

Parameters

<i>lineValue</i>	<ul style="list-style-type: none">• ALS Line Value.<ul style="list-style-type: none">– 0x00 - ALS_LINE1 - Line 1 (default)– 0x01 - ALS_LINE2 - Line 2
------------------	--

8.511.2 Field Documentation

8.511.2.1 uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue

8.512 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference

Data Fields

- uint8_t [switchOption](#)

8.512.1 Detailed Description

This structure contains ALS Set Line Switching Information Parameters.

Parameters

<i>switchOption</i>	<ul style="list-style-type: none">• Switch Option.<ul style="list-style-type: none">– 0x00 - VOICE_LINE_SWITCHING_NOT_ALLOWED - Line switching is not allowed– 0x01 - VOICE_LINE_SWITCHING_ALLOWED - Line switching is allowed
---------------------	---

8.512.2 Field Documentation

8.512.2.1 uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption

8.513 pack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- uint8_t * [pCallId](#)

8.513.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> • Unique call identifier for the call that must be answered.
----------------	--

8.513.2 Field Documentation

8.513.2.1 uint8_t* pack_voice_SLQSVoiceAnswerCall_t::pCallId

8.514 pack_voice_SLQSVoiceBindSubscription_t Struct Reference

Data Fields

- uint8_t [subsType](#)

8.514.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

Parameters

<i>subsType</i>	<ul style="list-style-type: none"> • Subscription Type. <ul style="list-style-type: none"> – 0x00 - VOICE_SUBS_TYPE_PRIMARY - Primary – 0x01 - VOICE_SUBS_TYPE_SECONDARY - Secondary
-----------------	--

8.514.2 Field Documentation

8.514.2.1 uint8_t pack_voice_SLQSVoiceBindSubscription_t::subsType

8.515 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference

Data Fields

- [voice_burstDTMFInfo](#) BurstDTMFInfo
- [voice_DTMFLengths](#) * [pBurstDTMFLengths](#)

8.515.1 Detailed Description

This structure contains parameters of burst Dual-Tone Multifrequency (DTMF)

Parameters

<i>BurstDTMFInfo</i>	<ul style="list-style-type: none"> Burst DTMF Information <ul style="list-style-type: none"> See voice_burstDTMFInfo for more information
<i>pBurstDTMFLengths</i>	[optional] <ul style="list-style-type: none"> DTMF Lengths <ul style="list-style-type: none"> See voice_DTMFLengths for more information NULL pointer - Invalid data.

8.515.2 Field Documentation

8.515.2.1 [voice_burstDTMFInfo](#) `pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo`

8.515.2.2 [voice_DTMFLengths*](#) `pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths`

8.516 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`

8.516.1 Detailed Description

This structure contains Voice Call Request Parameters

Parameters

<i>callNumber[81]</i>	<ul style="list-style-type: none"> Number to be dialed in ASCII string, NULL terminated. Length Range [1 to 81]
<i>pCallType(optional)</i>	<ul style="list-style-type: none"> the type of call to be dialed. CALL_TYPE_VOICE is automatically selected if this parameter is not provided. When CALL_TYPE_NON_STD_OTASP is selected, the call is sent as a nonstandard OTASP call regardless of the digit string Call type values are: <ul style="list-style-type: none"> 0x00 - CALL_TYPE_VOICE - Voice (automatic selection) 0x01 - CALL_TYPE_VOICE_FORCED - Avoid modem call classification 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP* 0x09 - CALL_TYPE_EMERGENCY - Emergency

<i>pCLIR-Type(optional)</i>	<ul style="list-style-type: none"> CLIR type values are: <ul style="list-style-type: none"> 0x01 - CLIR_SUPPRESSION - Suppression 0x02 - CLIR_INVOCATION - Invocation
<i>pUUSInfo(optional)</i>	<ul style="list-style-type: none"> Pointer to structure of UUSInfo <ul style="list-style-type: none"> See voice_UUSInfo for more information
<i>pCUG-Info(optional)</i>	<ul style="list-style-type: none"> Pointer to structure of CUGInfo <ul style="list-style-type: none"> See voice_CUGInfo for more information
<i>pEmergency-Category(optional)</i>	<ul style="list-style-type: none"> Bit mask of emergency number categories. This is only applicable when the call type is set to Emergency. <ul style="list-style-type: none"> Bit 0 - VOICE_EMER_CAT_POLICE_BIT - Police Bit 1 - VOICE_EMER_CAT_AMBULANCE_BIT - Ambulance Bit 2 - VOICE_EMER_CAT_FIRE_BRIGADE_BIT - Fire brigade Bit 3 - VOICE_EMER_CAT_MARINE_GUARD_BIT - Marine guard Bit 4 - VOICE_EMER_CAT_MOUNTAIN_RESCUE_BIT - Mountain rescue Bit 5 - VOICE_EMER_CAT_MANUAL_ECALL_BIT - Manual emergency call Bit 6 - VOICE_EMER_CAT_AUTO_ECALL_BIT - Automatic emergency call Bit 7 - VOICE_EMER_CAT_SPARE_BIT - Spare bit
<i>pCallPartySub-Add(optional)</i>	<ul style="list-style-type: none"> Pointer to structure of calledPartySubAdd <ul style="list-style-type: none"> See voice_calledPartySubAdd for more information
<i>pSvc-Type(optional)</i>	<ul style="list-style-type: none"> Service Type. <ul style="list-style-type: none"> 0x01 - VOICE_DIAL_CALL_SRV_TYPE_AUTOMATIC - Automatic 0x02 - VOICE_DIAL_CALL_SRV_TYPE_GSM - GSM 0x03 - VOICE_DIAL_CALL_SRV_TYPE_WCDMA - WCDMA 0x04 - VOICE_DIAL_CALL_SRV_TYPE_CDMA_AUTOMATIC - CDMA automatic 0x05 - VOICE_DIAL_CALL_SRV_TYPE_GSM_WCDMA - GSM or WCDMA 0x06 - VOICE_DIAL_CALL_SRV_TYPE_LTE -LTE

8.516.2 Field Documentation

8.516.2.1 `uint8_t pack_voice_SLQSVoiceDialCall_t::callNumber[81]`

8.516.2.2 `voice_calledPartySubAdd* pack_voice_SLQSVoiceDialCall_t::pCallPartySubAdd`

8.516.2.3 `uint8_t* pack_voice_SLQSVoiceDialCall_t::pCallType`

8.516.2.4 `uint8_t* pack_voice_SLQSVoiceDialCall_t::pCLIRType`

8.516.2.5 `voice_CUGInfo* pack_voice_SLQSVoiceDialCall_t::pCUGInfo`

8.516.2.6 uint8_t* pack_voice_SLQSVoiceDialCall_t::pEmergencyCategory

8.516.2.7 uint8_t* pack_voice_SLQSVoiceDialCall_t::pSvcType

8.516.2.8 voice_UUSInfo* pack_voice_SLQSVoiceDialCall_t::pUUSInfo

8.517 pack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- uint8_t * [pCallId](#)

8.517.1 Detailed Description

This structure contain pack voice end call parameter.

Parameters

<i>pCallId</i>	<ul style="list-style-type: none">• Unique call identifier for the call that must be ended
----------------	--

8.517.2 Field Documentation

8.517.2.1 uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId

8.518 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference

Data Fields

- uint8_t [reason](#)
- uint8_t * [pSvcClass](#)

8.518.1 Detailed Description

This structure contains Voice Get Call Barring Pack Parameters

Parameters

<i>reason</i>	<ul style="list-style-type: none"> • Call Barring Reason • Values: <ul style="list-style-type: none"> – 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing – 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal – 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOTHOME - Outgoing external to home – 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming – 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming – 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred – 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred – 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred
<i>pSvcClass</i>	<ul style="list-style-type: none"> • Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF, if Not Available

8.518.2 Field Documentation

8.518.2.1 uint8_t* pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

8.518.2.2 uint8_t pack_voice_SLQSVoiceGetCallBarring_t::reason

8.519 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

Data Fields

- uint8_t [Reason](#)
- uint8_t * [pSvcClass](#)

8.519.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Pack Parameters

Parameters

<i>Reason</i>	<ul style="list-style-type: none"> • Call Forwarding Reason • Values: <ul style="list-style-type: none"> – 0x01 - QMI_VOICE_REASON_FWDREASON_UNCONDITIONAL - Unconditional call forwarding – 0x02 - QMI_VOICE_REASON_FWDREASON_MOBILEBUSY - Forward when the mobile is busy – 0x03 - QMI_VOICE_REASON_FWDREASON_NOREPLY - Forward when there is no reply – 0x04 - QMI_VOICE_REASON_FWDREASON_UNREACHABLE - Forward when the call is unreachable – 0x05 - QMI_VOICE_REASON_FWDREASON_ALLFORWARDING - All forwarding – 0x06 - QMI_VOICE_REASON_FWDREASON_ALLCONDITIONAL - All conditional forwarding
<i>pSvc-Class(optional)</i>	<ul style="list-style-type: none"> • Service Class is a combination (sum) of information class constants • See qaGobiApiTableSupServiceInfoClasses.h for service classes.

8.519.2 Field Documentation

8.519.2.1 uint8_t* pack_voice_SLQSVoiceGetCallForwardingStatus_t::pSvcClass

8.519.2.2 uint8_t pack_voice_SLQSVoiceGetCallForwardingStatus_t::Reason

8.520 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference

Data Fields

- uint8_t [callID](#)

8.520.1 Detailed Description

This structure contains information of the request parameters associated with a call.

Parameters

<i>callID</i>	<ul style="list-style-type: none"> • Call identifier for the call queried for information.
---------------	---

8.520.2 Field Documentation

8.520.2.1 uint8_t pack_voice_SLQSVoiceGetCallInfo_t::callID

8.521 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

Data Fields

- uint8_t * [pSvcClass](#)

8.521.1 Detailed Description

This structure contains Voice Get Call Waiting Response Parameters

Parameters

<i>pSvcClass</i>	<ul style="list-style-type: none"> • Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available
------------------	--

8.521.2 Field Documentation

8.521.2.1 uint8_t* pack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

8.522 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](#)

8.522.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

Parameters

<i>pAuto-Answer(optional)</i>	<ul style="list-style-type: none"> • Indicator to retrieve the Auto Answer Information. <ul style="list-style-type: none"> – 0x01 - Include auto answer information
<i>pAir-Timer(optional)</i>	<ul style="list-style-type: none"> • Indicator to retrieve the Air Timer Information. <ul style="list-style-type: none"> – 0x01 - Include air calls timer count information • Currently Not Supported.
<i>pRoam-Timer(optional)</i>	<ul style="list-style-type: none"> • Indicator to retrieve the Roam Timer Information. <ul style="list-style-type: none"> – 0x01 - Include roam calls timer information • Currently Not Supported.

<i>pTTY-Mode(optional)</i>	<ul style="list-style-type: none"> Indicator to retrieve the TTY Mode Information. <ul style="list-style-type: none"> 0x01 - Include TTY configuration status information
<i>pPrefVoiceSO(optional)</i>	<ul style="list-style-type: none"> Indicator to retrieve the Preferred Voice SO Information. <ul style="list-style-type: none"> 0x01 - Include preferred voice configuration status information Currently Not Supported.
<i>pAMR-Status(optional)</i>	<ul style="list-style-type: none"> Indicator to retrieve the AMR Status Information. <ul style="list-style-type: none"> 0x01 - Include AMR status information
<i>pPrefVoice-Privacy(optional)</i>	<ul style="list-style-type: none"> Indicator to retrieve the Preferred Voice Privacy Information. <ul style="list-style-type: none"> 0x01 - Include preferred voice privacy status information
<i>pNamID(optional)</i>	<ul style="list-style-type: none"> Index of the Number Assignment Module Index (CDMA subscription) to be configured Range: 0 to 3. Some modems support only 1 or 2 NAMs. The NAM Index is valid only when the request contains at least one of Air Timer, Roam Timer, and Preferred Voice SO. If no nam_id value is specified in the request, the default value is 0.
<i>pVoiceDomain-Pref(optional)</i>	<ul style="list-style-type: none"> Indicator to retrieve the Preferred Voice Domain Information. <ul style="list-style-type: none"> 0x01 - Include voice domain preference information

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.522.2 Field Documentation

8.522.2.1 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAirTimer

8.522.2.2 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAMRStatus

8.522.2.3 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAutoAnswer

8.522.2.4 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pNamID

8.522.2.5 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoicePrivacy

8.522.2.6 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoiceSO

8.522.2.7 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pRoamTimer

8.522.2.8 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pTTYMode

8.522.2.9 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pVoiceDomainPref

8.523 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference

Data Fields

- uint8_t * [pRegDTMFEvents](#)
- uint8_t * [pRegVoicePrivacyEvents](#)
- uint8_t * [pSuppsNotifEvents](#)

8.523.1 Detailed Description

This structure contains parameters of Indication Register Information

Parameters

<i>pRegDTMF-Events(optional)</i>	<ul style="list-style-type: none"> • Registration Indication For DTMF Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_DTMF_IND indication. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pRegVoice-Privacy-Events(optional)</i>	<ul style="list-style-type: none"> • Registration Indication For Voice Privacy Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_PRIVACY_IND indication. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pSuppsNotif-Events(optional)</i>	<ul style="list-style-type: none"> • Registration Indication For Supplementary Service Notification Events. • When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_SUPPS_NOTIFICATION_IND indication. <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable

Note

One of the optional parameter is mandatory to be present in the request.

8.523.2 Field Documentation

8.523.2.1 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegDTMFEvents

8.523.2.2 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegVoicePrivacyEvents

8.523.2.3 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pSuppsNotifEvents

8.524 pack_voice_SLQSVoiceManageCalls_t Struct Reference

Data Fields

- uint8_t [SUPSType](#)

- uint8_t * [pCallID](#)

8.524.1 Detailed Description

This structure contains Manage Calls Information.

Parameters

<i>SUPSType</i>	<ul style="list-style-type: none"> • Supplementary service type during the call. <ul style="list-style-type: none"> – 0x01 - SUPS_TYPE_RELEASE_HELD_OR_WAITING <ul style="list-style-type: none"> * Release is held or waiting – 0x02 - SUPS_TYPE_RELEASE_ACTIVE_ACCEPT_HELD_OR_WAITING <ul style="list-style-type: none"> * Release is active and accepting held or waiting – 0x03 - SUPS_TYPE_HOLD_ACTIVE_ACCEPT_WAITING_OR_HELD <ul style="list-style-type: none"> * Hold is active and accepting waiting or held – 0x04 - SUPS_TYPE_HOLD_ALL_EXCEPT_SPECIFIED_CALL <ul style="list-style-type: none"> * Hold all calls except a specified one – 0x05 - SUPS_TYPE_MAKE_CONFERENCE_CALL <ul style="list-style-type: none"> * Make a conference call – 0x06 - SUPS_TYPE_EXPLICIT_CALL_TRANSFER <ul style="list-style-type: none"> * Explicit call transfer – 0x07 - SUPS_TYPE_CCBS_ACTIVATION <ul style="list-style-type: none"> * Activate completion of calls to busy subscriber – 0x08 - SUPS_TYPE_END_ALL_CALLS <ul style="list-style-type: none"> * End all calls – 0x09 - SUPS_TYPE_RELEASE_SPECIFIED_CALL <ul style="list-style-type: none"> * Release a specified call
<i>pCallID[Optional]</i>	<ul style="list-style-type: none"> • Applicable only for SUPSType 0x04, 0x07, and 0x09 • NULL pointer - Invalid data.

8.524.2 Field Documentation

8.524.2.1 uint8_t* pack_voice_SLQSVoiceManageCalls_t::pCallID

8.524.2.2 uint8_t pack_voice_SLQSVoiceManageCalls_t::SUPSType

8.525 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference

Data Fields

- struct [voice_USSInfo](#) [USSInformation](#)

8.525.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

Parameters

<i>USSInformation</i>	<ul style="list-style-type: none"> • See voice_USSInfo for more information.
-----------------------	---

8.525.2 Field Documentation

8.525.2.1 struct `voice_USSInfo` `pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation`

8.526 `pack_voice_SLQSVoiceSendFlash_t` Struct Reference

Data Fields

- `uint8_t * pCallID`
- `uint8_t * pFlashPayLd`
- `uint8_t * pFlashType`

8.526.1 Detailed Description

This structure contains the flash information associated with a call.

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> • Unique call identifier associated with the current call.
<i>pFlashPayLd(optional)</i>	<ul style="list-style-type: none"> • Payload in ASCII to be sent in Flash. • Variable Length, NULL terminated.
<i>pFlashType(optional)</i>	<ul style="list-style-type: none"> • Flash type. <ul style="list-style-type: none"> – 0 - Simple Flash (default) – 1 - Activate answer hold – 2 - Deactivate answer hold

8.526.2 Field Documentation

8.526.2.1 `uint8_t*` `pack_voice_SLQSVoiceSendFlash_t::pCallID`

8.526.2.2 `uint8_t*` `pack_voice_SLQSVoiceSendFlash_t::pFlashPayLd`

8.526.2.3 `uint8_t*` `pack_voice_SLQSVoiceSendFlash_t::pFlashType`

8.527 `pack_voice_SLQSVoiceSetCallBarringPassword_t` Struct Reference

Data Fields

- `uint8_t Reason`
- `uint8_t oldPasswd` [4]
- `uint8_t newPasswd` [4]

- uint8_t newPasswdAgain [4]

8.527.1 Detailed Description

This structure contains Voice Set Call Barring Password Pack Parameters

Parameters

<i>Reason</i>	<ul style="list-style-type: none"> • Call Barring Reason • Values: <ul style="list-style-type: none"> – 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing – 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal – 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME - 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
<i>oldPasswd[BARRING_PASSWORD_LENGTH]</i>	<ul style="list-style-type: none"> • Old password. <ul style="list-style-type: none"> – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
<i>newPasswd[BARRING_PASSWORD_LENGTH]</i>	<ul style="list-style-type: none"> • New password. <ul style="list-style-type: none"> – Password consists of 4 ASCII digits. – Range: 0000 to 9999.
<i>newPasswdAgain[BARRING_PASSWORD_LENGTH]</i>	<ul style="list-style-type: none"> • New password Again. <ul style="list-style-type: none"> – Password consists of 4 ASCII digits. – Range: 0000 to 9999.

8.527.2 Field Documentation

8.527.2.1 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]

8.527.2.2 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]

8.527.2.3 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]

8.527.2.4 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason

8.528 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`

8.528.1 Detailed Description

This structure contains information about the Set Configuration Request Parameters.

Parameters

<i>pAutoAnswer</i>	<ul style="list-style-type: none"> • Value specified is written to NV_AUTO_ANSWER_I. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pAirTimerConfig</i>	<ul style="list-style-type: none"> • Value specified is written to NV_AIR_CNT_I. (optional) • See voice_airTimer for more information
<i>pRoamTimer-Config</i>	<ul style="list-style-type: none"> • Value specified is written to NV_ROAM_CNT_I. (optional) • See voice_roamTimer for more information
<i>pTTYMode</i>	<ul style="list-style-type: none"> • Value specified is written to NV_TTY_I. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - TTY_MODE_FULL - Full – 0x01 - TTY_MODE_VCO - Voice carry over – 0x02 - TTY_MODE_HCO - Hearing carry over – 0x03 - TTY_MODE_OFF - Off
<i>pPrefVoiceSO</i>	<ul style="list-style-type: none"> • Value specified is written to NV_PREF_VOICE_SO_I. (optional) • See voice_prefVoiceSO for more information
<i>pPrefVoice-Domain</i>	<ul style="list-style-type: none"> • Preferred Voice-Domain. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - VOICE_DOMAIN_PREF_CS_ONLY - Circuit-switched (CS) only – 0x01 - VOICE_DOMAIN_PREF_PS_ONLY - Packet-switched (PS) only – 0x02 - VOICE_DOMAIN_PREF_CS_PREF - CS is preferred, PS is secondary – 0x03 - VOICE_DOMAIN_PREF_PS_PREF - PS is preferred, CS is secondary

Note

One of the optional parameters must be present in the request.

8.528.2 Field Documentation

8.528.2.1 `voice_airTimer*` `pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig`

8.528.2.2 `uint8_t*` `pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer`

8.528.2.3 `uint8_t*` `pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain`

8.528.2.4 `voice_prefVoiceSO*` `pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO`

8.528.2.5 `voice_roamTimer*` `pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig`

8.528.2.6 `uint8_t*` `pack_voice_SLQSVoiceSetConfig_t::pTTYMode`

8.529 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference

Data Fields

- `uint8_t` [privacyPref](#)

8.529.1 Detailed Description

This structure contains the preferred voice privacy values.

Parameters

<i>privacyPref</i>	<ul style="list-style-type: none"> • Voice Privacy Preference <ul style="list-style-type: none"> – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy
--------------------	--

8.529.2 Field Documentation

8.529.2.1 `uint8_t` `pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref`

8.530 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](#)

8.530.1 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"> • Manages all call-independent supplementary services, such as activation, deactivation, registration, and erasure (mandatory) <ul style="list-style-type: none"> – 0x01 - VOICE_SERVICE_ACTIVATE – 0x02 - VOICE_SERVICE_DEACTIVATE – 0x03 - VOICE_SERVICE_REGISTER – 0x04 - VOICE_SERVICE_ERASE
<i>reason</i>	<ul style="list-style-type: none"> • supplementary service reason values (mandatory) <ul style="list-style-type: none"> – 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 calls are barred – 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT Outgoing internal calls are barred – 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOTHOME Outgoing calls external to home are barred – 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING All incoming calls are barred – 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING Roaming incoming calls are barred – 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
<i>pServiceClass</i>	<ul style="list-style-type: none"> • Service class is a combination (sum) of information class constants (optional) <ul style="list-style-type: none"> – See liteServiceClassInformation for more information
<i>pCallBarring-Passwd</i>	<ul style="list-style-type: none"> • Password is required if call barring is provisioned using a password. Password consists of 4 ASCII digits. Range: 0000 to 9999 (optional)

<i>pCallForwarding- Number</i>	<ul style="list-style-type: none"> • 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)
<i>pTimerVal</i>	<ul style="list-style-type: none"> • 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"> – Range: 5 to 30 in steps of 5
<i>pCallFwdType- AndPlan</i>	<ul style="list-style-type: none"> • 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"> – See voice_callFwdTypeAndPlan for more information

8.530.2 Field Documentation

8.530.2.1 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd`

8.530.2.2 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber`

8.530.2.3 `voice_callFwdTypeAndPlan* pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan`

8.530.2.4 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass`

8.530.2.5 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal`

8.530.2.6 `uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason`

8.530.2.7 `uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc`

8.531 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference

Data Fields

- `uint8_t *` [pCallID](#)
- `uint8_t` [DTMFdigit](#)

8.531.1 Detailed Description

This structure contains parameters of continuous DTMF

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> • 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. • 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. • If the call ID value received is 0, no value has been returned by the device
----------------	---

<i>DTMFdigit</i>	<ul style="list-style-type: none"> DTMF digit in ASCII.
------------------	--

8.531.2 Field Documentation

8.531.2.1 `uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit`

8.531.2.2 `uint8_t* pack_voice_SLQSVoiceStartContDTMF_t::pCallID`

8.532 `pack_voice_SLQSVoiceStopContDTMF_t` Struct Reference

Data Fields

- `uint8_t callID`

8.532.1 Detailed Description

This structure contains parameters of stop continuous DTMF

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> 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. 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. If the call ID value received is 0, no value has been returned by the device
----------------	--

8.532.2 Field Documentation

8.532.2.1 `uint8_t pack_voice_SLQSVoiceStopContDTMF_t::callID`

8.533 `pack_wds_DHCPv4ClientLeaseChange_t` Struct Reference

Data Fields

- `uint8_t * pEnableNotification`

8.533.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

Parameters

<i>pEnable-Notification</i>	<ul style="list-style-type: none"> Enable Notification or not
-----------------------------	--

8.533.2 Field Documentation

8.533.2.1 uint8_t* pack_wds_DHCPv4ClientLeaseChange_t::pEnableNotification

8.534 pack_wds_GetDefaultProfile_t Struct Reference

Data Fields

- uint32_t [profiletype](#)

8.534.1 Detailed Description

This structure contains pack get default profile information.

Parameters

<i>profiletype</i>	profile type <ul style="list-style-type: none">• 0 - WDS_PROFILE_TYPE_3GPP (0x00) - 3GPP• 1 - WDS_PROFILE_TYPE_3GPP (0x01) - 3GPP2• 2 - WDS_PROFILE_TYPE_EPC (0x02) - EPC
--------------------	---

8.534.2 Field Documentation

8.534.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.535 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [family](#)

8.535.1 Detailed Description

This structure contains pack Get Default Profile Number information.

Parameters

<i>type</i>	profile type <ul style="list-style-type: none">• 0 - 3GPP• 1 - 3GPP2
<i>family</i>	profile family <ul style="list-style-type: none">• 0 - Embedded• 1 - Tethered

8.535.2 Field Documentation

8.535.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

8.535.2.2 `uint8_t pack_wds_GetDefaultProfileNum_t::type`

8.536 `pack_wds_GetDefaultProfileV2_t` Struct Reference

Data Fields

- `uint32_t` [profiletype](#)

8.536.1 Detailed Description

This structure contains pack get default profile information.

Parameters

<i>profiletype</i>	profile type <ul style="list-style-type: none"> • 0 - WDS_PROFILE_TYPE_3GPP (0x00) - 3GPP • 1 - WDS_PROFILE_TYPE_3GPP (0x01) - 3GPP2 • 2 - WDS_PROFILE_TYPE_EPC (0x02) - EPC
--------------------	---

8.536.2 Field Documentation

8.536.2.1 `uint32_t pack_wds_GetDefaultProfileV2_t::profiletype`

8.537 `pack_wds_GetDormancyState_t` Struct Reference

8.537.1 Detailed Description

This structure contains pack get dormancy state information.

Parameters

<i>NULL</i>	<ul style="list-style-type: none"> • this is a dummy structure
-------------	---

8.538 `pack_wds_GetLastMobileIPError_t` Struct Reference

8.538.1 Detailed Description

This structure contains pack get last mobile IP error information.

Parameters

<i>NULL</i>	<ul style="list-style-type: none"> • this is a dummy structure
-------------	---

8.539 `pack_wds_GetMobileIP_t` Struct Reference

8.539.1 Detailed Description

This structure contains pack get mobile IP information.

Parameters

<i>NULL</i>	<ul style="list-style-type: none">this is a dummy structure
-------------	---

8.540 pack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t [index](#)

8.540.1 Detailed Description

This structure contains pack get mobile IP profile information.

Parameters

<i>index</i>	<ul style="list-style-type: none">Mobile IP profile ID
--------------	--

8.540.2 Field Documentation

8.540.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

8.541 pack_wds_GetPacketStatistics_t Struct Reference

Data Fields

- uint32_t * [pStatMask](#)

8.541.1 Detailed Description

This structure contains pack get packet statistics information.

Parameters

<i>pStatMask</i>	<ul style="list-style-type: none">Packet Statistics Mask<ul style="list-style-type: none">0x00000001 - Tx packets OK0x00000002 - Rx packets OK0x00000004 - Tx packet errors0x00000008 - Rx packet errors0x00000010 - Tx overflows0x00000020 - Rx overflows0x00000040 - Tx bytes OK0x00000080 - Rx bytes OK
------------------	---

8.541.2 Field Documentation

8.541.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

8.542 pack_wds_GetPacketStatus_t Struct Reference

Data Fields

- uint32_t [statmask](#)

8.542.1 Detailed Description

This structure contains pack get packet status information.

Parameters

<i>statmask</i>	packet statistics mask <ul style="list-style-type: none"> • 0x00000001 - Tx packets OK • 0x00000002 - Rx packets OK • 0x00000004 - Tx packet errors • 0x00000008 - Rx packet errors • 0x00000010 - Tx overflows • 0x00000020 - Rx overflows • 0x00000040 - Tx bytes OK • 0x00000080 - Rx bytes OK • 0x00000100 - Tx packets dropped • 0x00000200 - Rx packets dropped
-----------------	---

8.542.2 Field Documentation

8.542.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.543 pack_wds_GetSessionDuration_t Struct Reference

8.543.1 Detailed Description

This structure contains pack get session duration information.

Parameters

<i>NULL</i>	<ul style="list-style-type: none"> • this is a dummy structure
-------------	---

8.544 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- [rmTrasnferStaticsReq](#) RmTrasnferStaticsReq

8.544.1 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"> See rmTrasferStaticsReq for more information
---------------------------------	--

8.544.2 Field Documentation

8.544.2.1 `rmTrasferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasferStaticsReq`

8.545 pack_wds_SetAutoconnect_t Struct Reference

Data Fields

- uint8_t [acsetting](#)
- uint8_t [acroamsetting](#)

8.545.1 Detailed Description

auto connect data session parameters.

Parameters

<i>acsetting</i>	<ul style="list-style-type: none"> Current autoconnect setting: <ul style="list-style-type: none"> 0x00 - Autoconnect disabled 0x01 - Autoconnect enabled 0x02 - Autoconnect paused (resume on powercycle)
<i>acroamsetting</i>	<ul style="list-style-type: none"> Current autoconnect roaming status <ul style="list-style-type: none"> 0x00 - Autoconnect always allowed 0x01 - Autoconnect while in home service area only

8.545.2 Field Documentation

8.545.2.1 `uint8_t pack_wds_SetAutoconnect_t::acroamsetting`

8.545.2.2 `uint8_t pack_wds_SetAutoconnect_t::acsetting`

8.546 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](#)

8.546.1 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"> • Type of profile <ul style="list-style-type: none"> – 0 - 3GPP – 1 - 3GPP2
<i>pdpType</i>	<ul style="list-style-type: none"> • 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"> – 0 - PDP-IP (IPv4) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6
<i>ipAddress</i>	<ul style="list-style-type: none"> • Preferred IPv4 addr to be assigned to device (optional)
<i>primaryDNS</i>	<ul style="list-style-type: none"> • Primary DNS ipv4 address preference (optional)
<i>secondaryDNS</i>	<ul style="list-style-type: none"> • Secondary DNS ipv4 address preference (optional)
<i>authentication</i>	<ul style="list-style-type: none"> • Bitmap that indicates authentication algorithm preference (optional) <ul style="list-style-type: none"> – 0x00000001 - PAP preference <ul style="list-style-type: none"> * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference <ul style="list-style-type: none"> * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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.
<i>pName</i>	<ul style="list-style-type: none"> • profile Name (optional)
<i>pUsername</i>	<ul style="list-style-type: none"> • Username used during network authentication (optional)

<i>pAPNName</i>	<ul style="list-style-type: none"> Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional) If value is NULL or omitted, then subscription default value will be requested.
<i>pPassword</i>	<ul style="list-style-type: none"> Password used during network authentication (optional)

8.546.2 Field Documentation

8.546.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

8.546.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

8.546.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

8.546.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

8.546.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

8.546.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

8.546.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

8.546.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

8.546.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

8.546.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

8.547 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [family](#)
- uint8_t [index](#)

8.547.1 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"> 0 - 3GPP 1 - 3GPP2
<i>family</i>	Identifies the family of profile <ul style="list-style-type: none"> 0 - Embedded 1 - Tethered
<i>index</i>	Profile number to be set as default profile.

8.547.2 Field Documentation

8.547.2.1 `uint8_t pack_wds_SetDefaultProfileNum_t::family`

8.547.2.2 `uint8_t pack_wds_SetDefaultProfileNum_t::index`

8.547.2.3 `uint8_t pack_wds_SetDefaultProfileNum_t::type`

8.548 `pack_wds_SetMobileIP_t` Struct Reference

Data Fields

- `uint32_t mode`

8.548.1 Detailed Description

This structure contains set mobile IP pack information.

Parameters

<i>mode</i>	<ul style="list-style-type: none"> • Mobile IP setting <ul style="list-style-type: none"> – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 2 - Mobile IP only
-------------	--

8.548.2 Field Documentation

8.548.2.1 `uint32_t pack_wds_SetMobileIP_t::mode`

8.549 `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 * pHAAuthenticator`
- `uint8_t * pHA2002bis`

8.549.1 Detailed Description

Mobile IP parameters information.

Parameters

<i>pSPC</i>	<ul style="list-style-type: none"> • NULL-terminated string representing six digit service programming code.
-------------	---

<i>pMode</i>	<ul style="list-style-type: none"> Mode to be set (optional) <ul style="list-style-type: none"> 0 - Mobile IP off (simple IP only) 1 - Mobile IP preferred 2 - Mobile IP only
<i>pRetryLimit</i>	<ul style="list-style-type: none"> Registration retry attempt limit (optional)
<i>pRetryInterval</i>	<ul style="list-style-type: none"> Registration retry attempt interval used to determine the time between registration attempts (optional)
<i>pReRegPeriod</i>	<ul style="list-style-type: none"> Period (in minutes) to attempt re-registration before current registration expires (optional)
<i>pReRegTraffic</i>	<ul style="list-style-type: none"> Re-registration only if traffic since last attempt (optional) <ul style="list-style-type: none"> Zero - Disabled NonZero - Enabled
<i>pHA-Authenticator</i>	<ul style="list-style-type: none"> MH-HA authenticator calculator (optional) <ul style="list-style-type: none"> Zero - Disabled NonZero - Enabled
<i>pHA2002bis</i>	<ul style="list-style-type: none"> MH-HA RFC 2002bis authentication instead of RFC2002 (optional) <ul style="list-style-type: none"> Zero - Disabled NonZero - Enabled

8.549.2 Field Documentation

8.549.2.1 uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis

8.549.2.2 uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator

8.549.2.3 uint32_t* pack_wds_SetMobileIPParameters_t::pMode

8.549.2.4 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod

8.549.2.5 uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic

8.549.2.6 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval

8.549.2.7 uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit

8.549.2.8 char* pack_wds_SetMobileIPParameters_t::pSPC

8.550 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](#)

8.550.1 Detailed Description

Sets the mobile IP parameters.

Parameters

<i>spc</i>	<ul style="list-style-type: none"> • Six digit service programming code string
<i>index</i>	<ul style="list-style-type: none"> • Index of the profile to modify
<i>pEnabled</i>	<ul style="list-style-type: none"> • (Optional) Enable profile? 0 - Disabled Nonzero - Enabled
<i>pAddress</i>	<ul style="list-style-type: none"> • (Optional) Home IPv4 address
<i>pPrimaryHA</i>	<ul style="list-style-type: none"> • (Optional) Primary home agent IPv4 address
<i>pSecondaryHA</i>	<ul style="list-style-type: none"> • (Optional) Secondary home agent IPv4 address
<i>pRevTunneling</i>	<ul style="list-style-type: none"> • (Optional) Enable reverse tunneling? 0 - Disabled Nonzero - Enabled
<i>pNAI</i>	<ul style="list-style-type: none"> • (Optional) Network access identifier string
<i>pHASPI</i>	<ul style="list-style-type: none"> • (Optional) Home agent security parameter index
<i>pAAASPI</i>	<ul style="list-style-type: none"> • (Optional) AAA server security parameter index
<i>pMNHA</i>	<ul style="list-style-type: none"> • (Optional) MN-HA key string

<i>pMNAAA</i>	<ul style="list-style-type: none"> • (Optional) MN-AAA key string
---------------	--

8.550.2 Field Documentation

8.550.2.1 `uint8_t pack_wds_SetMobileIPProfile_t::index`

8.550.2.2 `uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI`

8.550.2.3 `uint32_t* pack_wds_SetMobileIPProfile_t::pAddress`

8.550.2.4 `uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled`

8.550.2.5 `uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI`

8.550.2.6 `int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA`

8.550.2.7 `int8_t* pack_wds_SetMobileIPProfile_t::pMNHA`

8.550.2.8 `int8_t* pack_wds_SetMobileIPProfile_t::pNAI`

8.550.2.9 `uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA`

8.550.2.10 `uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling`

8.550.2.11 `uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA`

8.550.2.12 `int8_t pack_wds_SetMobileIPProfile_t::spc[10]`

8.551 pack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

- `uint8_t * pProfileId`
- `uint8_t * pProfileType`
- `wds_profileInfo * pCurProfile`

8.551.1 Detailed Description

This structure contains pack create profile.

Parameters

<i>pProfileId</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
-------------------	---

<i>pProfileType</i>	<ul style="list-style-type: none"> Identifies the technology type of the profile <ul style="list-style-type: none"> 0x00 - 3GPP 0x01 - 3GPP2 NULL is not allowed
<i>pCurProfile</i>	<ul style="list-style-type: none"> union of 3GPP and 3GPP2 profile See wds_profileInfo

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

8.551.2 Field Documentation

8.551.2.1 `wds_profileInfo*` `pack_wds_SLQSCreateProfile_t::pCurProfile`

8.551.2.2 `uint8_t*` `pack_wds_SLQSCreateProfile_t::pProfileId`

8.551.2.3 `uint8_t*` `pack_wds_SLQSCreateProfile_t::pProfileType`

8.552 `pack_wds_SLQSDestroyProfile_t` Struct Reference**Data Fields**

- `uint8_t` [profileType](#)
- `uint8_t` [profileIndex](#)

8.552.1 Detailed Description

This structure contains the information about the profile to be deleted.

Parameters

<i>profileType</i>	<ul style="list-style-type: none"> Identifies the type of profile <ul style="list-style-type: none"> 0x00 - 3GPP Note: Deletion of 3GPP2 profiles is not supported.
<i>profileIndex</i>	<ul style="list-style-type: none"> Index of the configured profile to be deleted <ul style="list-style-type: none"> Value between 1 - 16 (EM/MC73xx or earlier) Value between 1 - 24 (EM/MC74xx onwards)

8.552.2 Field Documentation

8.552.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.552.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.553 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.553.1 Detailed Description

This structure contains pack get current data system state information.

Parameters

<i>NULL</i>	<ul style="list-style-type: none">this is a dummy structure
-------------	---

8.554 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.554.1 Detailed Description

This structure contains pack get data bearer technology information.

Parameters

<i>NULL</i>	<ul style="list-style-type: none">this is a dummy structure
-------------	---

8.555 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](#)

8.555.1 Detailed Description

This structure contains the DUN Call Info Request parameters.

Parameters

<i>Mask</i>	<ul style="list-style-type: none"> • Mandatory parameter • Set the bits corresponding to the information requested to 1 • All other bits must be set to 0. • If any values are not available or applicable, the corresponding TLVs are not returned in the response. <ul style="list-style-type: none"> – Bit 0 - Connection Status – Bit 1 - Last call end reason – Bit 2 - Tx/Rx bytes OK – Bit 3 - Dormancy status – Bit 4 - Data bearer – Bit 5 - Channel rate – Bit 6 - Call active duration
<i>pReportConn-Status</i>	<ul style="list-style-type: none"> • Connect Status Indicator <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report connection status and call end reason
<i>pTransferStatInd</i>	<ul style="list-style-type: none"> • See wds_transferStatInd for more information
<i>pReportDorm-Status</i>	<ul style="list-style-type: none"> • Dormancy Status Indicator <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report traffic channel state of interface used for data connection
<i>pReportData-BearerTech</i>	<ul style="list-style-type: none"> • Current Data Bearer Technology Indicator <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report radio interface used for data transfer when it changes
<i>pReport-ChannelRate</i>	<ul style="list-style-type: none"> • Channel Rate Indicator <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report channel rate

8.555.2 Field Documentation

8.555.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.555.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.555.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.555.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.555.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.555.2.6 wds_transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

8.556 pack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- uint8_t [ProfileId](#)
- uint8_t [ProfileType](#)

8.556.1 Detailed Description

This structure contains pack get profile settings information.

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
<i>ProfileType</i>	<ul style="list-style-type: none"> • Identifies the technology type of the profile <ul style="list-style-type: none"> – 0x00 - 3GPP – 0x01 - 3GPP2

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

8.556.2 Field Documentation

8.556.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

8.556.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

8.557 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

Data Fields

- uint32_t * [pReqSettings](#)

8.557.1 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"> • Set bits to 1, corresponding to requested information. All other bits must be set to 0. • If the values are not available, the corresponding TLVs are not returned in the response. • Absence of this mask TLV results in the device returning all of the available information corresponding to bits 0 through 12. • In cases where the information from bit 13 or greater is required, this TLV with all the necessary bits set must be present in the request. • Values <ul style="list-style-type: none"> – Bit 0 - Profile identifier – Bit 1 - Profile name – Bit 2 - PDP type – Bit 3 - APN name – Bit 4 - DNS address – Bit 5 - UMTS/GPRS granted QoS – Bit 6 - Username – Bit 7 - Authentication Protocol – Bit 8 - IP address – Bit 9 - Gateway info (address and subnet mask) – Bit 10 - PCSCF address using PCO flag – Bit 11 - PCSCF server address list – Bit 12 - PCSCF domain name list – Bit 13 - MTU – Bit 14 - domain name list – Bit 15 - IP family – Bit 16 - IM_CM flag – Bit 17 - Technology name – Bit 18 - Operator reserved PCO (Not Supported on MC/EM73xx)
---------------------	---

8.557.2 Field Documentation

8.557.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

8.558 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint8_t * [pProfileId](#)
- uint8_t * [pProfileType](#)
- [wds_profileInfo](#) curProfile

8.558.1 Detailed Description

This structure contains pack modify profile information.

Parameters

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

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

8.558.2 Field Documentation

8.558.2.1 wds_profileInfo pack_wds_SLQSMModifyProfile_t::curProfile

8.558.2.2 uint8_t* pack_wds_SLQSMModifyProfile_t::pProfileId

8.558.2.3 uint8_t* pack_wds_SLQSMModifyProfile_t::pProfileType

8.559 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](#)

8.559.1 Detailed Description

This structure contains pack set 3GPP configuration item information.

Parameters

<i>pLTEAttach-Profile</i>	<ul style="list-style-type: none"> • Optional parameter • LTE Attach Profile <ul style="list-style-type: none"> – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) • This setting is deprecated on MC/EM74xx
---------------------------	---

<i>ProfileList</i>	Profile List <ul style="list-style-type: none"> • an array of 4 profile configurations • Each element points to a single WORD value indicating profile • Optional parameter with possible values <ul style="list-style-type: none"> – 1 - 16 (MC/EM73xx and before) – 1 - 24 (MC/EM74xx and onwards) • function SLQSSet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
<i>pDefaultPDN-Enabled</i>	<ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – 0 - disabled – 1 - enabled
<i>p3gppRelease</i>	3GPP release <ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – 0 - Release_99 – 1 - Release_5 – 2 - Release_6 – 3 - Release_7 – 4 - Release_8 • In 9x30 and onwards <ul style="list-style-type: none"> – 5 - Release 9 – 6 - Release 10 – 7 - Release 11
<i>pLTEAttach-ProfileList</i>	<ul style="list-style-type: none"> • pointer to WORD array indicating LTE Attach Profile List <ul style="list-style-type: none"> – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – Please provide attach profiles in order of decreasing priority in this list.
<i>LTEAttach-ProfileListLen</i>	<ul style="list-style-type: none"> • Number of element in pLTEAttachProfileList <ul style="list-style-type: none"> – valid range: 1-24 – This setting is only supported for MC/EM74xx onwards

8.559.2 Field Documentation

8.559.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.559.2.2 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease

8.559.2.3 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled

8.559.2.4 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile

8.559.2.5 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList

8.559.2.6 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList

8.560 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint8_t [IPFamilyPreference](#)

8.560.1 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"> • PACK_WDS_IPV4 IP Version 4 • PACK_WDS_IPV6 IP Version 6
----------------------------	---

8.560.2 Field Documentation

8.560.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

8.561 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](#)

8.561.1 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

8.561.2 Field Documentation

8.561.2.1 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer`

8.561.2.2 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer`

8.561.2.3 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearerTechExt`

8.561.2.4 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus`

8.561.2.5 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus`

8.561.2.6 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval`

8.561.2.7 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP`

8.561.2.8 `uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats`

8.562 `pack_wds_SLQSSetDHCPv4ClientConfig_t` Struct Reference

Data Fields

- [wdsDhcpv4ProfileId](#) * `pProfileId`

8.562.1 Detailed Description

This structure contain get DHCPv4 client configure.

Parameters

<i>pProfileId</i>	pointer to Profile Id structure <ul style="list-style-type: none"> • See wdsDhcpv4ProfileId
-------------------	--

8.562.2 Field Documentation

8.562.2.1 `wdsDhcpv4ProfileId* pack_wds_SLQSSetDHCPv4ClientConfig_t::pProfileId`

8.563 `pack_wds_SLQSSetDHCPv4ClientConfig_t` Struct Reference

Data Fields

- `wds_DHCPv4ProfileId` * `pProfileId`
- `wds_DHCPv4HWConfig` * `pHwConfig`
- `wds_DHCPv4OptionList` * `pRequestOptionList`

8.563.1 Detailed Description

WDS SWI DHCPv4 Config Structure

Parameters

<i>pProfileId</i>	<ul style="list-style-type: none"> • pointer to Profile Id structure
<i>pHWConfig</i>	<ul style="list-style-type: none"> • pointer to HW Config structure
<i>pRequestOption-List</i>	<ul style="list-style-type: none"> • pointer to Option List structure to be sent in DHCP request

8.563.2 Field Documentation

8.563.2.1 wds_DHCPv4HWConfig* pack_wds_SLQSSSetDHCPv4ClientConfig_t::pHwConfig

8.563.2.2 wds_DHCPv4ProfileId* pack_wds_SLQSSSetDHCPv4ClientConfig_t::pProfileId

8.563.2.3 wds_DHCPv4OptionList* pack_wds_SLQSSSetDHCPv4ClientConfig_t::pRequestOptionList

8.564 pack_wds_SLQSSSetLoopback_t Struct Reference

Data Fields

- uint8_t [loopbackMode](#)
- uint8_t [loopbackMultiplier](#)

8.564.1 Detailed Description

This structure contains pack set loopback information.

Parameters

<i>loopbackMode</i>	<ul style="list-style-type: none"> • Loopback Mode. <ul style="list-style-type: none"> – 0 - Disable – 1 - Enable
<i>loopback-Multiplier</i>	<ul style="list-style-type: none"> • Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.564.2 Field Documentation

8.564.2.1 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode

8.564.2.2 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier

8.565 pack_wds_SLQSSStartDataSession_t Struct Reference

Data Fields

- uint8_t * [pTech](#)

- uint32_t * [pprofileid3gpp](#)
- uint32_t * [pprofileid3gpp2](#)
- uint32_t * [pAuth](#)
- char * [pUser](#)
- char * [pPass](#)

8.565.1 Detailed Description

This structure contains pack Start Data Session Information.

Parameters

<i>pTech</i>	<ul style="list-style-type: none"> • Indicates the technology preference <ul style="list-style-type: none"> – 1 - UMTS – 2 - CDMA – 3 - eMBMS – 4 - Modem Link Label. Modem Link is an interface for transferring data between entities on AP and modem. • optional
<i>pprofileid3gpp</i>	<ul style="list-style-type: none"> • pointer to 3GPP profile id • optional
<i>pprofileid3gpp2</i>	<ul style="list-style-type: none"> • pointer to 3GPP2 profile id • optional
<i>pAuth</i>	<ul style="list-style-type: none"> • Authentication type, it can be PAP or CHAP • optional
<i>pUser</i>	<ul style="list-style-type: none"> • username for authentication process • optional
<i>pPass</i>	<ul style="list-style-type: none"> • password for authentication process • optional

8.565.2 Field Documentation

8.565.2.1 uint32_t* [pack_wds_SLQSStartDataSession_t::pAuth](#)

8.565.2.2 char* [pack_wds_SLQSStartDataSession_t::pPass](#)

8.565.2.3 uint32_t* [pack_wds_SLQSStartDataSession_t::pprofileid3gpp](#)

8.565.2.4 uint32_t* [pack_wds_SLQSStartDataSession_t::pprofileid3gpp2](#)

8.565.2.5 uint8_t* [pack_wds_SLQSStartDataSession_t::pTech](#)

8.565.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

8.566 pack_wds_SLQSStopDataSession_t Struct Reference

Data Fields

- uint32_t * [psid](#)

8.566.1 Detailed Description

This structure contains pack stop data session information.

Parameters

<i>psid</i>	session id
-------------	------------

8.566.2 Field Documentation

8.566.2.1 uint32_t* pack_wds_SLQSStopDataSession_t::psid

8.567 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](#)

8.567.1 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"> • Current Channel Rate Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report channel rate when it changes
<i>pTransferStatInd</i>	(optional) <ul style="list-style-type: none"> • See wds_TrStatInd for more information.
<i>pDataBearer-TechInd</i>	(optional) <ul style="list-style-type: none"> • Data Bearer Technology Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report radio interface used for data transfer when it changes

<i>pDormancy-StatusInd</i>	(optional) <ul style="list-style-type: none"> • Dormancy Status indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report traffic channel state of interface used for data connection
<i>pMIPStatusInd</i>	(optional) <ul style="list-style-type: none"> • MIP Status Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report MIP status
<i>pCurrData-BearerTechInd</i>	(optional) <ul style="list-style-type: none"> • Current Data Bearer Technology Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report current data bearer technology when it changes
<i>pDataCallStatus-ChangeInd</i>	(optional) <ul style="list-style-type: none"> • Data Call Status Change Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report data call status change when it changes
<i>pCurrPrefData-SysInd</i>	(optional) <ul style="list-style-type: none"> • Current Preferred Data System Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report preferred data system when it changes
<i>pEVDOPage-MonPerChange-Ind</i>	(optional) <ul style="list-style-type: none"> • EV-DO Page Monitor Period Change Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report EV-DO page monitor period change event
<i>pDataSystem-StatusChange-Ind</i>	(optional) <ul style="list-style-type: none"> • Data System Status Change Indicator. <ul style="list-style-type: none"> – 0 - Do not report – 1 - Report data system status change event

Note

At least one parameter should be present.

8.567.2 Field Documentation

8.567.2.1 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd`

8.567.2.2 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd`

8.567.2.3 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd`

8.567.2.4 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd`

8.567.2.5 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd`

8.567.2.6 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd

8.567.2.7 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd

8.567.2.8 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd

8.567.2.9 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd

8.567.2.10 wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd

8.568 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t [contextId](#)
- uint8_t [contextType](#)

8.568.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

Parameters

<i>contextId</i>	<ul style="list-style-type: none"> • Context Identifier
<i>contextType</i>	<ul style="list-style-type: none"> • Identifies technology type <ul style="list-style-type: none"> – 0 - 3GPP – 1 - 3GPP2

8.568.2 Field Documentation

8.568.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.568.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

8.569 PackCreateProfileOut Struct Reference

Data Fields

- uint8_t [ProfileType](#)
- uint8_t [ProfileIndex](#)
- uint16_t [ExtErrorCode](#)

8.569.1 Detailed Description

This structure contains out parameter Information

Parameters

<i>ProfileType</i>	<ul style="list-style-type: none"> Identifies the type of profile 0x00 = 3GPP 0x01 = 3GPP2 Bit to check in ParamPresenceMask - 1
<i>ProfileIndex</i>	<ul style="list-style-type: none"> Index identifying the profile that was created Bit to check in ParamPresenceMask - 1
<i>ExtErrorCode</i>	<ul style="list-style-type: none"> The extended error code received from DS Profile subsystem Bit to check in ParamPresenceMask - 224

8.569.2 Field Documentation

8.569.2.1 uint16_t PackCreateProfileOut::ExtErrorCode

8.569.2.2 uint8_t PackCreateProfileOut::ProfileIndex

8.569.2.3 uint8_t PackCreateProfileOut::ProfileType

8.570 packgetDyingGaspCfg Struct Reference

Data Fields

- uint8_t * [pDestSMSNum](#)
- uint8_t * [pDestSMSContent](#)

8.570.1 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"> SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars. Optional parameter.
<i>pDestSMS-Content</i> [IN]	<ul style="list-style-type: none"> SMS CContent as a string of 8 bit ASCII text characters Max 160 chars. Optional parameter.

8.570.2 Field Documentation

8.570.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.570.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.571 packgetDyingGaspStatistics Struct Reference

Data Fields

- uint32_t * [pTimeStamp](#)
- uint8_t * [pSMSAttemptedFlag](#)

8.571.1 Detailed Description

This structure contains Get Dying GASP Statistics

Parameters

<i>pTimeStamp[O-UT]</i>	<ul style="list-style-type: none"> • Timestamp of the last time power loss was detected and Dying Gasp feature was triggered. • UTC time in seconds since Jan 06, 1980 (GPS Epoch).
<i>pSMS-AttemptedFlag[OUT]</i>	<ul style="list-style-type: none"> • Indicates whether device attempted to send SMS in the last power loss event. • 0 - SMS not attempted • 1 - SMS attempted • This only indicates device sent the SMS but does not guarantee network delivery.

8.571.2 Field Documentation

8.571.2.1 uint8_t* `packgetDyingGaspStatistics::pSMSAttemptedFlag`

8.571.2.2 uint32_t* `packgetDyingGaspStatistics::pTimeStamp`

8.572 PackSwiAvmsSetSettingsAPNInfo Struct Reference

Data Fields

- uint8_t [bAPNLength](#)
- uint8_t [szAPN](#) [49]
- uint8_t [bUsernameLength](#)
- uint8_t [szUsername](#) [29]
- uint8_t [bPWDLength](#)
- uint8_t [szPWD](#) [29]

8.572.1 Detailed Description

This structure contains pack set settings APN information.

Parameters

<i>bAPNLength</i>	APN Length.
<i>szAPN</i>	APN String.
<i>bUsernameLength</i>	Username Length.
<i>szUsername</i>	User Name String.
<i>bPWDLength</i>	Password Length.
<i>szPWD</i>	Password String.

8.572.2 Field Documentation

8.572.2.1 uint8_t PackSwiAvmsSetSettingsAPNInfo::bAPNLength

8.572.2.2 uint8_t PackSwiAvmsSetSettingsAPNInfo::bPWDLength

8.572.2.3 uint8_t PackSwiAvmsSetSettingsAPNInfo::bUnameLength

8.572.2.4 uint8_t PackSwiAvmsSetSettingsAPNInfo::szAPN[49]

8.572.2.5 uint8_t PackSwiAvmsSetSettingsAPNInfo::szPWD[29]

8.572.2.6 uint8_t PackSwiAvmsSetSettingsAPNInfo::szUname[29]

8.573 PackSwiAvmsSetSettingsConnectionRetryTimers Struct Reference

Data Fields

- uint16_t [Timers](#) [8]

8.573.1 Detailed Description

This structure contains the SLQSAVMSSetSettings Polling timer to connect to AVMS server.

Parameters

<i>Timers[0]</i>	- Timer 1. • 0-20160 (min) • 0:retry disabled
<i>Timers[1]</i>	- Timer 2. • 1-20160 (min)
<i>Timers[2]</i>	- Timer 3. • 1-20160 (min)
<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)

8.573.2 Field Documentation

8.573.2.1 `uint16_t` PackSwiAvmsSetSettingsConnectionRetryTimers::Timers[8]

8.574 PackSwiAvmsSetSettingsPeriodInfo Struct Reference

Data Fields

- `uint32_t` [ulMin](#)
- `uint32_t` [ulMax](#)

8.574.1 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.

8.574.2 Field Documentation

8.574.2.1 `uint32_t` PackSwiAvmsSetSettingsPeriodInfo::ulMax

8.574.2.2 `uint32_t` PackSwiAvmsSetSettingsPeriodInfo::ulMin

8.575 PackSwiAVMSSettingsAPNInfo Struct Reference

Data Fields

- `uint8_t` [bAPNLength](#)
- `uint8_t *` [pAPN](#)
- `uint8_t` [bUnameLength](#)
- `uint8_t *` [pUname](#)
- `uint8_t` [bPWDLength](#)
- `uint8_t *` [pPWD](#)

8.575.1 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.

8.575.2 Field Documentation

8.575.2.1 `uint8_t` PackSwiAVMSSettingsAPNInfo::bAPNLength

- 8.575.2.2 `uint8_t` `PackSwiAVMSSettingsAPNInfo::bPWDLenght`
- 8.575.2.3 `uint8_t` `PackSwiAVMSSettingsAPNInfo::bUnameLength`
- 8.575.2.4 `uint8_t*` `PackSwiAVMSSettingsAPNInfo::pAPN`
- 8.575.2.5 `uint8_t*` `PackSwiAVMSSettingsAPNInfo::pPWD`
- 8.575.2.6 `uint8_t*` `PackSwiAVMSSettingsAPNInfo::pUname`

8.576 `PackSwiAVMSSettingsConnectionRetryTimers` Struct Reference

Data Fields

- `uint16_t` [Timers](#) [8]

8.576.1 Detailed Description

This structure contains the SLQSAVMSGetSettings Polling timer to connect to AVMS server.

Parameters

<i>Timers[0]</i>	- Timer 1. • 0-20160 (min) • 0:retry disabled
<i>Timers[1]</i>	- Timer 2. • 1-20160 (min)
<i>Timers[2]</i>	- Timer 3. • 1-20160 (min)
<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)

8.576.2 Field Documentation

- 8.576.2.1 `uint16_t` `PackSwiAVMSSettingsConnectionRetryTimers::Timers[8]`

8.577 PackSwiAVMSSettingsPeriodsInfo Struct Reference

Data Fields

- uint32_t [min](#)
- uint32_t [max](#)

8.577.1 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.

8.577.2 Field Documentation

8.577.2.1 uint32_t PackSwiAVMSSettingsPeriodsInfo::max

8.577.2.2 uint32_t PackSwiAVMSSettingsPeriodsInfo::min

8.578 qmiSmsMessageList Struct Reference

Data Fields

- uint32_t [messageIndex](#)
- uint32_t [messageTag](#)

8.578.1 Detailed Description

This structure contains SMS message list.

Parameters

<i>messageIndex</i>	<ul style="list-style-type: none">• Message index of each matched message
<i>messageTag</i>	<ul style="list-style-type: none">• Messagetag

8.578.2 Field Documentation

8.578.2.1 uint32_t qmiSmsMessageList::messageIndex

8.578.2.2 uint32_t qmiSmsMessageList::messageTag

8.579 qmiWSDDataBearerTechnology Struct Reference

Data Fields

- uint8_t [currentNetwork](#)
- uint32_t [ratMask](#)
- uint32_t [soMask](#)

8.579.1 Detailed Description

Structure to hold the current data bearer technology values

Parameters

<i>currentNetwork[OUT]</i>	<ul style="list-style-type: none"> • current selected network <ul style="list-style-type: none"> – 0 - UNKNOWN – 1 - 3GPP2 – 2 - 3GPP – 0xff - Invalid data.
<i>ratMask[OUT]</i>	RAT Mask
<i>soMask[OUT]</i>	SO Mask

8.579.2 Field Documentation

8.579.2.1 uint8_t qmiWSDDataBearerTechnology::currentNetwork

8.579.2.2 uint32_t qmiWSDDataBearerTechnology::ratMask

8.579.2.3 uint32_t qmiWSDDataBearerTechnology::soMask

8.580 qmTlvResult Struct Reference

Data Fields

- [swi_uint256_t](#) TlvPresenceMask
- [qmulong](#) TlvResultCode
- [qmuint16](#) DeviceResult
- [qmuint16](#) DeviceError

8.580.1 Field Documentation

8.580.1.1 qmuint16 qmTlvResult::DeviceError

8.580.1.2 qmuint16 qmTlvResult::DeviceResult

8.580.1.3 swi_uint256_t qmTlvResult::TlvPresenceMask

8.580.1.4 qmulong qmTlvResult::TlvResultCode

8.581 qos_BindDataPortMuxID_t Struct Reference

Data Fields

- uint8_t [MuxID](#)

8.581.1 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">• 0x80-0x8F : valid value
--------------	---

8.581.2 Field Documentation

8.581.2.1 uint8_t qos_BindDataPortMuxID_t::MuxID

8.582 qos_BindDataPortPeripheralEndPointID_t Struct Reference

Data Fields

- uint32_t [EndPointType](#)
- uint32_t [IfaceID](#)

8.582.1 Detailed Description

Structure that contains the request Binds a control point to a data port Periapheral End Point Type data.

Parameters

<i>EndPointType</i>	Peripheral end point type. Values: <ul style="list-style-type: none">• 0 - Reserved• 1 - HSIC• 2 - HSUSB• 3 - PCIE• 4 - Embedded
<i>IfaceID</i>	Peripheral interface number.

8.582.2 Field Documentation

8.582.2.1 uint32_t qos_BindDataPortPeripheralEndPointID_t::EndPointType

8.582.2.2 uint32_t qos_BindDataPortPeripheralEndPointID_t::IfaceID

8.583 qos_BindDataPortSIODDataPort_t Struct Reference

Data Fields

- uint16_t [SIODDataPort](#)

8.583.1 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
-----------------------	---

8.583.2 Field Documentation

8.583.2.1 `uint16_t qos_BindDataPortSIODataPort_t::SIODataPort`

8.584 RFBandInfoElements Struct Reference

Data Fields

- `uint8_t` [radioInterface](#)
- `uint16_t` [activeBandClass](#)
- `uint16_t` [activeChannel](#)

8.584.1 Detailed Description

This structure contains the RFBandInfo response parameters.

Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio interface technology <ul style="list-style-type: none"> – See Tables for Radio Interface
<i>activeBandClass</i>	<ul style="list-style-type: none"> • Active Band Class <ul style="list-style-type: none"> – See Tables for Band Classes
<i>activeChannel</i>	<ul style="list-style-type: none"> • Active channel (0 if channel is not relevant to the reported technology)

8.584.2 Field Documentation

8.584.2.1 `uint16_t RFBandInfoElements::activeBandClass`

8.584.2.2 `uint16_t RFBandInfoElements::activeChannel`

8.584.2.3 `uint8_t RFBandInfoElements::radioInterface`

8.585 rmTrasnferStaticsReq Struct Reference

Data Fields

- `uint8_t` [bResetStatistics](#)
- `uint32_t` [ulMask](#)

8.585.1 Detailed Description

RM Transfer Statistics Structure

Parameters

<i>bResetStatistics</i>	<ul style="list-style-type: none"> Reset Statistics Values: 0 - Not Reset Other - Reset
<i>ulMask</i>	<ul style="list-style-type: none"> Enable/Disable RM Transfer Statistics Indiscation Mask Bit 0: Tx Packet Ok Bit 1: Rx Packet Ok Bit 2: Tx Bytes Ok Bit 3: Rx Bytes Ok Bit 4: Tx Packets Dropped Bit 5: Rx Packets Dropped Value: -0 - Disable -1 - Enable

8.585.2 Field Documentation

8.585.2.1 uint8_t rmTransferStaticsReq::bResetStatistics

8.585.2.2 uint32_t rmTransferStaticsReq::ulMask

8.586 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]

8.586.1 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>timeOfFirstSample</i>	<ul style="list-style-type: none"> Denotes a full 32-bit time stamp of the first (oldest) sample in this message. The time stamp is in the time reference scale that is used by the sensor time source. Units - Milliseconds
--------------------------	---

<i>flags</i>	<ul style="list-style-type: none"> Flags to indicate any deviation from the default measurement assumptions. All unused bits in this field must be set to 0. Valid bitmasks <ul style="list-style-type: none"> 0x01 - Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples 0x02 - Bitmask to specify that the sensor time stamp is the same as the modem time stamp
<i>sensorDataLen</i>	<ul style="list-style-type: none"> Number of sets of the following elements <ul style="list-style-type: none"> timeOffset xAxis yAxis zAxis
<i>timeOffset</i>	<ul style="list-style-type: none"> Sample time offset Units - Milliseconds
<i>xAxis</i>	<ul style="list-style-type: none"> Sensor x-axis sample. Units Accelerometer - Meters/seconds square Units Gyroscope - Radians/second
<i>yAxis</i>	<ul style="list-style-type: none"> Sensor Y-axis sample. Units Accelerometer - Meters/seconds square Units Gyroscope - Radians/second
<i>zAxis</i>	<ul style="list-style-type: none"> Sensor Z-axis sample. Units Accelerometer - Meters/seconds square Units Gyroscope - Radians/second

8.586.2 Field Documentation

8.586.2.1 `uint8_t sensorData_t::flags`

8.586.2.2 `uint8_t sensorData_t::sensorDataLen`

8.586.2.3 `uint32_t sensorData_t::timeOfFirstSample`

8.586.2.4 `uint16_t sensorData_t::timeOffset[64]`

8.586.2.5 `uint32_t sensorData_t::xAxis[64]`

8.586.2.6 `uint32_t sensorData_t::yAxis[64]`

8.586.2.7 `uint32_t sensorData_t::zAxis[64]`

8.587 slot_t Struct Reference

Data Fields

- uint32_t [uPhyCardStatus](#)
- uint32_t [uPhySlotStatus](#)
- uint8_t [bLogicalSlot](#)
- uint8_t [bICCIDLength](#)
- uint8_t [bICCID](#) [255]

8.587.1 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"> • State of the card in the Pyhsical Slot Status. <ul style="list-style-type: none"> – 0x00 - Unknown. – 0x01 - Absent. – 0x02 - Present.
<i>uPhySlotStatus</i>	<ul style="list-style-type: none"> • State of the Physical Slot status. <ul style="list-style-type: none"> – 0x00 Inactive. – 0x01 Activate.
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> • Logical Slot associated with this physical slot. THis is valid if the physical slot is active. <ul style="list-style-type: none"> – 1 - Slot 1. – 2 - Slot 2. – 3 - Slot 3. – 4 - Slot 4. – 5 - Slot 5.
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> • Number of sets the sets of ICCID
<i>bICCID[MAX_ICCID_LENGTH]</i>	<ul style="list-style-type: none"> • Contains the ICCID of the card in the physical slot.

8.587.2 Field Documentation

8.587.2.1 uint8_t slot_t::bICCID[255]

8.587.2.2 uint8_t slot_t::bICCIDLength

8.587.2.3 uint8_t slot_t::bLogicalSlot

8.587.2.4 uint32_t slot_t::uPhyCardStatus

8.587.2.5 uint32_t slot_t::uPhySlotStatus

8.588 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]

8.588.1 Detailed Description

This structure contains information about the SLOTS present.

Parameters

<i>cardState</i>	<ul style="list-style-type: none"> • Indicates the state of the card for each slot. <ul style="list-style-type: none"> – 0 - Absent – 1 - Present – 2 - Error
<i>upinState</i>	<ul style="list-style-type: none"> • Indicates the state of UPIN. <ul style="list-style-type: none"> – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked – 0xFF - Not Available
<i>upinRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to verify the UPIN. • If 0xFF, information not available.
<i>upukRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock the UPIN. • If 0xFF, information not available.

<i>errorState</i>	<ul style="list-style-type: none"> Indicates the reason for the card error, and is valid only when the card state is Error <ul style="list-style-type: none"> 0 - Unknown 1 - Power down 2 - Poll error 3 - No ATR received 4 - Volt mismatch 5 - Parity error 6 - Unknown; possibly removed 7 - Card returned technical problems 0xFF - Not Available Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as "Unknown".
<i>numApp</i>	<ul style="list-style-type: none"> Indicates the number of applications available on the card. The following block is repeated for each application. i.e. AppStatus. If zero(0) then no AppStatus information exists.
<i>AppStatus[MAX_NO_OF_APPLICATIONS]</i>	<ul style="list-style-type: none"> See appStats for more information.

8.588.2 Field Documentation

8.588.2.1 `appStats slotInf::AppStatus[10]`

8.588.2.2 `uint8_t slotInf::cardState`

8.588.2.3 `uint8_t slotInf::errorState`

8.588.2.4 `uint8_t slotInf::numApp`

8.588.2.5 `uint8_t slotInf::upinRetries`

8.588.2.6 `uint8_t slotInf::upinState`

8.588.2.7 `uint8_t slotInf::upukRetries`

8.589 slots_t Struct Reference

Data Fields

- [slot_t uimSlotStatus](#) [255]

8.589.1 Detailed Description

This structure contains slots status.

Parameters

<i>uimSlotStatus</i>	<ul style="list-style-type: none"> • See slot_t
----------------------	--

8.589.2 Field Documentation

8.589.2.1 `slot_t slots_t::uimSlotStatus[255]`8.590 `sms_BroadcastConfig` Struct Reference

Data Fields

- `uint16_t` [fromServiceId](#)
- `uint16_t` [toServiceId](#)
- `uint8_t` [selected](#)

8.590.1 Detailed Description

This structure contains BroadcastConfig parameters

Parameters

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

8.590.2 Field Documentation

8.590.2.1 `uint16_t sms_BroadcastConfig::fromServiceId`8.590.2.2 `uint8_t sms_BroadcastConfig::selected`8.590.2.3 `uint16_t sms_BroadcastConfig::toServiceId`8.591 `sms_CDMABroadcastConfig` Struct Reference

Data Fields

- `uint16_t` [serviceCategory](#)
- `uint16_t` [language](#)
- `uint8_t` [selected](#)

8.591.1 Detailed Description

This structure contains CDMABroadcastConfig parameters

Parameters

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

8.591.2 Field Documentation

8.591.2.1 uint16_t sms_CDMABroadcastConfig::language

8.591.2.2 uint8_t sms_CDMABroadcastConfig::selected

8.591.2.3 uint16_t sms_CDMABroadcastConfig::serviceCategory

8.592 sms_getIndicationReg Struct Reference

Data Fields

- uint8_t * [pRegTransLayerInfoEvt](#)
- uint8_t * [pRegTransNWRegInfoEvt](#)
- uint8_t * [pRegCallStatInfoEvt](#)

8.592.1 Detailed Description

This structure contains Get Indication Register Response parameters

Parameters

<i>pRegTransLayerInfoEvt</i>	<ul style="list-style-type: none"> - • Optional parameter indicating registration status of transport layer information events • Values: <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled
<i>pRegTransNWRegInfoEvt</i>	<ul style="list-style-type: none"> - • Optional parameter indicating registration status of transport network registration information events • Values: <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled

<i>pRegCallStat-InfoEvt</i>	- <ul style="list-style-type: none"> Optional parameter indicating registration status of call status information events Values: <ul style="list-style-type: none"> 0x00 - Disabled 0x01 - Enabled
-----------------------------	---

8.592.2 Field Documentation

8.592.2.1 `uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt`

8.592.2.2 `uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt`

8.592.2.3 `uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt`

8.593 sms_getMsgWaitingInfo Struct Reference

Data Fields

- `uint8_t numInstances`
- `sms_messageWaitingInfoContent msgWaitInfo` [0xFF]

8.593.1 Detailed Description

This structure contains Get Message Waiting Info Response parameters

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> Number of sets of the elements in structure <code>sms_messageWaitingInfoContent</code>
<i>pMsgWaitInfo</i>	<ul style="list-style-type: none"> Pointer to structure of <code>sms_messageWaitingInfoContent</code>. <ul style="list-style-type: none"> See <code>sms_messageWaitingInfoContent</code> for more information.

8.593.2 Field Documentation

8.593.2.1 `sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo`[0xFF]

8.593.2.2 `uint8_t sms_getMsgWaitingInfo::numInstances`

8.594 sms_getTransLayerInfo Struct Reference

Data Fields

- `uint8_t * pRegInd`
- `sms_transLayerInfo * pTransLayerInfo`

8.594.1 Detailed Description

This structure contains Get Transport Layer Info Response parameters

Parameters

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

8.594.2 Field Documentation

8.594.2.1 `uint8_t* sms_getTransLayerInfo::pRegInd`

8.594.2.2 `sms_transLayerInfo* sms_getTransLayerInfo::pTransLayerInfo`

8.595 sms_getTransNWRegInfo Struct Reference

Data Fields

- `uint8_t * pRegStatus`

8.595.1 Detailed Description

This structure contains transport network registration info parameter

Parameters

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

8.595.2 Field Documentation

8.595.2.1 `uint8_t* sms_getTransNWRegInfo::pRegStatus`

8.596 sms_maxStorageSizeReq Struct Reference

Data Fields

- `uint8_t` [storageType](#)
- `uint8_t *` [pMessageMode](#)

8.596.1 Detailed Description

This structure contains get store max size request parameters

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> • SMS message storage type <ul style="list-style-type: none"> – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV
<i>pMessage-Mode(optional)</i>	parameter) <ul style="list-style-type: none"> • 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

Note

The Message Mode TLV must be included if the device is capable of supporting more than one protocol

8.596.2 Field Documentation

8.596.2.1 `uint8_t* sms_maxStorageSizeReq::pMessageMode`

8.596.2.2 `uint8_t sms_maxStorageSizeReq::storageType`

8.597 sms_maxStorageSizeResp Struct Reference

Data Fields

- `uint32_t` [maxStorageSize](#)
- `uint32_t` [freeSlots](#)

8.597.1 Detailed Description

This structure contains get store max size response parameters

Parameters

<i>maxStorageSize</i>	- <ul style="list-style-type: none"> • Memory Store Size
-----------------------	---

<i>freeSlots</i>	- <ul style="list-style-type: none"> Optional parameter indicating how much Memory is available returns a default value 0xFFFFFFFF for parameter values if no response is received from the device.
------------------	---

8.597.2 Field Documentation

8.597.2.1 uint32_t sms_maxStorageSizeResp::freeSlots

8.597.2.2 uint32_t sms_maxStorageSizeResp::maxStorageSize

8.598 sms_messageWaitingInfoContent Struct Reference

Data Fields

- uint8_t [msgType](#)
- uint8_t [activeInd](#)
- uint8_t [msgCount](#)

8.598.1 Detailed Description

This structure contains message waiting information per instance

Parameters

<i>msgType</i>	<ul style="list-style-type: none"> Message type <ul style="list-style-type: none"> 0x00 - MWI_MESSAGE_TYPE_VOICMAIL - Voicemail 0x01 - MWI_MESSAGE_TYPE_FAX - Fax 0x02 - MWI_MESSAGE_TYPE_EMAIL - Email 0x03 - MWI_MESSAGE_TYPE_OTHER - Other 0x04 - MWI_MESSAGE_TYPE_VIDEOMAIL - Videomail
<i>activeInd</i>	<ul style="list-style-type: none"> Indicates whether the indication is active <ul style="list-style-type: none"> 0x00 - Inactive 0x01 - Active
<i>msgCount</i>	<ul style="list-style-type: none"> Number of messages

8.598.2 Field Documentation

8.598.2.1 uint8_t sms_messageWaitingInfoContent::activeInd

8.598.2.2 uint8_t sms_messageWaitingInfoContent::msgCount

8.598.2.3 uint8_t sms_messageWaitingInfoContent::msgType

8.599 sms_msgProtocolResp Struct Reference

Data Fields

- uint8_t [msgProtocol](#)

8.599.1 Detailed Description

This structure contains get message protocol response parameters

Parameters

<i>msgProtocol</i>	- <ul style="list-style-type: none"> • Message Protocol • Values: <ul style="list-style-type: none"> – 0x00 - MESSAGE_PROTOCOL_CDMA – 0x01 - MESSAGE_PROTOCOL_WCDMA
--------------------	--

8.599.2 Field Documentation

8.599.2.1 uint8_t sms_msgProtocolResp::msgProtocol

8.600 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference

Data Fields

- uint8_t [activated_ind](#)
- uint16_t [num_instances](#)
- [sms_CDMABroadcastConfig](#) [CDMABroadcastConfig](#) [0x05]

8.600.1 Detailed Description

This structure contains the 3GPP2 Broadcast Configuration Information parameters

Parameters

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

8.600.2 Field Documentation

8.600.2.1 `uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind`

8.600.2.2 `sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]`

8.600.2.3 `uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances`

8.601 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference

Data Fields

- `uint8_t activated_ind`
- `uint16_t num_instances`
- `sms_BroadcastConfig broadcastConfig` [0x05]

8.601.1 Detailed Description

This structure contains the 3GPP Broadcast Configuration Information parameters

Parameters

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

8.601.2 Field Documentation

8.601.2.1 `uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind`

8.601.2.2 `sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]`

8.601.2.3 `uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances`

8.602 sms_routeEntry Struct Reference

Data Fields

- `uint8_t messageType`
- `uint8_t messageClass`

- uint8_t [routeStorage](#)
- uint8_t [receiptAction](#)

8.602.1 Detailed Description

This structure contains SMS route entry details

Parameters

<i>messageType</i>	- <ul style="list-style-type: none"> • Message type matching this route • Values: <ul style="list-style-type: none"> – 0x00 - MESSAGE_TYPE_POINT_TO_POINT
<i>messageClass</i>	- <ul style="list-style-type: none"> • Message Class • Values: <ul style="list-style-type: none"> – 0x00 - MESSAGE_CLASS_0 – 0x01 - MESSAGE_CLASS_1 – 0x02 - MESSAGE_CLASS_2 – 0x03 - MESSAGE_CLASS_3 – 0x04 - MESSAGE_CLASS_NONE – 0x05 - MESSAGE_CLASS_CDMA
<i>routeStorage</i>	- <ul style="list-style-type: none"> • If the receiptAction is store where to store the message • Values: <ul style="list-style-type: none"> – 0x00 - STORAGE_TYPE_UIM – 0x01 - STORAGE_TYPE_NV – 0xFF - STORAGE_TYPE_NONE
<i>receiptAction</i>	- <ul style="list-style-type: none"> • Action to be taken on receipt of a message matching the specified type and class for this route • Values: <ul style="list-style-type: none"> – 0x00 - DISCARD (discarded without notification) – 0x01 - STORE AND NOTIFY (stored and notified to the registered clients) – 0x02 - TRANSFER ONLY (transferred to the client, client expected to send the ACK) – 0x03 - TRANSFER AND ACK (transferred to the client, device expected to send the ACK)

8.602.2 Field Documentation

8.602.2.1 uint8_t sms_routeEntry::messageClass

8.602.2.2 uint8_t sms_routeEntry::messageType

8.602.2.3 uint8_t sms_routeEntry::receiptAction

8.602.2.4 uint8_t sms_routeEntry::routeStorage

8.603 sms_sendAsynsmsParams 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](#)

8.603.1 Detailed Description

This structure contains SMS parameters

Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> • Message format • Values: <ul style="list-style-type: none"> – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 6 - GSM/WCDMA PP
<i>messageSize</i>	<ul style="list-style-type: none"> • The length of the message contents in bytes
<i>pMessage</i>	<ul style="list-style-type: none"> • The message contents
<i>pForceOnDC</i>	<ul style="list-style-type: none"> • Force the message to be sent on the CDMA dedicated channel. • Values: <ul style="list-style-type: none"> – 0x00 - Do not care about the channel on which the message is sent – 0x01 - Request to send the message over the dedicated channel
<i>pServiceOption</i>	<ul style="list-style-type: none"> • Service option: • Values: <ul style="list-style-type: none"> – 0x00 - SO_AUTO - AUTO (choose the best service option) – 0x06 - SO_6 - Service option 6 – 0x0E - SO_14 - Service option 14
<i>pFollowOnDC</i>	<ul style="list-style-type: none"> • Flag to request not to disconnect the CDMA dedicated channel after the send operation is complete. • This TLV can be included if more messages are expected to follow. • Values: <ul style="list-style-type: none"> – 0x01 - FOLLOW_ON_DC_ON - On (don't disconnect after send operation) Any value other than 0x01 is treated as an absence of this TLV.

<i>pLinktimer</i>	<ul style="list-style-type: none"> Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow
<i>pSmsOnIms</i>	<ul style="list-style-type: none"> Indicates whether the message is to be sent on IMS. Values: <ul style="list-style-type: none"> 0x00 - Message is not to be sent on IMS 0x01 - Message is to be sent on IMS 0x02 to 0xFF - Reserved
<i>pRetryMessage</i>	<ul style="list-style-type: none"> Indicates this message is a retry message. Values: <ul style="list-style-type: none"> 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.
<i>pRetryMessage-Id</i>	<ul style="list-style-type: none"> Message ID to be used in the retry message. The message ID specified here is used instead of the message ID encoded in the raw message.
<i>pUserData</i>	<ul style="list-style-type: none"> Enables the control point to associate the request with the corresponding indication. The control point might send numerous requests. This TLV will help the control point to identify the request for which the received indication belongs.

8.603.2 Field Documentation

8.603.2.1 `uint32_t sms_sendAsyncsmsParams::messageFormat`

8.603.2.2 `uint32_t sms_sendAsyncsmsParams::messageSize`

8.603.2.3 `uint8_t* sms_sendAsyncsmsParams::pFollowOnDC`

8.603.2.4 `uint8_t* sms_sendAsyncsmsParams::pForceOnDC`

8.603.2.5 `uint8_t* sms_sendAsyncsmsParams::pLinktimer`

8.603.2.6 `uint8_t* sms_sendAsyncsmsParams::pMessage`

8.603.2.7 `uint8_t* sms_sendAsyncsmsParams::pRetryMessage`

8.603.2.8 `uint32_t* sms_sendAsyncsmsParams::pRetryMessageId`

8.603.2.9 `uint8_t* sms_sendAsyncsmsParams::pServiceOption`

8.603.2.10 `uint8_t* sms_sendAsyncsmsParams::pSmsOnIms`

8.603.2.11 `uint32_t* sms_sendAsyncsmsParams::pUserData`

8.604 sms_setIndicationReg Struct Reference

Data Fields

- uint8_t * [pRegTransLayerInfoEvt](#)
- uint8_t * [pRegTransNWRegInfoEvt](#)
- uint8_t * [pRegCallStatInfoEvt](#)

8.604.1 Detailed Description

This structure contains Indication Register request parameters

Parameters

<i>pRegTransLayerInfoEvt</i>	- <ul style="list-style-type: none"> • Optional parameter indicating registration status of transport layer information events • Values: <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled – 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
<i>pRegTransNWRegInfoEvt</i>	- <ul style="list-style-type: none"> • Optional parameter indicating registration status of transport network registration information events • Values: <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled – 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
<i>pRegCallStatInfoEvt</i>	- <ul style="list-style-type: none"> • Optional parameter indicating registration status of call status information events • Values: <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled – 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

8.604.2 Field Documentation

8.604.2.1 uint8_t* sms_setIndicationReg::pRegCallStatInfoEvt

8.604.2.2 uint8_t* sms_setIndicationReg::pRegTransLayerInfoEvt

8.604.2.3 uint8_t* sms_setIndicationReg::pRegTransNWRegInfoEvt

8.605 sms_setRoutesReq Struct Reference

Data Fields

- uint16_t [numOfRoutes](#)

- [sms_routeEntry](#) routeList [0x0A]
- uint8_t * [pTransferStatusReport](#)

8.605.1 Detailed Description

This structure contains SMS route request parameters

Parameters

<i>numOfRoutes</i>	- <ul style="list-style-type: none"> • Number of sets of the following element
<i>routeList</i>	- <ul style="list-style-type: none"> • Array containing the set of sms_routeEntry
<i>pTransferStatus-Report</i>	- <ul style="list-style-type: none"> • 0x01 - Status report are transferred to the client (optional)

8.605.2 Field Documentation

8.605.2.1 uint16_t sms_setRoutesReq::numOfRoutes

8.605.2.2 uint8_t* sms_setRoutesReq::pTransferStatusReport

8.605.2.3 sms_routeEntry sms_setRoutesReq::routeList[0x0A]

8.606 sms_transLayerInfo Struct Reference

Data Fields

- uint8_t [TransType](#)
- uint8_t [TransCap](#)

8.606.1 Detailed Description

This structure contains Transport Layer Information

Parameters

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

8.606.2 Field Documentation

8.606.2.1 `uint8_t sms_transLayerInfo::TransCap`

8.606.2.2 `uint8_t sms_transLayerInfo::TransType`

8.607 sMSCAddressInfo Struct Reference

Data Fields

- `uint8_t length`
- `uint8_t data` [256]

8.607.1 Detailed Description

This structure contains SMSC information

Parameters

<i>length</i>	<ul style="list-style-type: none">• Number of sets of following element
<i>data</i>	<ul style="list-style-type: none">• SMSC address

8.607.2 Field Documentation

8.607.2.1 `uint8_t sMSCAddressInfo::data`[256]

8.607.2.2 `uint8_t sMSCAddressInfo::length`

8.608 sMSCAddressTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSCAddressInfo SMSCInfo`

8.608.1 Detailed Description

This structure contains SMSC TLV information

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• Boolean indicating the presence of the TLV in the QMI response
<i>SMSCInfo</i>	<ul style="list-style-type: none">• SMSC Address• See sMSCAddressInfo for more information

8.608.2 Field Documentation

8.608.2.1 **sMSCAddressInfo** sMSCAddressTlv::SMSCInfo

8.608.2.2 **uint8_t** sMSCAddressTlv::TlvPresent

8.609 sMSEtwsMessageInfo Struct Reference

Data Fields

- **uint8_t** [notificationType](#)
- **uint16_t** [length](#)
- **uint8_t** [data](#) [1254]

8.609.1 Detailed Description

This structure contains ETWS information

Parameters

<i>notificationType</i>	<ul style="list-style-type: none"> • Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS
<i>length</i>	<ul style="list-style-type: none"> • Number of sets of following elements
<i>data</i>	<ul style="list-style-type: none"> • Raw message data

8.609.2 Field Documentation

8.609.2.1 **uint8_t** sMSEtwsMessageInfo::data[1254]

8.609.2.2 **uint16_t** sMSEtwsMessageInfo::length

8.609.2.3 **uint8_t** sMSEtwsMessageInfo::notificationType

8.610 sMSEtwsMessageTlv Struct Reference

Data Fields

- **uint8_t** [TlvPresent](#)
- [sMSEtwsMessageInfo](#) [EtwsMessageInfo](#)

8.610.1 Detailed Description

This structure contains ETWS TLV information

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Boolean indicating the presence of the TLV in the QMI response
-------------------	--

<i>EtwsMessage-Info</i>	<ul style="list-style-type: none">• ETWS Message• See sMSEtwsMessageInfo for more information
-------------------------	--

8.610.2 Field Documentation

8.610.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

8.610.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

8.611 sMSEtwsPlmnInfo Struct Reference

Data Fields

- uint16_t [mobileCountryCode](#)
- uint16_t [mobileNetworkCode](#)

8.611.1 Detailed Description

This structure contains ETWS PLMN information

Parameters

<i>mobileCountry-Code</i>	<ul style="list-style-type: none">• 16 bit representation of MCC value range : 0 -999
<i>mobileNetwork-Code</i>	<ul style="list-style-type: none">• 16 bit representation of MNC value range : 0 -999

8.611.2 Field Documentation

8.611.2.1 uint16_t sMSEtwsPlmnInfo::mobileCountryCode

8.611.2.2 uint16_t sMSEtwsPlmnInfo::mobileNetworkCode

8.612 sMSMessageModelInfo Struct Reference

Data Fields

- uint8_t [messageMode](#)

8.612.1 Detailed Description

This structure contains message mode information.

Parameters

<i>messageMode</i>	<ul style="list-style-type: none"> • Message Mode <ul style="list-style-type: none"> – 0x00 - CDMA – 0x01 - GW
--------------------	--

8.612.2 Field Documentation

8.612.2.1 uint8_t sMSMessageModelInfo::messageMode

8.613 sMSMTMessageInfo Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t [messageIndex](#)

8.613.1 Detailed Description

This structure contains MT message information.

Parameters

<i>storageType</i>	memory storage 0x00-UIIM 0x01-NV
<i>messageIndex</i>	MT Message index

8.613.2 Field Documentation

8.613.2.1 uint32_t sMSMTMessageInfo::messageIndex

8.613.2.2 uint32_t sMSMTMessageInfo::storageType

8.614 sMSOnIMSInfo Struct Reference

Data Fields

- uint8_t [smsOnIMS](#)

8.614.1 Detailed Description

This structure contains SMS on IMS information

Parameters

<i>smsOnIMS</i>	SMS on IMS
-----------------	------------

8.614.2 Field Documentation

8.614.2.1 uint8_t sMSOnIMSInfo::smsOnIMS

8.615 sMSOnIMSTlv Struct Reference

Data Fields

- [uint8_t TlvPresent](#)
- [sMSOnIMSInfo IMSInfo](#)

8.615.1 Detailed Description

This structure contains SMS on IMS TLV information.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• Boolean indicating the presence of the TLV in the QMI response
<i>IMSInfo</i>	<ul style="list-style-type: none">• SMS on IMS• See sMSOnIMSInfo for more information

8.615.2 Field Documentation

8.615.2.1 [sMSOnIMSInfo](#) [sMSOnIMSTlv::IMSInfo](#)

8.615.2.2 [uint8_t](#) [sMSOnIMSTlv::TlvPresent](#)

8.616 sMSTransferRouteMTMessageInfo Struct Reference

Data Fields

- [uint8_t ackIndicator](#)
- [uint32_t transactionID](#)
- [uint8_t format](#)
- [uint16_t length](#)
- [uint8_t data](#) [256]

8.616.1 Detailed Description

This structure contains Transfer route MT message information.

Parameters

<i>ackIndicator</i>	<ul style="list-style-type: none">• Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK
<i>transactionID</i>	<ul style="list-style-type: none">• Transaction ID of the message
<i>format</i>	<ul style="list-style-type: none">• Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC

<i>length</i>	<ul style="list-style-type: none"> Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes
<i>data</i>	<ul style="list-style-type: none"> Raw message data

8.616.2 Field Documentation

8.616.2.1 `uint8_t sMSTransferRouteMTMessageInfo::ackIndicator`

8.616.2.2 `uint8_t sMSTransferRouteMTMessageInfo::data[256]`

8.616.2.3 `uint8_t sMSTransferRouteMTMessageInfo::format`

8.616.2.4 `uint16_t sMSTransferRouteMTMessageInfo::length`

8.616.2.5 `uint32_t sMSTransferRouteMTMessageInfo::transactionID`

8.617 `swi_uint256_t` Struct Reference

Data Fields

- `uint16_t word` [16]

8.617.1 Detailed Description

SWI 256 bit data type

8.617.2 Field Documentation

8.617.2.1 `uint16_t swi_uint256_t::word[16]`

8.618 `swiaudio_PCMparams` Struct Reference

Data Fields

- `uint8_t iFaceTabLen`
- `uint8_t iFaceTab` [255]

8.618.1 Detailed Description

This structure contains the PCM parameters.

Parameters

<i>iFaceTabLen</i>	<ul style="list-style-type: none"> Number of sets of iface table
--------------------	---

<i>iFaceTab</i>	<ul style="list-style-type: none"> Physical Interface Parameters See qaGobiApiTableSwiAudio.h for more information on physical interface parameters
-----------------	---

8.618.2 Field Documentation

8.618.2.1 `uint8_t swiaudio_PCMparams::iFaceTab[255]`

8.618.2.2 `uint8_t swiaudio_PCMparams::iFaceTabLen`

8.619 swidms_ehrpdMTUSizeTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint16_t ehrpdMTUSize`

8.619.1 Detailed Description

This structure contains the parameters for EHRPD MTU Size.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present.
<i>ehrpdmTUSize</i>	EHRPD size <ul style="list-style-type: none"> Size of EHRPD MTU

8.619.2 Field Documentation

8.619.2.1 `uint16_t swidms_ehrpdMTUSizeTlv::ehrpdmTUSize`

8.619.2.2 `uint8_t swidms_ehrpdMTUSizeTlv::TlvPresent`

8.620 swidms_hrpdmTUSizeTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint16_t hrpdMTUSize`

8.620.1 Detailed Description

This structure contains the parameters for HRPD MTU Size.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.
<i>hrpdMTUSize</i>	HRPD size <ul style="list-style-type: none"> • Size of HRPD MTU

8.620.2 Field Documentation

8.620.2.1 uint16_t swidms_hrpdmTUSizeTlv::hrpdMTUSize

8.620.2.2 uint8_t swidms_hrpdmTUSizeTlv::TlvPresent

8.621 swidms_ifaceCfgTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [CurrentCfgType](#)
- uint32_t [CfgValue](#)

8.621.1 Detailed Description

This structure contains the parameters for host usb interface composition

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.
<i>CurrentCfgType</i>	<ul style="list-style-type: none"> • Current configure type, see values below: <ul style="list-style-type: none"> – 1 - MBIM – 2 - USBIF + PCIE – 3 - RMNET – 4 - RNDIS – 5 - PCIE only

<i>CfgValue</i>	<ul style="list-style-type: none"> • Host composition bit mask, see supported USB interface bitmasks, not supported by QMI object setting for 9x50 modules <ul style="list-style-type: none"> – 0x00000001 - DIAG interface – 0x00000002 - ADB interface – 0x00000004 - NMEA interface – 0x00000008 - MODEM interface – 0x00000010 - RESERVED5 – 0x00000020 - RESERVED6 – 0x00000040 - RESERVED7 – 0x00000080 - RESERVED8 – 0x00000100 - RMENT0 interface – 0x00000200 - RESERVED10 – 0x00000400 - RMENT1 interface – 0x00000800 - RESERVED12 – 0x00001000 - MBIM interface – 0x00002000 - RESERVED14 – 0x00004000 - RNDIS interface – 0x00008000 - RESERVED16 – 0x00010000 - AUDIO interface – 0x00020000 - RESERVED18 – 0x00080000 - ECM interface – 0x00100000 - RESERVED21 – 0x00200000 - RESERVED22 – 0xFFC00000 - RESERVED
-----------------	--

8.621.2 Field Documentation

8.621.2.1 `uint32_t swidms_ifaceCfgTlv::CfgValue`

8.621.2.2 `uint32_t swidms_ifaceCfgTlv::CurrentCfgType`

8.621.2.3 `uint8_t swidms_ifaceCfgTlv::TlvPresent`

8.622 swidms_mtuSize3gppTlv Struct Reference

Data Fields

- `uint8_t` [TlvPresent](#)
- `uint16_t` [MTUSize3gpp](#)

8.622.1 Detailed Description

This structure contains the parameters for 3GPP MTU Size.

Parameters

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

<i>MTUSize3gpp</i>	3GPP MTU size <ul style="list-style-type: none"> • Size of 3GPP MTU
--------------------	--

8.622.2 Field Documentation

8.622.2.1 `uint16_t swidms_mtuSize3gppTlv::MTUSize3gpp`

8.622.2.2 `uint8_t swidms_mtuSize3gppTlv::TlvPresent`

8.623 swidms_supportedIntBitmaskTlv Struct Reference

Data Fields

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

8.623.1 Detailed Description

This structure contains the parameters for supported usb bitmasks

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.
<i>ValidBitmasks</i>	<ul style="list-style-type: none"> • for current configuration type, this is the supported bitmasks that the host can change.

8.623.2 Field Documentation

8.623.2.1 `uint8_t swidms_supportedIntBitmaskTlv::TlvPresent`

8.623.2.2 `uint32_t swidms_supportedIntBitmaskTlv::ValidBitmasks`

8.624 swidms_SwiDmsGetHWWatchdog Struct Reference

Data Fields

- `uint32_t` [timeout](#)
- `uint32_t` [resetDelay](#)
- `uint8_t` [enable](#)
- `uint32_t` [count](#)

8.624.1 Detailed Description

This structure contains the parameters for hardware watchdog settings

Parameters

<i>timeout</i>	<ul style="list-style-type: none"> • timeout value for HW watchdog (unit in second)
<i>resetDelay</i>	<ul style="list-style-type: none"> • delay before reset after watchdog timeout (unit in second)
<i>enable</i>	<ul style="list-style-type: none"> • 0 to disable watchdog; 1 to enable watchdog
<i>count</i>	<ul style="list-style-type: none"> • once <timeout> has occurred, the <count> will be increased by 1 and the timer will be restarted automatically. This <count> indicates the number of renewals.

8.624.2 Field Documentation

8.624.2.1 uint32_t swidms_SwiDmsGetHWWatchdog::count

8.624.2.2 uint8_t swidms_SwiDmsGetHWWatchdog::enable

8.624.2.3 uint32_t swidms_SwiDmsGetHWWatchdog::resetDelay

8.624.2.4 uint32_t swidms_SwiDmsGetHWWatchdog::timeout

8.625 swidms_usbMTUSizeTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint16_t [UsbMTUSize](#)

8.625.1 Detailed Description

This structure contains the parameters for USB MTU Size

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.
<i>UsbMTUSize</i>	USB MTU size <ul style="list-style-type: none"> • Size of USB MTU

8.625.2 Field Documentation

8.625.2.1 uint8_t swidms_usbMTUSizeTlv::TlvPresent

8.625.2.2 uint16_t swidms_usbMTUSizeTlv::UsbMTUSize

8.626 tdscdmaSigInfoExt Struct Reference

Data Fields

- float [rssi](#)
- float [rscp](#)
- float [ecio](#)
- float [sinr](#)

8.626.1 Detailed Description

This structure contains the TDSCDMA Signal Strength Info Extended

Parameters

<i>rssi</i>	<ul style="list-style-type: none"> • Measured RSSI in dB
<i>rscp[Optional]</i>	<ul style="list-style-type: none"> • Measured RSCP in dBm
<i>ecio[Optional]</i>	<ul style="list-style-type: none"> • Measured ECIO in dBm.
<i>sinr[Optional]</i>	<ul style="list-style-type: none"> • Measured SINR in dB. -15 dB is sent to clients if the actual SINR is less than -15 dB

8.626.2 Field Documentation

8.626.2.1 float tdscdmaSigInfoExt::ecio

8.626.2.2 float tdscdmaSigInfoExt::rscp

8.626.2.3 float tdscdmaSigInfoExt::rssi

8.626.2.4 float tdscdmaSigInfoExt::sinr

8.627 tempData_t Struct Reference

Data Fields

- uint32_t [timeSource](#)
- uint32_t [timeOfFirstSample](#)
- uint8_t [temperatureDataLen](#)
- uint16_t [timeOffset](#) [64]
- uint32_t [temperature](#) [64]

8.627.1 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"> • Time source of the sensor data • Valid values <ul style="list-style-type: none"> – 0 - Sensor time source is unspecified – 1 - Time source is common between the sensors and the location engine
<i>timeOfFirst-Sample</i>	<ul style="list-style-type: none"> • Denotes a full 32-bit time stamp of the first (oldest) sample in this message. • The time stamp is in the time reference scale that is used by the sensor time source. • Units - Milliseconds
<i>temperature-DataLen</i>	<ul style="list-style-type: none"> • Number of sets of the following elements <ul style="list-style-type: none"> – timeOffset – temperature
<i>timeOffset</i>	<ul style="list-style-type: none"> • Sample time offset • Units - Milliseconds
<i>temperature</i>	<ul style="list-style-type: none"> • Sensor temperature. • Type - Floating point • Units - Degrees Celsius • Range -50 to +100.00

8.627.2 Field Documentation

8.627.2.1 uint32_t tempData_t::temperature[64]

8.627.2.2 uint8_t tempData_t::temperatureDataLen

8.627.2.3 uint32_t tempData_t::timeOfFirstSample

8.627.2.4 uint16_t tempData_t::timeOffset[64]

8.627.2.5 uint32_t tempData_t::timeSource

8.628 tmd_mitigationDevList Struct Reference

Data Fields

- uint8_t [mitigationDevIdLen](#)
- char [mitigationDevId](#) [255]
- uint8_t [maxMitigationLevel](#)

8.628.1 Detailed Description

This structure contains mitigation devices list

Parameters

<i>mitigationDevIdLen</i>	<ul style="list-style-type: none"> Number of sets of the following elements <ul style="list-style-type: none"> mitigation_dev_id
<i>mitigationDevId</i>	<ul style="list-style-type: none"> Mitigation device ID.
<i>maxMitigationLevel</i>	<ul style="list-style-type: none"> Maximum valid mitigation level. Valid range - 0 to max_mitigation_level.

8.628.2 Field Documentation

8.628.2.1 `uint8_t tmd_mitigationDevList::maxMitigationLevel`8.628.2.2 `char tmd_mitigationDevList::mitigationDevId[255]`8.628.2.3 `uint8_t tmd_mitigationDevList::mitigationDevIdLen`

8.629 transferRouteMessageTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo`

8.629.1 Detailed Description

This structure contains Transfer route MT message TLV information.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> Boolean indicating the presence of the TLV in the QMI response
<i>TransferRouteMTMessageInfo</i>	<ul style="list-style-type: none"> Transfer Route MT Message See sMSTransferRouteMTMessageInfo for more information

8.629.2 Field Documentation

8.629.2.1 `uint8_t transferRouteMessageTlv::TlvPresent`8.629.2.2 `sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo`

8.630 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](#)

8.630.1 Detailed Description

This structure contains Application Status Information loaded on the card.

Parameters

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

<i>persoFeature</i>	<ul style="list-style-type: none"> • Indicates the personalization feature. • This applies only when a personalization code is required to deactivate or unblock personalization. <ul style="list-style-type: none"> – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization – 11 - Unknown
<i>persoRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to disable the personalization.
<i>persoUnblock-Retries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock the personalization.
<i>aidLength</i>	<ul style="list-style-type: none"> • Number of sets of the following elements. i.e. aidVal • If zero(0) then no aidVal information exists.
<i>aidVal[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • Application identifier value.
<i>univPin</i>	<ul style="list-style-type: none"> • Indicates whether UPIN replaces PIN1. <ul style="list-style-type: none"> – 0 - PIN1 is used – 1 - UPIN replaces PIN1
<i>pin1State</i>	<ul style="list-style-type: none"> • Indicates the state of PIN1. <ul style="list-style-type: none"> – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked
<i>pin1Retries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to verify PIN1.
<i>puk1Retries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock PIN1.

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

8.630.2 Field Documentation

8.630.2.1 uint8_t uim_appStatus::aidLength

8.630.2.2 uint8_t uim_appStatus::aidVal[255]

8.630.2.3 uint8_t uim_appStatus::appState

8.630.2.4 uint8_t uim_appStatus::appType

8.630.2.5 uint8_t uim_appStatus::persoFeature

8.630.2.6 uint8_t uim_appStatus::persoRetries

8.630.2.7 uint8_t uim_appStatus::persoState

8.630.2.8 uint8_t uim_appStatus::persoUnblockRetries

8.630.2.9 uint8_t uim_appStatus::pin1Retries

8.630.2.10 uint8_t uim_appStatus::pin1State

8.630.2.11 uint8_t uim_appStatus::pin2Retries

8.630.2.12 uint8_t uim_appStatus::pin2State

8.630.2.13 uint8_t uim_appStatus::puk1Retries

8.630.2.14 uint8_t uim_appStatus::puk2Retries

8.630.2.15 uint8_t uim_appStatus::univPin

8.631 uim_authenticateResult Struct Reference

Data Fields

- uint16_t [contentLen](#)

- uint8_t [content](#) [1024]

8.631.1 Detailed Description

This structure contains the information about the authenticate result.

Parameters

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

8.631.2 Field Documentation

8.631.2.1 uint8_t uim_authenticateResult::content[1024]

8.631.2.2 uint16_t uim_authenticateResult::contentLen

8.632 uim_authenticationData Struct Reference

Data Fields

- uint8_t [context](#)
- uint16_t [dataLen](#)
- uint8_t [data](#) [1024]

8.632.1 Detailed Description

This structure contains the Session Information.

Parameters

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

8.632.2 Field Documentation

8.632.2.1 uint8_t uim_authenticationData::context

8.632.2.2 uint8_t uim_authenticationData::data[1024]

8.632.2.3 uint16_t uim_authenticationData::dataLen

8.633 uim_cardResult Struct Reference

Data Fields

- uint8_t [sw1](#)
- uint8_t [sw2](#)

8.633.1 Detailed Description

This structure contains the information about the card result.

Parameters

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

8.633.2 Field Documentation

8.633.2.1 `uint8_t uim_cardResult::sw1`8.633.2.2 `uint8_t uim_cardResult::sw2`8.634 `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]

8.634.1 Detailed Description

This structure contains Card Status Information.

Parameters

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

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

8.634.2 Field Documentation

8.634.2.1 `uint16_t uim_cardStatus::index1xPri`

8.634.2.2 `uint16_t uim_cardStatus::index1xSec`

8.634.2.3 `uint16_t uim_cardStatus::indexGwPri`

8.634.2.4 `uint16_t uim_cardStatus::indexGwSec`

8.634.2.5 `uint8_t uim_cardStatus::numSlot`

8.634.2.6 `uim_slotInfo uim_cardStatus::SlotInfo[5]`

8.635 uim_changeUIMPIN Struct Reference

Data Fields

- `uint8_t pinID`
- `uint8_t oldPINLen`
- `uint8_t oldPINVal [255]`
- `uint8_t pinLen`
- `uint8_t pinVal [255]`

8.635.1 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"> • Indicates the PIN ID to be changed. <ul style="list-style-type: none"> – 1 - PIN1 (also called PIN) – 2 - PIN2 – 3 - Universal PIN – 4 - Hidden key
--------------	--

<i>oldPINLen</i>	<ul style="list-style-type: none"> Length of the following elements i.e. old pin value.
<i>oldPINVal</i> [MAX- _DESCRIPTIO- N_LENGTH]	<ul style="list-style-type: none"> Old PIN value. This value is a sequence of ASCII characters.
<i>pinLen</i>	<ul style="list-style-type: none"> Length of the following elements i.e. new pin value.
<i>pinVal</i> [MAX_DE- SCRIPTION_LE- NGTH]	<ul style="list-style-type: none"> New PIN value. This value is a sequence of ASCII characters.

8.635.2 Field Documentation

8.635.2.1 `uint8_t uim_changeUIMPIN::oldPINLen`

8.635.2.2 `uint8_t uim_changeUIMPIN::oldPINVal[255]`

8.635.2.3 `uint8_t uim_changeUIMPIN::pinID`

8.635.2.4 `uint8_t uim_changeUIMPIN::pinLen`

8.635.2.5 `uint8_t uim_changeUIMPIN::pinVal[255]`

8.636 uim_depersonalizationInformation Struct Reference

Data Fields

- `uint8_t feature`
- `uint8_t operation`
- `uint8_t ckLen`
- `uint8_t ckVal [255]`

8.636.1 Detailed Description

This structure contains the Depersonalization Information.

Parameters

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

8.636.2 Field Documentation

8.636.2.1 uint8_t uim_depersonalizationInformation::ckLen

8.636.2.2 uint8_t uim_depersonalizationInformation::ckVal[255]

8.636.2.3 uint8_t uim_depersonalizationInformation::feature

8.636.2.4 uint8_t uim_depersonalizationInformation::operation

8.637 uim_encryptedPIN1 Struct Reference

Data Fields

- uint8_t [pin1Len](#)
- uint8_t [pin1Val](#) [255]

8.637.1 Detailed Description

This structure contains the encrypted PIN1 Information.

Parameters

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

Note

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

8.637.2 Field Documentation

8.637.2.1 `uint8_t uim_encryptedPIN1::pin1Len`

8.637.2.2 `uint8_t uim_encryptedPIN1::pin1Val[255]`

8.638 `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]`

8.638.1 Detailed Description

This structure contains the information about the File Attributes.

Parameters

<i>fileSize</i>	<ul style="list-style-type: none"> • Indicates the size of the file.
<i>fileID</i>	<ul style="list-style-type: none"> • Indicates the ID of the file.

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

<i>secIncrease</i>	<ul style="list-style-type: none"> • Increase security attributes. <ul style="list-style-type: none"> – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
<i>secIncrease-Mask</i>	<ul style="list-style-type: none"> • Mask with increase security attributes. • This field is valid only when required by security attributes. <ul style="list-style-type: none"> – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
<i>secDeactivate</i>	<ul style="list-style-type: none"> • Deactivate security attributes. <ul style="list-style-type: none"> – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
<i>secDeactivate-Mask</i>	<ul style="list-style-type: none"> • Mask with deactivate security attributes. • This field is valid only when required by security attributes. <ul style="list-style-type: none"> – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
<i>secActivate</i>	<ul style="list-style-type: none"> • Activate security attributes. <ul style="list-style-type: none"> – 0 - Always – 1 - Never – 2 - AND condition – 3 - OR condition – 4 - Single condition
<i>secActivateMask</i>	<ul style="list-style-type: none"> • Mask with activate security attributes. • This field is valid only when required by security attributes. <ul style="list-style-type: none"> – Bit 0 - PIN1 – Bit 1 - PIN2 – Bit 2 - UPIN – Bit 3 - ADM
<i>rawLen</i>	<ul style="list-style-type: none"> • Length of the following elements i.e. raw value.

<i>rawValue</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> Raw value of file attributes.
--	---

8.638.2 Field Documentation

- 8.638.2.1 `uint16_t uim_fileAttributes::fileID`
- 8.638.2.2 `uint16_t uim_fileAttributes::fileSize`
- 8.638.2.3 `uint8_t uim_fileAttributes::fileType`
- 8.638.2.4 `uint16_t uim_fileAttributes::rawLen`
- 8.638.2.5 `uint8_t uim_fileAttributes::rawValue[255]`
- 8.638.2.6 `uint16_t uim_fileAttributes::recordCount`
- 8.638.2.7 `uint16_t uim_fileAttributes::recordSize`
- 8.638.2.8 `uint8_t uim_fileAttributes::secActivate`
- 8.638.2.9 `uint16_t uim_fileAttributes::secActivateMask`
- 8.638.2.10 `uint8_t uim_fileAttributes::secDeactivate`
- 8.638.2.11 `uint16_t uim_fileAttributes::secDeactivateMask`
- 8.638.2.12 `uint8_t uim_fileAttributes::secIncrease`
- 8.638.2.13 `uint16_t uim_fileAttributes::secIncreaseMask`
- 8.638.2.14 `uint8_t uim_fileAttributes::secRead`
- 8.638.2.15 `uint16_t uim_fileAttributes::secReadMask`
- 8.638.2.16 `uint8_t uim_fileAttributes::secWrite`
- 8.638.2.17 `uint16_t uim_fileAttributes::secWriteMask`

8.639 uim_fileInfo Struct Reference

Data Fields

- `uint16_t fileID`
- `uint8_t pathLen`
- `uint16_t path [255]`

8.639.1 Detailed Description

This structure contains paramaters for file Information

Parameters

<i>fileID</i>	<ul style="list-style-type: none"> This is Identifier to SIM files; e.g. in UIM "6F07" is Identifier of IMSI File
<i>pathLen</i>	<ul style="list-style-type: none"> Length of file Path
<i>path</i>	<ul style="list-style-type: none"> 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.

8.639.2 Field Documentation

8.639.2.1 uint16_t uim_fileInfo::fileID

8.639.2.2 uint16_t uim_fileInfo::path[255]

8.639.2.3 uint8_t uim_fileInfo::pathLen

8.640 uim_GetSlotsInfoTlv Struct Reference

Data Fields

- uint8_t [NumberOfPhySlotInfo](#)
- [uim_physlotInfo](#) [uimSlotInfo](#) [255]
- uint8_t [TlvPresent](#)

8.640.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Information TLV.

Parameters

<i>NumberOfPhySlotInfo</i>	<ul style="list-style-type: none"> Number of sets of the Slot Information.
<i>uimSlotInfo</i>	<ul style="list-style-type: none"> Slots information See UIMSlotInfo for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> Physical slot information TLV present.

8.640.2 Field Documentation

8.640.2.1 uint8_t uim_GetSlotsInfoTlv::NumberOfPhySlotInfo

8.640.2.2 uint8_t uim_GetSlotsInfoTlv::TlvPresent

8.640.2.3 uim_physlotInfo uim_GetSlotsInfoTlv::uimSlotInfo[255]

8.641 uim_GetSlotsStatusTlv Struct Reference

Data Fields

- uint8_t [NumberOfPhySlot](#)
- [slot_t uimSlotStatus](#) [255]
- uint8_t [TlvPresent](#)

8.641.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status TLV.

Parameters

<i>NumberOfPhySlot</i>	<ul style="list-style-type: none">• Number of sets of the Physical Slot Status.
<i>uimSlotStatus</i>	<ul style="list-style-type: none">• Slots Status See UIMSlotStatus for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none">• Physical slot status TLV present.

8.641.2 Field Documentation

8.641.2.1 uint8_t uim_GetSlotsStatusTlv::NumberOfPhySlot

8.641.2.2 uint8_t uim_GetSlotsStatusTlv::TlvPresent

8.641.2.3 slot_t uim_GetSlotsStatusTlv::uimSlotStatus[255]

8.642 uim_hotSwapStatus Struct Reference

Data Fields

- uint8_t [hotSwapLength](#)
- uint8_t [hotSwap](#) [255]

8.642.1 Detailed Description

This structure contains Hot Swap Status Information.

Parameters

<i>hotSwapLength</i>	<ul style="list-style-type: none"> • Number of sets of the following elements. i.e. hot_swap
<i>hotSwap</i>	<ul style="list-style-type: none"> • Indicates the status of the hot-swap switch. <ul style="list-style-type: none"> – 0 - Hot-swap is not supported – 1 - Hot-swap is supported, but the status of the switch is not supported – 2 - Switch indicates that the card is present – 3 - Switch indicates that the card is not present

8.642.2 Field Documentation

8.642.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.642.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.643 uim_personalizationStatus Struct Reference

Data Fields

- uint8_t [numFeatures](#)
- uint8_t [feature](#) [12]
- uint8_t [verifyLeft](#) [12]
- uint8_t [unblockLeft](#) [12]

8.643.1 Detailed Description

This structure contains the information about the card result.

Parameters

<i>numFeatures</i>	<ul style="list-style-type: none"> • Number of active personalization features. The following block is repeated for each feature.
<i>feature</i>	<ul style="list-style-type: none"> • Indicates the personalization feature to deactivate or unblock. Valid values: <ul style="list-style-type: none"> – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization
<i>verifyLeft</i>	<ul style="list-style-type: none"> • Number of the remaining attempts to verify the personalization feature.

<i>unblockLeft</i>	<ul style="list-style-type: none"> Number of the remaining attempts to unblock the personalization feature.
--------------------	--

8.643.2 Field Documentation

8.643.2.1 `uint8_t uim_personalizationStatus::feature[12]`

8.643.2.2 `uint8_t uim_personalizationStatus::numFeatures`

8.643.2.3 `uint8_t uim_personalizationStatus::unblockLeft[12]`

8.643.2.4 `uint8_t uim_personalizationStatus::verifyLeft[12]`

8.644 uim_physlotInfo Struct Reference

Data Fields

- `uint32_t cardProtocol`
- `uint8_t numApp`
- `uint8_t atrValueLen`
- `uint8_t atrValue [255]`
- `uint8_t iseUICC`

8.644.1 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"> Protocol of the card. Valid values: <ul style="list-style-type: none"> 0x00 - Unknown 0x01 - ICC Protocol 0x02 - UICC Protocol
<i>numApp</i>	<ul style="list-style-type: none"> Number of valid applications present in the EF-DIR of card.
<i>atrValueLen</i>	<ul style="list-style-type: none"> Number of sets of the atrValue
<i>atrValue</i>	<ul style="list-style-type: none"> Raw value of the ATR sent by the card during the initialization.
<i>iseUICC</i>	<ul style="list-style-type: none"> Indicates whether the card is an eUICC card based on the ATR. Values <ul style="list-style-type: none"> 0 - Not an eUICC card 1 - eUICC card

8.644.2 Field Documentation

8.644.2.1 `uint8_t uim_physlotInfo::atrValue[255]`

8.644.2.2 `uint8_t uim_physlotInfo::atrValueLen`

8.644.2.3 `uint32_t uim_physlotInfo::cardProtocol`

8.644.2.4 `uint8_t uim_physlotInfo::iseUICC`

8.644.2.5 `uint8_t uim_physlotInfo::numApp`

8.645 `uim_physlotsInfo` Struct Reference

Data Fields

- [uim_physlotInfo uimSlotInfo](#) [255]

8.645.1 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"> • Contain all slots Information.
---	--

8.645.2 Field Documentation

8.645.2.1 `uim_physlotInfo uim_physlotsInfo::uimSlotInfo[255]`

8.646 `uim_readResult` Struct Reference

Data Fields

- `uint16_t contentLen`
- `uint8_t content` [255]

8.646.1 Detailed Description

This structure contains the information for write operation.

Parameters

<i>contentLen</i>	<ul style="list-style-type: none"> • Number of sets of content.
<i>content</i> [255]	<ul style="list-style-type: none"> • Read content. • The content is the sequence of bytes as read from the card.

8.646.2 Field Documentation

8.646.2.1 `uint8_t uim_readResult::content[255]`

8.646.2.2 `uint16_t uim_readResult::contentLen`

8.647 uim_readTransparentInfo Struct Reference

Data Fields

- `uint16_t offset`
- `uint16_t length`

8.647.1 Detailed Description

This structure contains the information for read operation.

Parameters

<i>offset</i>	<ul style="list-style-type: none">• Offset for the read operation.
<i>length</i>	<ul style="list-style-type: none">• Length of the content to be read.• The value 0 is used to read the complete file.

8.647.2 Field Documentation

8.647.2.1 `uint16_t uim_readTransparentInfo::length`

8.647.2.2 `uint16_t uim_readTransparentInfo::offset`

8.648 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]`

8.648.1 Detailed Description

This structure contains information of parameters associated with the Refresh Event.

Parameters

<i>stage</i>	<ul style="list-style-type: none"> Indicates the stage of the Refresh procedure. <ul style="list-style-type: none"> 0 - Waiting for OK to refresh 1 - Refresh started 2 - Refresh ended successfully 3 - Refresh failed
<i>mode</i>	<ul style="list-style-type: none"> Indicates the Refresh mode. <ul style="list-style-type: none"> 0 - Reset 1 - Init 2 - Init and FCN 3 - FCN 4 - Init and Full FCN 5 - Application reset 6 - 3G session reset
<i>sessionType</i>	<ul style="list-style-type: none"> Indicates the session type. <ul style="list-style-type: none"> 0 - Primary GW provisioning 1 - Primary 1X provisioning 2 - Secondary GW provisioning 3 - Secondary 1X provisioning 4 - Nonprovisioning on slot 1 5 - Nonprovisioning on slot 2 6 - Card on slot 1 7 - Card on slot 2 8 - Logical channel on slot 1 9 - Logical channel on slot 2
<i>aidLength</i>	<ul style="list-style-type: none"> Number of sets of the following elements <ul style="list-style-type: none"> Application Identifier
<i>aid</i>	<ul style="list-style-type: none"> 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
<i>numFiles</i>	<ul style="list-style-type: none"> Number of sets of the following elements: <ul style="list-style-type: none"> file_id path_len path
<i>arrfileInfo</i>	<ul style="list-style-type: none"> Array of file Information struct

8.648.2 Field Documentation

- 8.648.2.1 `uint8_t uim_refreshevent::aid[255]`
- 8.648.2.2 `uint8_t uim_refreshevent::aidLength`
- 8.648.2.3 `uim_fileInfo uim_refreshevent::arrfileInfo[255]`
- 8.648.2.4 `uint8_t uim_refreshevent::mode`
- 8.648.2.5 `uint16_t uim_refreshevent::numOfFiles`
- 8.648.2.6 `uint8_t uim_refreshevent::sessionType`
- 8.648.2.7 `uint8_t uim_refreshevent::stage`

8.649 uim_registerRefresh Struct Reference

Data Fields

- `uint8_t` [registerFlag](#)
- `uint8_t` [voteForInit](#)
- `uint16_t` [numFiles](#)
- `uim_fileInfo` [arrfileInfo](#) [255]

8.649.1 Detailed Description

This structure contains parameters of refresh Information

Parameters

<i>registerFlag</i>	<ul style="list-style-type: none"> • Flag that indicates whether to register or deregister for refresh indications. Valid values: <ul style="list-style-type: none"> – 0 - Deregister – 1 - Register
<i>voteForInit</i>	<ul style="list-style-type: none"> • Flag that indicates whether to vote for the init when there is a refresh. Valid values: <ul style="list-style-type: none"> – 0 - Client does not vote for initialization – 1 - Client votes for initialization
<i>numFiles</i>	<ul style="list-style-type: none"> • Number of sets of the following elements: <ul style="list-style-type: none"> – <code>file_id</code> – <code>path_len</code> – <code>path</code>
<i>arrfileInfo</i>	<ul style="list-style-type: none"> • Array of file Information structure. • See uim_fileInfo for more information

8.649.2 Field Documentation

8.649.2.1 `uim_fileInfo` `uim_registerRefresh::arrfileInfo[255]`

8.649.2.2 `uint16_t` `uim_registerRefresh::numFiles`

8.649.2.3 `uint8_t` `uim_registerRefresh::registerFlag`

8.649.2.4 `uint8_t` `uim_registerRefresh::voteForInit`

8.650 `uim_remainingRetries` Struct Reference

Data Fields

- `uint8_t` [verifyLeft](#)
- `uint8_t` [unblockLeft](#)

8.650.1 Detailed Description

This structure contains the information about the retries remaining.

Parameters

<i>verifyLeft</i>	<ul style="list-style-type: none"> • Number of remaining attempts to verify the PIN. • 0xFF, if unavailable.
<i>unblockLeft</i>	<ul style="list-style-type: none"> • Number of remaining attempts to unblock the PIN. • 0xFF, if unavailable.

Note

This value is returned only when the enable/disable operation has failed. This information is not sent for a hidden key PIN type.

8.650.2 Field Documentation

8.650.2.1 `uint8_t` `uim_remainingRetries::unblockLeft`

8.650.2.2 `uint8_t` `uim_remainingRetries::verifyLeft`

8.651 `uim_sessionInformation` Struct Reference

Data Fields

- `uint8_t` [sessionType](#)
- `uint8_t` [aidLength](#)
- `uint8_t` [aid](#) [255]

8.651.1 Detailed Description

This structure contains the Session Information.

Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> Indicates the session type. <ul style="list-style-type: none"> 0 - Primary GW provisioning 1 - Primary 1X provisioning 2 - Secondary GW provisioning 3 - Secondary 1X provisioning 4 - Non-provisioning on slot 1 5 - Non-provisioning on slot 2 6 - Card on slot 1 7 - Card on slot 2 8 - Logical channel on slot 1 9 - Logical channel on slot 2
<i>aidLength</i>	<ul style="list-style-type: none"> Length of the following elements i.e. Application Identifier.
<i>aid</i>	<ul style="list-style-type: none"> 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.

8.651.2 Field Documentation

8.651.2.1 uint8_t uim_sessionInformation::aid[255]

8.651.2.2 uint8_t uim_sessionInformation::aidLength

8.651.2.3 uint8_t uim_sessionInformation::sessionType

8.652 uim_setPINProtection Struct Reference

Data Fields

- uint8_t [pinID](#)
- uint8_t [pinOperation](#)
- uint8_t [pinLength](#)
- uint8_t [pinValue](#) [255]

8.652.1 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"> Indicates the PIN ID to be enabled or disabled. <ul style="list-style-type: none"> 1 - PIN1 (also called PIN) 2 - PIN2 3 - Universal PIN 4 - Hidden key
--------------	--

<i>pinOperation</i>	<ul style="list-style-type: none"> Indicates whether the PIN is enabled or disabled. <ul style="list-style-type: none"> 0 - Disable the PIN 1 - Enable the PIN
<i>pinLength</i>	<ul style="list-style-type: none"> Length of the following elements i.e. pin value.
<i>pinValue</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> PIN value. This value is a sequence of ASCII characters.

8.652.2 Field Documentation

8.652.2.1 `uint8_t uim_setPINProtection::pinID`

8.652.2.2 `uint8_t uim_setPINProtection::pinLength`

8.652.2.3 `uint8_t uim_setPINProtection::pinOperation`

8.652.2.4 `uint8_t uim_setPINProtection::pinValue[255]`

8.653 uim_simBusyStatus Struct Reference

Data Fields

- `uint8_t simBusyLength`
- `uint8_t simBusy [255]`

8.653.1 Detailed Description

This structure contains Sim Busy Status Information.

Parameters

<i>simBusyLength</i>	<ul style="list-style-type: none"> Number of sets of the following elements. i.e. simBusy
<i>simBusy</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> Indicates the status of the valid card. <ul style="list-style-type: none"> 0 - Sim card is not busy 1 - Sim card is busy

8.653.2 Field Documentation

8.653.2.1 `uint8_t uim_simBusyStatus::simBusy[255]`

8.653.2.2 `uint8_t uim_simBusyStatus::simBusyLength`

8.654 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]

8.654.1 Detailed Description

This structure contains information about the SLOTS present.

Parameters

<i>cardState</i>	<ul style="list-style-type: none"> • Indicates the state of the card for each slot. <ul style="list-style-type: none"> – 0 - Absent – 1 - Present – 2 - Error
<i>upinState</i>	<ul style="list-style-type: none"> • Indicates the state of UPIN. <ul style="list-style-type: none"> – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked – 0xFF - Not Available
<i>upinRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to verify the UPIN. • If 0xFF, information not available.
<i>upukRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock the UPIN. • If 0xFF, information not available.

<i>errorState</i>	<ul style="list-style-type: none"> Indicates the reason for the card error, and is valid only when the card state is Error <ul style="list-style-type: none"> 0 - Unknown 1 - Power down 2 - Poll error 3 - No ATR received 4 - Volt mismatch 5 - Parity error 6 - Unknown; possibly removed 7 - Card returned technical problems 0xFF - Not Available Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as "Unknown".
<i>numApp</i>	<ul style="list-style-type: none"> Indicates the number of applications available on the card. The following block is repeated for each application. i.e. AppStatus. If zero(0) then no AppStatus information exists.
<i>AppStatus</i>	<ul style="list-style-type: none"> See uim_appStatus for more information.

8.654.2 Field Documentation

8.654.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.654.2.2 `uint8_t uim_slotInfo::cardState`

8.654.2.3 `uint8_t uim_slotInfo::errorState`

8.654.2.4 `uint8_t uim_slotInfo::numApp`

8.654.2.5 `uint8_t uim_slotInfo::upinRetries`

8.654.2.6 `uint8_t uim_slotInfo::upinState`

8.654.2.7 `uint8_t uim_slotInfo::upukRetries`

8.655 uim_UIMSessionInformation Struct Reference

Data Fields

- `uint8_t sessionType`
- `uint8_t aidLength`
- `uint8_t aid [255]`

8.655.1 Detailed Description

This structure contains the Session Information.

Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> Indicates the session type. <ul style="list-style-type: none"> 0 - Primary GW provisioning 1 - Primary 1X provisioning 2 - Secondary GW provisioning 3 - Secondary 1X provisioning 4 - Non-provisioning on slot 1 5 - Non-provisioning on slot 2 6 - Card on slot 1 7 - Card on slot 2 8 - Logical channel on slot 1 9 - Logical channel on slot 2
<i>aidLength</i>	<ul style="list-style-type: none"> Length of the following elements i.e. Application Identifier.
<i>aid</i>	<ul style="list-style-type: none"> 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.

8.655.2 Field Documentation

8.655.2.1 uint8_t uim_UIMSessionInformation::aid[255]

8.655.2.2 uint8_t uim_UIMSessionInformation::aidLength

8.655.2.3 uint8_t uim_UIMSessionInformation::sessionType

8.656 uim_unblockUIMPIN Struct Reference

Data Fields

- uint8_t [pinID](#)
- uint8_t [pukLen](#)
- uint8_t [pukVal](#) [255]
- uint8_t [newPINLen](#)
- uint8_t [newPINVal](#) [255]

8.656.1 Detailed Description

This structure contains the information about the unblock pin parameters.

Parameters

<i>pinID</i>	<ul style="list-style-type: none"> Indicates the PIN ID to be changed. <ul style="list-style-type: none"> 1 - PIN1 (also called PIN) 2 - PIN2 3 - Universal PIN
--------------	--

<i>pukLen</i>	<ul style="list-style-type: none"> Length of the following elements i.e. puk value.
<i>pukVal[UIM_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> PIN Unlock Key value. This value is a sequence of ASCII characters.
<i>newPINLen</i>	<ul style="list-style-type: none"> Length of the following elements i.e. new pin value.
<i>newPINVal[UIM_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> New PIN value. This value is a sequence of ASCII characters.

8.656.2 Field Documentation

8.656.2.1 `uint8_t uim_unblockUIMPIN::newPINLen`

8.656.2.2 `uint8_t uim_unblockUIMPIN::newPINVal[255]`

8.656.2.3 `uint8_t uim_unblockUIMPIN::pinID`

8.656.2.4 `uint8_t uim_unblockUIMPIN::pukLen`

8.656.2.5 `uint8_t uim_unblockUIMPIN::pukVal[255]`

8.657 uim_validCardStatus Struct Reference

Data Fields

- `uint8_t validCardLength`
- `uint8_t validCard [255]`

8.657.1 Detailed Description

This structure contains Valid Card Status Information.

Parameters

<i>validCardLength</i>	<ul style="list-style-type: none"> Number of sets of the following elements. i.e. validCard
<i>validCard[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> Indicates the status of the valid card. <ul style="list-style-type: none"> 0 - Status of the card is unknown 1 - Status of the card is valid

8.657.2 Field Documentation

8.657.2.1 uint8_t uim_validCardStatus::validCard[255]

8.657.2.2 uint8_t uim_validCardStatus::validCardLength

8.658 uim_verifyUIMPIN Struct Reference

Data Fields

- uint8_t [pinID](#)
- uint8_t [pinLen](#)
- uint8_t [pinVal](#) [255]

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

8.658.2 Field Documentation

8.658.2.1 uint8_t uim_verifyUIMPIN::pinID

8.658.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.658.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.659 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`

8.659.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig response parameters.

Parameters

<i>pECMode</i>	[Optional] <ul style="list-style-type: none"> • AV_EC <ul style="list-style-type: none"> – 0 - Echo cancellation off – 1 - Handset mode – 2 - Headset mode – 3 - Car kit mode – 4 - Speaker Mode • Bit to check in ParamPresenceMask - 33
<i>pNSEnable</i>	[Optional] <ul style="list-style-type: none"> • AV_NS <ul style="list-style-type: none"> – 0 - Noise suppression off – 1 - Noise suppression on • Bit to check in ParamPresenceMask - 34
<i>pTXGain</i>	[Optional] <ul style="list-style-type: none"> • AV_TXVOL <ul style="list-style-type: none"> – 0x0000 - 0xffff • Bit to check in ParamPresenceMask - 35
<i>pDTMFTXGain</i>	[Optional] <ul style="list-style-type: none"> • AV_DTMFTXG <ul style="list-style-type: none"> – 0x0000 - 0xffff • Bit to check in ParamPresenceMask - 36
<i>pCodecSTGain</i>	[Optional] <ul style="list-style-type: none"> • AV_CODECASTG <ul style="list-style-type: none"> – 0x0000 - 0xffff • Bit to check in ParamPresenceMask - 37
<i>pTXPCMIIRFtr</i>	[Optional] <ul style="list-style-type: none"> • See audio_TXPCMIIRFtr for more information • Bit to check in ParamPresenceMask - 38
<i>pRXPCMIIRFtr</i>	[Optional] <ul style="list-style-type: none"> • See audio_RXPCMIIRFtr for more information • Bit to check in ParamPresenceMask - 39

<i>pMICGainSelect</i>	[Optional] <ul style="list-style-type: none"> • AV_MICGAIN • Bit to check in ParamPresenceMask - 40
<i>pRXAVCAGC-Switch</i>	[Optional] <ul style="list-style-type: none"> • RX AVC/AGC Switch • Bit to check in ParamPresenceMask - 41
<i>pTXAVCSwitch</i>	[Optional] <ul style="list-style-type: none"> • TX AVC Switch • Bit to check in ParamPresenceMask - 42
<i>pRXAGCList</i>	[Optional] <ul style="list-style-type: none"> • See audio_RXAGCList for more information • Bit to check in ParamPresenceMask - 43
<i>pRXAVCList</i>	[Optional] <ul style="list-style-type: none"> • See audio_RXAVCList for more information • Bit to check in ParamPresenceMask - 44
<i>pTXAGCList</i>	[Optional] <ul style="list-style-type: none"> • See audio_TXAGCList for more information • Bit to check in ParamPresenceMask - 45

8.659.2 Field Documentation

8.659.2.1 `swi_uint256_t unpack_audio_SLQSGetAudioPathConfig_t::ParamPresenceMask`

8.659.2.2 `uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pCodecSTGain`

8.659.2.3 `uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pDTMFTXGain`

8.659.2.4 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pECMode`

8.659.2.5 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pMICGainSelect`

8.659.2.6 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pNSEnable`

8.659.2.7 `audio_RXAGCList* unpack_audio_SLQSGetAudioPathConfig_t::pRXAGCList`

8.659.2.8 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pRXAVCAGCSwitch`

8.659.2.9 `audio_RXAVCList* unpack_audio_SLQSGetAudioPathConfig_t::pRXAVCList`

8.659.2.10 `audio_RXPCMIIRFitr* unpack_audio_SLQSGetAudioPathConfig_t::pRXPCMIIRFitr`

8.659.2.11 `audio_TXAGCList* unpack_audio_SLQSGetAudioPathConfig_t::pTXAGCList`

8.659.2.12 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pTXAVCSwitch`

8.659.2.13 `uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pTXGain`

8.659.2.14 `audio_TXPCMIIRFiltr*` `unpack_audio_SLQSGetAudioPathConfig_t::pTXPCMIIRFiltr`

8.660 `unpack_audio_SLQSGetAudioProfile_t` Struct Reference

Data Fields

- `uint8_t` [Profile](#)
- `uint8_t` [EarMute](#)
- `uint8_t` [MicMute](#)
- `uint8_t` [Volume](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.660.1 Detailed Description

This structure contains the unpack parameters to Get Audio Profile.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile <ul style="list-style-type: none"> – 0 - Handset – 1 - Headset – 2 - Car Kit – 3 - Speaker phone – 4 - Auxiliary – 5 - TTY – 6 - Auxiliary external PCM – 7 - Primary external PCM – 8 - External slave PCM – 9 - I2S • Bit to check in ParamPresenceMask - 16
<i>EarMute</i>	<ul style="list-style-type: none"> • Ear Mute Setting <ul style="list-style-type: none"> – 0 - unmuted – 1 - muted • Bit to check in ParamPresenceMask - 17
<i>MicMute</i>	<ul style="list-style-type: none"> • MIC Mute Setting <ul style="list-style-type: none"> – 0 - unmuted – 1 - muted • Bit to check in ParamPresenceMask - 18
<i>Volume</i>	<ul style="list-style-type: none"> • Audio Volume Level <ul style="list-style-type: none"> – 0 to 7 • Bit to check in ParamPresenceMask - 20

8.660.2 Field Documentation

8.660.2.1 uint8_t unpack_audio_SLQSGetAudioProfile_t::EarMute

8.660.2.2 uint8_t unpack_audio_SLQSGetAudioProfile_t::MicMute

8.660.2.3 swi_uint256_t unpack_audio_SLQSGetAudioProfile_t::ParamPresenceMask

8.660.2.4 uint8_t unpack_audio_SLQSGetAudioProfile_t::Profile

8.660.2.5 uint8_t unpack_audio_SLQSGetAudioProfile_t::Volume

8.661 unpack_audio_SLQSGetAudioVoITLBConfig_t Struct Reference

Data Fields

- uint16_t [ResCode](#)
- swi_uint256_t [ParamPresenceMask](#)

8.661.1 Detailed Description

This structure contains the unpack parameters for SLQSGetAudioVoITLBConfig.

Parameters

<i>ResCode</i>	<ul style="list-style-type: none">• Result of requested item• Bit to check in ParamPresenceMask - 33
----------------	--

8.661.2 Field Documentation

8.661.2.1 swi_uint256_t unpack_audio_SLQSGetAudioVoITLBConfig_t::ParamPresenceMask

8.661.2.2 uint16_t unpack_audio_SLQSGetAudioVoITLBConfig_t::ResCode

8.662 unpack_audio_SLQSSetAudioVoITLBConfig_t Struct Reference

Data Fields

- uint16_t [ResCode](#)
- swi_uint256_t [ParamPresenceMask](#)

8.662.1 Detailed Description

This structure contains the unpack parameters for SLQSSetAudioVoITLBConfig.

Parameters

<i>ResCode</i>	<ul style="list-style-type: none">• Result of requested item• Bit to check in ParamPresenceMask - 33
----------------	--

8.662.2 Field Documentation

8.662.2.1 `swi_uint256_t` `unpack_audio_SLQSSetAudioVoITLBConfig_t::ParamPresenceMask`

8.662.2.2 `uint16_t` `unpack_audio_SLQSSetAudioVoITLBConfig_t::ResCode`

8.663 `unpack_cat_SetCatEventCallback_ind_t` Struct Reference

Data Fields

- `uint8_t` `event_Index`
- struct `cat_commonEventTlv` `CCETlv` [11]
- `swi_uint256_t` `ParamPresenceMask`

8.663.1 Detailed Description

Structure used to store all CAT Notification Parameters.

Parameters

<i>event_Index</i>	- Event Index
<i>CCETlv</i>	- CAT common event Tlv <code>cat_commonEventTlv</code> for more info. <ul style="list-style-type: none"> • [0] - Bit to check in ParamPresenceMask for Display Text - 16 • [1] - Bit to check in ParamPresenceMask for Get In-Key - 17 • [2] - Bit to check in ParamPresenceMask for Get Input - 18 • [3] - Bit to check in ParamPresenceMask for Setup Menu - 19 • [4] - Bit to check in ParamPresenceMask for Select Item - 20 • [5] - Bit to check in ParamPresenceMask for Alpha Identifier - 21 • [6] - Bit to check in ParamPresenceMask for Setup Event List - 22 • [7] - Bit to check in ParamPresenceMask for Setup Idle Mode Text - 23 • [8] - Bit to check in ParamPresenceMask for Language Notification - 24 • [9] - Bit to check in ParamPresenceMask for Refresh - 25 • [10] - Bit to check in ParamPresenceMask for End Proactive Session - 26

8.663.2 Field Documentation

8.663.2.1 `struct cat_commonEventTlv` `unpack_cat_SetCatEventCallback_ind_t::CCETlv`[11]

8.663.2.2 `uint8_t` `unpack_cat_SetCatEventCallback_ind_t::event_Index`

8.663.2.3 `swi_uint256_t` `unpack_cat_SetCatEventCallback_ind_t::ParamPresenceMask`

8.664 `unpack_cat_SetCATEventCallback_t` Struct Reference

Data Fields

- `uint32_t` `errorMask`
- `uint16_t` `Tlvresult`
- `swi_uint256_t` `ParamPresenceMask`

8.664.1 Detailed Description

This structure contains sEnables/disables the CAT event callback unpack variable.

Parameters

<i>errorMask</i>	<ul style="list-style-type: none"> error bitmask. Each bit set indicates the proactive command that caused the error <ul style="list-style-type: none"> 0x00000001 - Display Text 0x00000002 - Get In-Key 0x00000004 - Get Input 0x00000008 - Setup Menu 0x00000010 - Select Item 0x00000020 - Send SMS - Alpha Identifier 0x00000040 - Setup Event: User Activity 0x00000080 - Setup Event: Idle Screen Notify 0x00000100 - Setup Event: Language Sel Notify 0x00000200 - Setup Idle Mode Text 0x00000400 - Language Notification 0x00000800 - Refresh 0x00001000 - End Proactive Session Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result

8.664.2 Field Documentation

8.664.2.1 `uint32_t unpack_cat_SetCATEventCallback_t::errorMask`

8.664.2.2 `swi_uint256_t unpack_cat_SetCATEventCallback_t::ParamPresenceMask`

8.664.2.3 `uint16_t unpack_cat_SetCATEventCallback_t::Tlvresult`

8.665 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- `uint8_t state`
- `swi_uint256_t ParamPresenceMask`

8.665.1 Detailed Description

This structure contains unpack_dms_GetActivationState parameters.

Parameters

<i>state[OUT]</i>	<ul style="list-style-type: none"> • Service Activation Code <ul style="list-style-type: none"> 0 - Service not activated 1 - Service activated 2 - Activation connecting 3 - Activation connected 4 - OTASP security authenticated 5 - OTASP NAM downloaded 6 - OTASP MDN downloaded 7 - OTASP IMSI downloaded 8 - OTASP PRL downloaded 9 - OTASP SPC downloaded 10 - OTASP settings committed • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.665.2 Field Documentation

8.665.2.1 `swi_uint256_t unpack_dms_GetActivationState_t::ParamPresenceMask`8.665.2.2 `uint8_t unpack_dms_GetActivationState_t::state`8.666 `unpack_dms_GetBandCapability_t` Struct Reference

Data Fields

- `uint64_t` [BandCapability](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.666.1 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"> • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19 • Bit 60 - WCDMA Japan 850 band • Bit 61 - WCDMA 1500 band • Bits 62 and 63 - Reserved
-----------------------	---

- Bit to check in ParamPresenceMask - 1

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.666.2 Field Documentation

8.666.2.1 uint64_t unpack_dms_GetBandCapability_t::BandCapability

8.666.2.2 swi_uint256_t unpack_dms_GetBandCapability_t::ParamPresenceMask

8.666.2.3 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.667 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

- uint8_t [DevCrashState](#)
- uint16_t [Tlvresult](#)
- swi_uint256_t [ParamPresenceMask](#)

8.667.1 Detailed Description

This structure is used to store the Crash State from the device.

Parameters

<i>DevCrashState</i>	<ul style="list-style-type: none"> Device Crash State Values: <ul style="list-style-type: none"> 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. 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 2 - No action Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.667.2 Field Documentation

8.667.2.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.667.2.2 swi_uint256_t unpack_dms_GetCrashAction_t::ParamPresenceMask

8.667.2.3 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.668 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](#)

8.668.1 Detailed Description

This structure contains current settings of custom features

Parameters

<i>GpsEnable</i>	<ul style="list-style-type: none"> • describes if GPS is enabled or disabled • values: <ul style="list-style-type: none"> – 0x00 - GPS is disabled – 0x01 - GPS is enabled • Bit to check in ParamPresenceMask - 16
<i>DisableIMSI</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • describes if IMSI display is enabled or disabled • values: <ul style="list-style-type: none"> – 0x00 - Allow display of IMSI – 0x01 - Do not display IMSI • Bit to check in ParamPresenceMask - 17
<i>IPFamSupport</i>	<ul style="list-style-type: none"> • optional 2 byte BitMask • bitmask representing the IP families supported • values: <ul style="list-style-type: none"> – 0x01 - IPv4 – 0x02 - IPv6 – 0x04 - IPv4v6 • Bit to check in ParamPresenceMask - 18

<i>RMAutoConnect</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • QMI Mode RM Net Auto Connect Support • values: <ul style="list-style-type: none"> – 0x00 - Not Supported – 0x01 - Supported • Bit to check in ParamPresenceMask - 19
<i>GPSSel</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • GPS Antenna Select • values: <ul style="list-style-type: none"> – 0x00 - Dedicated GPS Port – 0x01 - GPS Rx over AUX Port – 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied • Bit to check in ParamPresenceMask - 20
<i>SMSSupport</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • SMS support • values: <ul style="list-style-type: none"> – 0x00 - Not supported – 0x01 - supported • Bit to check in ParamPresenceMask - 21 • Used to determine whether or not to hide SMS from user
<i>IsVoiceEnabled</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • Voice support • values: <ul style="list-style-type: none"> – 0x00 - Enable voice on both AT and QMI interface (default) – 0x01 - Reserved – 0x02 - Disable voice on both AT and QMI interface • Bit to check in ParamPresenceMask - 22
<i>DHCPRelay-Enabled</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • DHCP Relay support • values: <ul style="list-style-type: none"> – 0x00 - Disable DHCP relay – 0x01 - Enable DHCP relay • Bit to check in ParamPresenceMask - 23
<i>GPSSLPM</i>	<ul style="list-style-type: none"> • optional 1 byte parameter • GPSSLPM support • values: <ul style="list-style-type: none"> – 0x00 - Enable GPS in Low Power Mode – 0x01 - Disable GPS in Low Power Mode • Bit to check in ParamPresenceMask - 24

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.668.2 Field Documentation

8.668.2.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.668.2.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.668.2.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.668.2.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.668.2.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.668.2.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.668.2.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

8.668.2.8 swi_uint256_t unpack_dms_GetCustFeature_t::ParamPresenceMask

8.668.2.9 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

8.668.2.10 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

8.668.2.11 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

8.669 unpack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- [DMSgetCustomFeatureV2 GetCustomFeatureV2](#)
- uint16_t *Tlvresult*
- [swi_uint256_t ParamPresenceMask](#)

8.669.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

<i>GetCustomFeatureV2</i>	<ul style="list-style-type: none"> See DMSgetCustomFeatureV2 for more information pCustSettingInfo <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 32 pCustSettingList <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 33
---------------------------	--

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.669.2 Field Documentation

8.669.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

8.669.2.2 swi_uint256_t unpack_dms_GetCustFeaturesV2_t::ParamPresenceMask

8.669.2.3 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

8.670 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](#)

8.670.1 Detailed Description

Structure to store the device capabilities

Parameters

<i>MaxTXChannelRate</i>	<ul style="list-style-type: none"> • Maximum transmission rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies • Bit to check in ParamPresenceMask - 1
<i>MaxRXChannelRate</i>	<ul style="list-style-type: none"> • Maximum reception rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies • Bit to check in ParamPresenceMask - 1
<i>DataServiceCapability</i>	<ul style="list-style-type: none"> • CS/PS data service capability <ul style="list-style-type: none"> – 0 - No data services supported – 1 - Only Circuit Switched (CS) services supported – 2 - Only Packet Switched (PS) services supported – 3 - Simultaneous CS and PS – 4 - Non-simultaneous CS and PS • Bit to check in ParamPresenceMask - 1

<i>SimCapability</i>	Device SIM capability <ul style="list-style-type: none"> • 0 - SIM not supported • 1 - SIM supported
----------------------	--

- Bit to check in ParamPresenceMask - 1

Parameters

<i>RadiofacesSize</i>	<ul style="list-style-type: none"> • Upon input, the maximum number of elements that the radio interface array can contain • Upon successful output, actual number of elements in the radio interface array • Bit to check in ParamPresenceMask - 1
<i>Radiofaces</i>	<ul style="list-style-type: none"> • Radio interface array. This is a structure of array containing the elements below. uint8_t Radiofaces <ul style="list-style-type: none"> – See qaGobiApiTableRadioInterfaces.h for Radio Interfaces • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.670.2 Field Documentation

8.670.2.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

8.670.2.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

8.670.2.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

8.670.2.4 swi_uint256_t unpack_dms_GetDeviceCap_t::ParamPresenceMask

8.670.2.5 uint8_t unpack_dms_GetDeviceCap_t::Radiofaces[255]

8.670.2.6 uint32_t unpack_dms_GetDeviceCap_t::RadiofacesSize

8.670.2.7 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.670.2.8 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.671 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](#)

8.671.1 Detailed Description

Gets the device capabilities structure

Parameters

<i>maxTxChannel-Rate</i>	<ul style="list-style-type: none"> • Maximum transmission rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies • Bit to check in ParamPresenceMask - 1
<i>maxRxChannel-Rate</i>	<ul style="list-style-type: none"> • Maximum reception rate (in bps) supported by the device • In multi-technology devices, this value will be the greatest rate among all supported technologies • Bit to check in ParamPresenceMask - 1
<i>dataServiceCa-Capability</i>	<ul style="list-style-type: none"> • CS/PS data service capability <ul style="list-style-type: none"> – 0 - No data services supported – 1 - Only Circuit Switched (CS) services supported – 2 - Only Packet Switched (PS) services supported – 3 - Simultaneous CS and PS – 4 - Non-simultaneous CS and PS • Bit to check in ParamPresenceMask - 1
<i>simCapability</i>	<ul style="list-style-type: none"> • Device SIM capability <ul style="list-style-type: none"> – 0 - SIM not supported – 1 - SIM supported • Bit to check in ParamPresenceMask - 1
<i>radiolfacesSize</i>	<ul style="list-style-type: none"> • Upon input, the maximum number of elements that the radio interface array can contain • Upon successful output, actual number of elements in the radio interface array • Bit to check in ParamPresenceMask - 1
<i>Radiolfaces[OUT]</i>	<ul style="list-style-type: none"> • Radio interface array. This is a structure of array containing the elements below. <ul style="list-style-type: none"> – See qaGobiApiTableRadioInterfaces.h for Radio Interfaces • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.671.2 Field Documentation

- 8.671.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability
- 8.671.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate
- 8.671.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate
- 8.671.2.4 swi_uint256_t unpack_dms_GetDeviceCapabilities_t::ParamPresenceMask
- 8.671.2.5 uint8_t unpack_dms_GetDeviceCapabilities_t::Radiofaces[255]
- 8.671.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::radiofacesSize
- 8.671.2.7 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

8.672 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

8.672.1 Detailed Description

This structure contains the unpack_dms_GetDeviceCapabilitiesV2 response parameters.

Parameters

<i>DevCaps</i>	<ul style="list-style-type: none"> • See dms_devCaps for more information • Bit to check in ParamPresenceMask - 1
<i>pDevSrvCaps</i>	<ul style="list-style-type: none"> • Device service capabilities <ul style="list-style-type: none"> – DMS_DEVICE_CAP_DATA_ONLY -0x01 – DMS_DEVICE_CAP_VOICE_ONLY -0x02 – DMS_DEVICE_CAP_SIMUL_VOICE_AND_DATA -0x03 – DMS_DEVICE_CAP_NONSIMUL_VOICE_AND_DATA -0x04 • Bit to check in ParamPresenceMask - 16

<i>pDevVoiceCaps</i>	<ul style="list-style-type: none"> • Device voice capabilities <ul style="list-style-type: none"> – Bit 0 – GW CSFB <ul style="list-style-type: none"> * 0 – Not capable * 1 – Capable – Bit 1 – 1x CSFB <ul style="list-style-type: none"> * 0 – Not capable * 1 – Capable – Bit 2 – VoLTE <ul style="list-style-type: none"> * 0 – Not capable * 1 – Capable
----------------------	--

- Bit to check in ParamPresenceMask - **17**

Parameters

<i>pDevVoiceDataCaps</i>	<ul style="list-style-type: none"> • DevVoiceDataCaps <ul style="list-style-type: none"> – Bit 0 – SVLTE capability – Bit 1 – SVDO capability – Bit 2 – SGLTE capability • Bit to check in ParamPresenceMask - 18
<i>pDevMultiSimCaps</i>	<ul style="list-style-type: none"> • See dms_devMultiSimCaps for more information • Bit to check in ParamPresenceMask - 19
<i>pDevMultiSimVoiceDataCaps</i>	<ul style="list-style-type: none"> • See dms_devMultiSimVoiceDataCaps for more information • Bit to check in ParamPresenceMask - 20
<i>pDevCurSubsCaps</i>	<ul style="list-style-type: none"> • See dms_devCurSubsCaps for more information • Bit to check in ParamPresenceMask - 21
<i>pDevSubsVoiceDataCaps</i>	<ul style="list-style-type: none"> • See dms_devSubsVoiceDataCaps for more information • Bit to check in ParamPresenceMask - 22
<i>pDevSubsFeatureModeCaps</i>	<ul style="list-style-type: none"> • See dms_devSubsFeatureModeCaps for more information • Bit to check in ParamPresenceMask - 23
<i>pDevMaxActDataSubsCaps</i>	<ul style="list-style-type: none"> • Max number of subscriptions for data activity • Bit to check in ParamPresenceMask - 24
<i>pDevMaxSubsCaps</i>	<ul style="list-style-type: none"> • See dms_devMaxSubsCaps for more information • Bit to check in ParamPresenceMask - 25

<i>pDevMaxAct-DataSubsCaps</i>	<ul style="list-style-type: none"> • See dms_devMaxCfgListCaps for more information • Bit to check in ParamPresenceMask - 26
<i>pDevExplicitCfg-Index</i>	<ul style="list-style-type: none"> • Explicit Cfg Index <ul style="list-style-type: none"> – (-1) - Modem controlled cfg – any other valid value • Bit to check in ParamPresenceMask - 27
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.672.2 Field Documentation

8.672.2.1 **dms_devCaps** unpack_dms_GetDeviceCapabilitiesV2_t::DevCaps

8.672.2.2 **swi_uint256_t** unpack_dms_GetDeviceCapabilitiesV2_t::ParamPresenceMask

8.672.2.3 **dms_devCurSubsCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevCurSubsCaps

8.672.2.4 **int16_t*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevExplicitCfgIndex

8.672.2.5 **uint8_t*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevMaxActDataSubsCaps

8.672.2.6 **dms_devMaxCfgListCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevMaxCfgListCaps

8.672.2.7 **dms_devMaxSubsCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevMaxSubsCaps

8.672.2.8 **dms_devMultiSimCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevMultiSimCaps

8.672.2.9 **dms_devMultiSimVoiceDataCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevMultiSimVoiceDataCaps

8.672.2.10 **uint32_t*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevSrvCaps

8.672.2.11 **dms_devSubsFeatureModeCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevSubsFeatureModeCaps

8.672.2.12 **dms_devSubsVoiceDataCaps*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevSubsVoiceDataCaps

8.672.2.13 **uint64_t*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevVoiceCaps

8.672.2.14 **uint64_t*** unpack_dms_GetDeviceCapabilitiesV2_t::pDevVoiceDataCaps

8.672.2.15 **uint16_t** unpack_dms_GetDeviceCapabilitiesV2_t::Tlvresult

8.673 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

- uint8_t [stringSize](#)
- char [String](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.673.1 Detailed Description

Structure to store the hardware revision of the device

Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> • The maximum number of characters (including NULL terminator) that the string array can contain • Bit to check in ParamPresenceMask - 1
<i>String</i>	<ul style="list-style-type: none"> • NULL terminated Hardware Revision string • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.673.2 Field Documentation

8.673.2.1 [swi_uint256_t](#) [unpack_dms_GetDeviceHardwareRev_t::ParamPresenceMask](#)

8.673.2.2 [char](#) [unpack_dms_GetDeviceHardwareRev_t::String](#)[255]

8.673.2.3 [uint8_t](#) [unpack_dms_GetDeviceHardwareRev_t::stringSize](#)

8.673.2.4 [uint16_t](#) [unpack_dms_GetDeviceHardwareRev_t::Tlvresult](#)

8.674 [unpack_dms_GetDeviceMfr_t](#) Struct Reference

Data Fields

- uint8_t [stringSize](#)
- char [String](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.674.1 Detailed Description

This structure is used to store [unpack_dms_GetDeviceMfr](#) parameters

Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> The maximum number of characters (including NULL terminator) that the string array can contain
-------------------	--

- Bit to check in ParamPresenceMask - 1

Parameters

<i>String</i>	<ul style="list-style-type: none"> NULL terminated Device Manufacture string Maximum Length is 255 Bytes Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.674.2 Field Documentation

8.674.2.1 swi_uint256_t unpack_dms_GetDeviceMfr_t::ParamPresenceMask

8.674.2.2 char unpack_dms_GetDeviceMfr_t::String[255]

8.674.2.3 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

8.674.2.4 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

8.675 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](#)

8.675.1 Detailed Description

This structure used to store unpack_dms_GetDeviceSerialNumbers parameters

Parameters

<i>esnSize</i>	<ul style="list-style-type: none"> The maximum number of characters (including NULL terminator) that the ESN string array can contain Bit to check in ParamPresenceMask - 16
<i>ESNString</i>	<ul style="list-style-type: none"> NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed Maximum Length is 255 Bytes Bit to check in ParamPresenceMask - 16
<i>imeiSize</i>	<ul style="list-style-type: none"> The maximum number of characters (including NULL terminator) that the IMEI string array can contain Bit to check in ParamPresenceMask - 17
<i>IMEIString</i>	<ul style="list-style-type: none"> NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed Maximum Length is 255 Bytes Bit to check in ParamPresenceMask - 17
<i>meidSize</i>	<ul style="list-style-type: none"> The maximum number of characters (including NULL terminator) that the MEID string array can contain Bit to check in ParamPresenceMask - 18
<i>MEIDString</i>	<ul style="list-style-type: none"> NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed Maximum Length is 255 Bytes Bit to check in ParamPresenceMask - 18
<i>imeiSvnSize</i>	<ul style="list-style-type: none"> The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain. Bit to check in ParamPresenceMask - 19
<i>ImeiSvnString</i>	<ul style="list-style-type: none"> NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed. Maximum Length is 255 Bytes Bit to check in ParamPresenceMask - 19
<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.675.2 Field Documentation

- 8.675.2.1 `uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize`
- 8.675.2.2 `char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]`
- 8.675.2.3 `uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize`
- 8.675.2.4 `char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]`
- 8.675.2.5 `uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize`
- 8.675.2.6 `char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]`
- 8.675.2.7 `uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize`
- 8.675.2.8 `char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]`
- 8.675.2.9 `swi_uint256_t unpack_dms_GetDeviceSerialNumbers_t::ParamPresenceMask`
- 8.675.2.10 `uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult`

8.676 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`

8.676.1 Detailed Description

This structure used to store `unpack_dms_GetFirmwareInfo` parameters

Parameters

<i>modelid_str</i>	<ul style="list-style-type: none"> - Mode ID String. <ul style="list-style-type: none"> • NULL-terminated Mode ID String. • Bit to check in ParamPresenceMask - 16
<i>bootversion_str</i>	<ul style="list-style-type: none"> - Boot Version. <ul style="list-style-type: none"> • NULL-terminated Boot Version String. • Bit to check in ParamPresenceMask - 17
<i>appversion_str</i>	<ul style="list-style-type: none"> - Application Version String. <ul style="list-style-type: none"> • NULL-terminated Application Version String. • Bit to check in ParamPresenceMask - 18

<i>sku_str</i>	- SKU String. <ul style="list-style-type: none"> • NULL-terminated SKU String. • Bit to check in ParamPresenceMask - 19
<i>packageid_str</i>	- Package ID String. <ul style="list-style-type: none"> • NULL-terminated Package ID String. • deprecated on EM/MC74xx(9x30) devices • Bit to check in ParamPresenceMask - 20
<i>carrier_str</i>	- Carrier String. <ul style="list-style-type: none"> • NULL-terminated Carrier String. • Bit to check in ParamPresenceMask - 21
<i>prversion_str</i>	- PRI Version String. <ul style="list-style-type: none"> • NULL-terminated PRI Version String. • Bit to check in ParamPresenceMask - 22
<i>cur_carr_name</i>	- Current Carrier Name String. <ul style="list-style-type: none"> • NULL-terminated Current Carrier Name String. • Bit to check in ParamPresenceMask - 23
<i>cur_carr_rev</i>	- Current Carrier Revision String. <ul style="list-style-type: none"> • NULL-terminated Current Carrier Revision String. • Bit to check in ParamPresenceMask - 24
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.676.2 Field Documentation

8.676.2.1 `char unpack_dms_GetFirmwareInfo_t::appversion_str[85]`

8.676.2.2 `char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]`

8.676.2.3 `char unpack_dms_GetFirmwareInfo_t::carrier_str[20]`

8.676.2.4 `char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]`

8.676.2.5 `char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]`

8.676.2.6 `char unpack_dms_GetFirmwareInfo_t::modelid_str[20]`

8.676.2.7 `char unpack_dms_GetFirmwareInfo_t::packageid_str[85]`

8.676.2.8 `swi_uint256_t unpack_dms_GetFirmwareInfo_t::ParamPresenceMask`

8.676.2.9 `char unpack_dms_GetFirmwareInfo_t::prversion_str[16]`

8.676.2.10 `char unpack_dms_GetFirmwareInfo_t::sku_str[15]`

8.676.2.11 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.677 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

- uint8_t [amssSize](#)
- char [AMSSString](#) [255]
- char [PRIString](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.677.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareRevision parameters

Parameters

<i>amssSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that the AMSS string array can contain. • Bit to check in ParamPresenceMask - 1
<i>AMSSString</i>	<ul style="list-style-type: none"> • NULL-terminated AMSS revision string. • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>PRIString[OUT]</i>	<ul style="list-style-type: none"> • NULL-terminated PRI revision string. • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.677.2 Field Documentation

8.677.2.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.677.2.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.677.2.3 swi_uint256_t unpack_dms_GetFirmwareRevision_t::ParamPresenceMask

8.677.2.4 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

8.677.2.5 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

8.678 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](#)

8.678.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareRevisions parameters

Parameters

<i>amssSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that the AMSS string array can contain • Bit to check in ParamPresenceMask - 1
<i>AMSSString</i>	<ul style="list-style-type: none"> • NULL-terminated AMSS revision string • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>bootSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that the boot string array can contain • Bit to check in ParamPresenceMask - 16
<i>BootString</i>	<ul style="list-style-type: none"> • NULL-terminated boot code revision string • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 16
<i>priSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that the PRI string array can contain • Bit to check in ParamPresenceMask - 17
<i>PRIString</i>	<ul style="list-style-type: none"> • NULL-terminated PRI revision string • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.678.2 Field Documentation

8.678.2.1 `uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize`

8.678.2.2 `char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]`

8.678.2.3 `uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize`

8.678.2.4 `char unpack_dms_GetFirmwareRevisions_t::BootString[255]`

8.678.2.5 `swi_uint256_t unpack_dms_GetFirmwareRevisions_t::ParamPresenceMask`

8.678.2.6 `uint8_t unpack_dms_GetFirmwareRevisions_t::priSize`

8.678.2.7 `char unpack_dms_GetFirmwareRevisions_t::PRIString[255]`

8.678.2.8 `uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult`

8.679 unpack_dms_GetFSN_t Struct Reference

Data Fields

- `char String [255]`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.679.1 Detailed Description

This structure used to store Factory Sequence Number parameter

Parameters

<i>String</i>	<ul style="list-style-type: none"> • Factory Sequence Number • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.679.2 Field Documentation

8.679.2.1 `swi_uint256_t unpack_dms_GetFSN_t::ParamPresenceMask`

8.679.2.2 `char unpack_dms_GetFSN_t::String[255]`

8.679.2.3 `uint16_t unpack_dms_GetFSN_t::Tlvresult`

8.680 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char [hwVer](#) [255]
- [swi_uint256_t ParamPresenceMask](#)

8.680.1 Detailed Description

This structure used to store unpack_dms_GetHardwareRevision parameters

Parameters

<i>hwVer</i>	<ul style="list-style-type: none"> - Hardware version <ul style="list-style-type: none"> • NULL-terminated string • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.680.2 Field Documentation

8.680.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.680.2.2 swi_uint256_t unpack_dms_GetHardwareRevision_t::ParamPresenceMask

8.681 unpack_dms_GetIMSI_t Struct Reference

Data Fields

- char [imsi](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.681.1 Detailed Description

This structure used to store unpack_dms_GetIMSI parameters

Parameters

<i>imsi</i>	<ul style="list-style-type: none"> - IMSI no. <ul style="list-style-type: none"> • NULL-terminated String. • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> - unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.681.2 Field Documentation

8.681.2.1 char unpack_dms_GetIMSI_t::imsi[255]

8.681.2.2 swi_uint256_t unpack_dms_GetIMSI_t::ParamPresenceMask

8.681.2.3 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.682 unpack_dms_GetManufacturer_t Struct Reference

Data Fields

- char [manufacturer](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.682.1 Detailed Description

This structure is used to store device manufacturer information.

Parameters

<i>manufacturer</i> [O-UT]	<ul style="list-style-type: none"> • NULL terminated string • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.682.2 Field Documentation

8.682.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

8.682.2.2 swi_uint256_t unpack_dms_GetManufacturer_t::ParamPresenceMask

8.682.2.3 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

8.683 unpack_dms_GetModelID_t Struct Reference

Data Fields

- char [modelid](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.683.1 Detailed Description

This structure used to store unpack_dms_GetModelID parameters

Parameters

<i>modelid</i>	<ul style="list-style-type: none"> - Device model id. • NULL-terminated String. • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.683.2 Field Documentation

8.683.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.683.2.2 swi_uint256_t unpack_dms_GetModelID_t::ParamPresenceMask

8.683.2.3 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.684 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

- uint16_t [source](#)
- uint64_t [timestamp](#)
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.684.1 Detailed Description

This structure used to store unpack_dms_GetNetworkTime parameters

Parameters

<i>source</i>	<ul style="list-style-type: none"> • Source of timestamp 0 - 32 kHz device clock 1 - CDMA network 2 - cdma2000 1xEV-DO network • Bit to check in ParamPresenceMask - 1
<i>timestamp</i>	<ul style="list-style-type: none"> • Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980) • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

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.

8.684.2 Field Documentation

8.684.2.1 `swi_uint256_t unpack_dms_GetNetworkTime_t::ParamPresenceMask`

8.684.2.2 `uint16_t unpack_dms_GetNetworkTime_t::source`

8.684.2.3 `uint64_t unpack_dms_GetNetworkTime_t::timestamp`

8.684.2.4 `uint16_t unpack_dms_GetNetworkTime_t::Tlvresult`

8.685 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](#)

8.685.1 Detailed Description

This structure used to store `unpack_dms_GetNetworkTimeV2` parameters

Parameters

<i>source</i>	<ul style="list-style-type: none"> • Source of timestamp 0 - 32 kHz device clock 1 - CDMA network 2 - cdma2000 1xEV-DO network • Bit to check in ParamPresenceMask - 1
<i>timestamp</i>	<ul style="list-style-type: none"> • Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980) • Bit to check in ParamPresenceMask - 1
<i>pSysTime</i>	<ul style="list-style-type: none"> • Count of system time in ms that have elapsed from the start of GPS time (Jan 6, 1980) • Bit to check in ParamPresenceMask - 16
<i>pUsrTime</i>	<ul style="list-style-type: none"> • Count of user time in ms that have elapsed from the start of GPS time (Jan 6, 1980) • Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

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.

8.685.2 Field Documentation

8.685.2.1 `swi_uint256_t unpack_dms_GetNetworkTimeV2_t::ParamPresenceMask`

8.685.2.2 `uint64_t* unpack_dms_GetNetworkTimeV2_t::pSysTime`

8.685.2.3 `uint64_t* unpack_dms_GetNetworkTimeV2_t::pUsrTime`

8.685.2.4 `uint16_t unpack_dms_GetNetworkTimeV2_t::source`

8.685.2.5 `uint64_t unpack_dms_GetNetworkTimeV2_t::timestamp`

8.685.2.6 `uint16_t unpack_dms_GetNetworkTimeV2_t::Tlvresult`

8.686 `unpack_dms_GetOfflineReason_t` Struct Reference

Data Fields

- `uint32_t * pReasonMask`
- `uint32_t * pbPlatform`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.686.1 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"> Optional parameter Bitmask of offline reasons <ul style="list-style-type: none"> 0x00000001 - Host image configuration issue 0x00000002 - PRI image configuration issue 0x00000004 - PRI version incompatible 0x00000008 - PRI copy issue All others - Reserved Bit to check in ParamPresenceMask - 16
-------------------------	--

<i>pbPlatform[OUT]</i>	<ul style="list-style-type: none"> • Optional parameter • Is the device offline due to a platform restriction? <ul style="list-style-type: none"> – 0 - No – 1 - Yes • Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.686.2 Field Documentation

8.686.2.1 `swi_uint256_t unpack_dms_GetOfflineReason_t::ParamPresenceMask`

8.686.2.2 `uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform`

8.686.2.3 `uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask`

8.686.2.4 `uint16_t unpack_dms_GetOfflineReason_t::Tlvresult`

8.687 unpack_dms_GetPower_t Struct Reference

Data Fields

- `uint32_t` [OperationMode](#)
- `uint32_t` [OfflineReason](#)
- `uint32_t` [HardwareControlledMode](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.687.1 Detailed Description

This structure used to store `unpack_dms_GetPower` parameters

Parameters

<i>OperationMode</i>	<ul style="list-style-type: none"> - operating mode. <ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 1
----------------------	---

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"> • 0x0001 - Host image misconfiguration. • 0x0002 - PRI image misconfiguration. • 0x0004 - PRI version incompatible. • 0x0008 - Device memory is full, cannot copy PRI information. • Bit to check in ParamPresenceMask - 16
<i>Hardware-ControlledMode</i>	- hardware restricted mode. <ul style="list-style-type: none"> • 0x00 - FALSE. • 0x01 - TRUE. • Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.687.2 Field Documentation

8.687.2.1 `uint32_t unpack_dms_GetPower_t::HardwareControlledMode`

8.687.2.2 `uint32_t unpack_dms_GetPower_t::OfflineReason`

8.687.2.3 `uint32_t unpack_dms_GetPower_t::OperationMode`

8.687.2.4 `swi_uint256_t unpack_dms_GetPower_t::ParamPresenceMask`

8.687.2.5 `uint16_t unpack_dms_GetPower_t::Tlvresult`

8.688 unpack_dms_GetPRLVersion_t Struct Reference**Data Fields**

- `uint8_t u8PRLPreference`
- `uint16_t u16PRLVersion`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.688.1 Detailed Description

This structure used to store `unpack_dms_GetPRLVersion` parameters

Parameters

<i>u8PRL-Preference</i>	<ul style="list-style-type: none"> • PRL Preference <ul style="list-style-type: none"> – 0 - Unset – 1 - Set • Bit to check in ParamPresenceMask - 17
<i>u16PRLVersion</i>	<ul style="list-style-type: none"> • PRL version of device. • Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.688.2 Field Documentation

8.688.2.1 swi_uint256_t unpack_dms_GetPRLVersion_t::ParamPresenceMask

8.688.2.2 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.688.2.3 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.688.2.4 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.689 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](#)

8.689.1 Detailed Description

This structure used to store unpack_dms_GetSerialNumbers parameters

Parameters

<i>esn</i>	<ul style="list-style-type: none"> - Electronic Serial Number of the device. <ul style="list-style-type: none"> • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed. • Bit to check in ParamPresenceMask - 16
<i>imei_no</i>	<ul style="list-style-type: none"> - International Mobile Equipment Identity of the device. <ul style="list-style-type: none"> • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed. • Bit to check in ParamPresenceMask - 17

<i>meid</i>	<ul style="list-style-type: none"> - Mobile Equipment Identifier of the device. • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed. • Bit to check in ParamPresenceMask - 18
<i>imeisv_svn</i>	<ul style="list-style-type: none"> - NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.imei software version revision. • Bit to check in ParamPresenceMask - 19
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.689.2 Field Documentation

8.689.2.1 `char unpack_dms_GetSerialNumbers_t::esn[255]`

8.689.2.2 `char unpack_dms_GetSerialNumbers_t::imei_no[255]`

8.689.2.3 `char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]`

8.689.2.4 `char unpack_dms_GetSerialNumbers_t::meid[255]`

8.689.2.5 `swi_uint256_t unpack_dms_GetSerialNumbers_t::ParamPresenceMask`

8.690 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`

8.690.1 Detailed Description

This structure is used to store retrieved USB Composition

Parameters

<i>USBComp</i>	<ul style="list-style-type: none"> • Current USB Composition(optional parameter) • Values: <ul style="list-style-type: none"> – 0..5 - Reserved (non-QMI) – 6 - DM NMEA AT QMI – 7 - DM NMEA AT QMI1 QMI2 QMI3 – 8 - DM NMEA AT MBIM – 9 - MBIM – 10 - NMEA MBIM – 11 - DM MBIM – 12 - DM NMEA MBIM 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces – 13 - 6 for QMI, 8 for MBIM – 14 - 6 for QMI, 9 for MBIM – 15 - 6 for QMI, 10 for MBIM – 16 - 6 for QMI, 11 for MBIM – 17 - 6 for QMI, 12 for MBIM – 18 - 7 for QMI, 8 for MBIM – 19 - 7 for QMI, 9 for MBIM – 20 - 7 for QMI, 10 for MBIM – 21 - 7 for QMI, 11 for MBIM – 22 - 7 for QMI, 12 for MBIM • Bit to check in ParamPresenceMask - 16
<i>NumSupUSB-Comps</i>	<ul style="list-style-type: none"> • Number of supported USB compositions in the parameter to follow • Range - 0-255 • Bit to check in ParamPresenceMask - 17

<i>SupUSBComps</i>	<ul style="list-style-type: none"> • Optional parameter • List of supported USB compositions(1 Byte each - Max 255) • Total length is defined by pNumSupUSBComps parameter • Values: <ul style="list-style-type: none"> – 0..5 - Reserved (non-QMI) – 6 - DM NMEA AT QMI – 7 - DM NMEA AT QMI1 QMI2 QMI3 – 8 - DM NMEA AT MBIM – 9 - MBIM – 10 - NMEA MBIM – 11 - DM MBIM – 12 - DM NMEA MBIM – 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces – 13 - 6 for QMI, 8 for MBIM – 14 - 6 for QMI, 9 for MBIM – 15 - 6 for QMI, 10 for MBIM – 16 - 6 for QMI, 11 for MBIM – 17 - 6 for QMI, 12 for MBIM – 18 - 7 for QMI, 8 for MBIM – 19 - 7 for QMI, 9 for MBIM – 20 - 7 for QMI, 10 for MBIM – 21 - 7 for QMI, 11 for MBIM – 22 - 7 for QMI, 12 for MBIM • Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.690.2 Field Documentation

8.690.2.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.690.2.2 swi_uint256_t unpack_dms_GetUSBComp_t::ParamPresenceMask

8.690.2.3 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps[255]

8.690.2.4 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.690.2.5 uint8_t unpack_dms_GetUSBComp_t::USBComp

8.691 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](#)

8.691.1 Detailed Description

Structure to store the voice number in use by the device

Parameters

<i>voiceNumber-Size</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that the voice number array can contain. • Bit to check in ParamPresenceMask - 1
<i>VoiceNumber</i>	<ul style="list-style-type: none"> • Voice number string: MDN or MS ISDN • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>minSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that the MIN array can contain. • Bit to check in ParamPresenceMask - 16
<i>MIN</i>	<ul style="list-style-type: none"> • Optional Parameter • MIN string: Empty string returned when MIN is not supported/ programmed. • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.691.2 Field Documentation

8.691.2.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.691.2.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.691.2.3 swi_uint256_t unpack_dms_GetVoiceNumber_t::ParamPresenceMask

8.691.2.4 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.691.2.5 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.691.2.6 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.692 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](#)

8.692.1 Detailed Description

DMS Event Report indication structure

Parameters

<i>EnableState</i>	<ul style="list-style-type: none"> • See dms_PSMEnableStateIndTlv • Bit to check in ParamPresenceMask - 16
<i>ActiveTimer</i>	<ul style="list-style-type: none"> • See dms_PSMActiveTimerIndTlv • Bit to check in ParamPresenceMask - 17
<i>PeriodicUpdate-Timer</i>	<ul style="list-style-type: none"> • See dms_PSMPeriodicUpdateTimerIndTlv • Bit to check in ParamPresenceMask - 18
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.692.2 Field Documentation

8.692.2.1 [dms_PSMActiveTimerIndTlv](#) unpack_dms_PSMCfgChange_ind_t::ActiveTimer

8.692.2.2 [dms_PSMEnableStateIndTlv](#) unpack_dms_PSMCfgChange_ind_t::EnableState

8.692.2.3 [swi_uint256_t](#) unpack_dms_PSMCfgChange_ind_t::ParamPresenceMask

8.692.2.4 [dms_PSMPeriodicUpdateTimerIndTlv](#) unpack_dms_PSMCfgChange_ind_t::PeriodicUpdateTimer

8.692.2.5 [uint16_t](#) unpack_dms_PSMCfgChange_ind_t::Tlvresult

8.693 unpack_dms_ResetToFactoryDefaults_t Struct Reference

Data Fields

- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.693.1 Detailed Description

This structure contains reset to factory default unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">bitmask representation to indicate valid parameters.

8.693.2 Field Documentation

8.693.2.1 `swi_uint256_t unpack_dms_ResetToFactoryDefaults_t::ParamPresenceMask`

8.693.2.2 `uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult`

8.694 unpack_dms_SetActivationStatusCallback_t Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.694.1 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">Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">bitmask representation to indicate valid parameters.

8.694.2 Field Documentation

8.694.2.1 `swi_uint256_t unpack_dms_SetActivationStatusCallback_t::ParamPresenceMask`

8.694.2.2 `uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult`

8.695 unpack_dms_SetCrashAction_t Struct Reference

Data Fields

- `uint8_t notused`
- `swi_uint256_t ParamPresenceMask`

8.695.1 Detailed Description

This structure is used to store `unpack_dms_SetCrashAction` parameters

Parameters

<i>notused</i>	<ul style="list-style-type: none"> unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.695.2 Field Documentation

8.695.2.1 `uint8_t unpack_dms_SetCrashAction_t::notused`

8.695.2.2 `swi_uint256_t unpack_dms_SetCrashAction_t::ParamPresenceMask`

8.696 `unpack_dms_SetCustFeature_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.696.1 Detailed Description

This structure is used to store `unpack_dms_SetCustFeature` parameters

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.696.2 Field Documentation

8.696.2.1 `swi_uint256_t unpack_dms_SetCustFeature_t::ParamPresenceMask`

8.696.2.2 `uint16_t unpack_dms_SetCustFeature_t::Tlvresult`

8.697 `unpack_dms_SetCustFeaturesV2_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.697.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.697.2 Field Documentation

8.697.2.1 `swi_uint256_t unpack_dms_SetCustFeaturesV2_t::ParamPresenceMask`

8.697.2.2 `uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult`

8.698 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

- [dms_ActivationStatusTlv](#) ActivationStatusTlv
- [dms_OperatingModeTlv](#) OperatingModeTlv
- `uint16_t Tlvresult`
- [swi_uint256_t](#) ParamPresenceMask

8.698.1 Detailed Description

DMS Event Report indication structure

Parameters

<i>ActivationStatus-Tlv</i>	<ul style="list-style-type: none"> See dms_ActivationStatusTlv Bit to check in ParamPresenceMask - 19
<i>OperatingMode-Tlv</i>	<ul style="list-style-type: none"> See dms_OperatingModeTlv Bit to check in ParamPresenceMask - 20
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.698.2 Field Documentation

8.698.2.1 `dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv`

8.698.2.2 `dms_OperatingModeTlv` `unpack_dms_SetEventReport_ind_t::OperatingModeTlv`

8.698.2.3 `swi_uint256_t` `unpack_dms_SetEventReport_ind_t::ParamPresenceMask`

8.698.2.4 `uint16_t` `unpack_dms_SetEventReport_ind_t::Tlvresult`

8.699 `unpack_dms_SetEventReport_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.699.1 Detailed Description

This structure is used to store `unpack_dms_SetEventReport` parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.699.2 Field Documentation

8.699.2.1 `swi_uint256_t` `unpack_dms_SetEventReport_t::ParamPresenceMask`

8.699.2.2 `uint16_t` `unpack_dms_SetEventReport_t::Tlvresult`

8.700 `unpack_dms_SetFirmwarePreference_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.700.1 Detailed Description

This structure is used to store `unpack_dms_SetFirmwarePreference` parameters

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.700.2 Field Documentation

8.700.2.1 `swi_uint256_t` `unpack_dms_SetFirmwarePreference_t::ParamPresenceMask`

8.700.2.2 `uint16_t` `unpack_dms_SetFirmwarePreference_t::Tlvresult`

8.701 unpack_dms_SetIndicationRegister_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- [swi_uint256_t](#) `ParamPresenceMask`

8.701.1 Detailed Description

This structure is used to store `unpack_dms_SetIndicationRegister` parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack Tlv result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.701.2 Field Documentation

8.701.2.1 `swi_uint256_t` `unpack_dms_SetIndicationRegister_t::ParamPresenceMask`

8.701.2.2 `uint16_t` `unpack_dms_SetIndicationRegister_t::Tlvresult`

8.702 unpack_dms_SetPower_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- [swi_uint256_t](#) `ParamPresenceMask`

8.702.1 Detailed Description

Structure to store unpack the operating mode of the device.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.702.2 Field Documentation

8.702.2.1 `swi_uint256_t unpack_dms_SetPower_t::ParamPresenceMask`

8.702.2.2 `uint16_t unpack_dms_SetPower_t::Tlvresult`

8.703 `unpack_dms_SetUSBComp_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.703.1 Detailed Description

This structure is used to store unpack USB composition information

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.703.2 Field Documentation

8.703.2.1 `swi_uint256_t unpack_dms_SetUSBComp_t::ParamPresenceMask`

8.703.2.2 `uint16_t unpack_dms_SetUSBComp_t::Tlvresult`

8.704 `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`

8.704.1 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"> Selected operating mode. Values <ul style="list-style-type: none"> 0 - Online 1 - Low power 2 - Factory Test mode 3 - Offline 4 - Resetting 5 - Shutting down 6 - Persistent low power 8 - Conducting network test for GSM/WCDMA Bit to check in ParamPresenceMask - 1
<i>has_LpmFlag</i>	LPM Flag Availability. <ul style="list-style-type: none"> 0 : Unavailable. 1 : Available.

- Bit to check in ParamPresenceMask - **16**

Parameters

<i>LpmFlag[-Optional]</i>	<ul style="list-style-type: none"> LPM Force Flags Bitmask <ul style="list-style-type: none"> bit0 - Tracks the state of W_DISABLE TLV, indicating state of the W_DISABLE switch. 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. bit2 - Set when the device temperature is outside the valid operating range. Cleared if the temperature returns to the normal range. bit3 - Set when the device voltage is outside the valid operating range. Cleared if the voltage returns to the normal range. bit4 - Set on power up when BIOS locking is enabled. Cleared when the host has disabled the BIOS lock. bit5 - Set if the current device configuration does not match the GOBI image preference. bit6-31 - Additional LPM causes may be added to future products. Bit to check in ParamPresenceMask - 16
<i>has_Wdisable</i>	W-Disable Availability. <ul style="list-style-type: none"> 0 : Unavailable. 1 : Available.

- Bit to check in ParamPresenceMask - **17**

Parameters

<i>Wdisable[-Optional]</i>	<ul style="list-style-type: none"> • W_DISABLE • Values <ul style="list-style-type: none"> – 0 - Switch set to ON position – 1 - Switch set to OFF position • Bit to check in ParamPresenceMask - 17
<i>has_PowerOff-Mode</i>	Power off mode Availability. <ul style="list-style-type: none"> • 0 : Unavailable. • 1 : Available.

- Bit to check in ParamPresenceMask - **18**

Parameters

<i>PowerOffMode[-Optional]</i>	<ul style="list-style-type: none"> • Power-off Mode • Action taken when W_DISABLE is switched to the OFF position <ul style="list-style-type: none"> – 0 - LPM – 1 - Shutdown – 2 - Ignore • Bit to check in ParamPresenceMask - 18
<i>has_Persistent-Lpm</i>	Persistent LPM Availability. <ul style="list-style-type: none"> • 0 : Unavailable. • 1 : Available.

- Bit to check in ParamPresenceMask - **19**

Parameters

<i>PersistentLpm[-Optional]</i>	<ul style="list-style-type: none"> • LPM Persistence • Values <ul style="list-style-type: none"> – 0 - Non-persistent LPM – 1 - Persistent LPM • Bit to check in ParamPresenceMask - 19
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.704.2 Field Documentation

8.704.2.1 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_LpmFlag

8.704.2.2 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_PersistentLpm

8.704.2.3 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_PowerOffMode

8.704.2.4 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_Wdisable

8.704.2.5 uint32_t unpack_dms_SLQSDmsSwiGetPCInfo_t::LpmFlag

8.704.2.6 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::opMode

8.704.2.7 swi_uint256_t unpack_dms_SLQSDmsSwiGetPCInfo_t::ParamPresenceMask

8.704.2.8 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::PersistentLpm

8.704.2.9 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::PowerOffMode

8.704.2.10 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::Wdisable

8.705 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [source](#)
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.705.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

<i>type[OUT]</i>	<ul style="list-style-type: none"> • type of reset or power down, possible values listed below: <ul style="list-style-type: none"> – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down • Bit to check in ParamPresenceMask - 1
<i>source[OUT]</i>	<ul style="list-style-type: none"> • entity which initiated the reset or power down, possible values listed below: <ul style="list-style-type: none"> – 0 - unknown – 1 - user requested (AT!RESET, AT!BOOTHOLD, FW/PRI download - including host-initiated image switching) – 2 - hardware switch (W_DISABLE) – 3 - temperature critical – 4 - voltage critical – 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset) – 6 - LWM2M (Light Weight M2M client (internal process for LWM2M)) – 7 - OMA-DM – 8 - FOTA

- Bit to check in ParamPresenceMask - 1

Parameters

<i>Tlvresult[OUT]</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.705.2 Field Documentation

8.705.2.1 `swi_uint256_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::ParamPresenceMask`

8.705.2.2 `uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source`

8.705.2.3 `uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult`

8.705.2.4 `uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type`

8.706 `unpack_dms_SLQSDmsSwiGetResetInfo_t` Struct Reference

Data Fields

- `uint8_t type`
- `uint8_t source`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.706.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

<i>type[OUT]</i>	<ul style="list-style-type: none"> • type of reset or power down, possible values listed below: <ul style="list-style-type: none"> – 0 - unknown – 1 - warm – 2 - hard – 3 - crash – 4 - power down • Bit to check in ParamPresenceMask - 1
------------------	---

<i>source[OUT]</i>	<ul style="list-style-type: none"> entity which initiated the reset or power down, possible values listed below: <ul style="list-style-type: none"> 0 - unknown 1 - user requested (AT!RESET, AT!BOOTHOLD, FW/PRI download - including host-initiated image switching) 2 - hardware switch (W_DISABLE) 3 - temperature critical 4 - voltage critical 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset) 6 - LWM2M (Light Weight M2M client (internal process for LWM2M)) 7 - OMA-DM 8 - FOTA
--------------------	--

- Bit to check in ParamPresenceMask - 1

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.706.2 Field Documentation

8.706.2.1 swi_uint256_t unpack_dms_SLQSDmsSwiGetResetInfo_t::ParamPresenceMask

8.706.2.2 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.706.2.3 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.706.2.4 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.707 unpack_dms_SLQSDmsSwiGetUimSelection_t Struct Reference

Data Fields

- uint8_t [uimSelect](#)
- [dms_UimAutoSwitchActSlotTlv](#) * [pUimAutoSwitchActSlot](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.707.1 Detailed Description

Structure contains the TLV required to Get currently selected active UIM slot.

Parameters

<i>uimSelect</i>	<ul style="list-style-type: none"> • Active selected UIM slot • Values <ul style="list-style-type: none"> – 0 - slot 1 (e.g. external SIM) – 1 - slot 2 (e.g. embedded SIM) – 2 - remote SIM (if supported) – 3 - SIM auto-switch activated
------------------	--

- Bit to check in ParamPresenceMask - 1

Parameters

<i>pUimAutoSwitchActSlot</i>	<ul style="list-style-type: none"> • UIM Auto Switch Active slot • see dms_UimAutoSwitchActSlotTlv for more information • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

: None

8.707.2 Field Documentation

8.707.2.1 `swi_uint256_t unpack_dms_SLQSDmsSwiGetUimSelection_t::ParamPresenceMask`

8.707.2.2 `dms_UimAutoSwitchActSlotTlv* unpack_dms_SLQSDmsSwiGetUimSelection_t::pUimAutoSwitchActSlot`

8.707.2.3 `uint8_t unpack_dms_SLQSDmsSwiGetUimSelection_t::uimSelect`

8.708 `unpack_dms_SLQSDmsSwiIndicationRegister_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.708.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.708.2 Field Documentation

8.708.2.1 `swi_uint256_t` `unpack_dms_SLQSDmsSwilIndicationRegister_t::ParamPresenceMask`

8.708.2.2 `uint16_t` `unpack_dms_SLQSDmsSwilIndicationRegister_t::Tlvresult`

8.709 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`

8.709.1 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"> • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19 • Bit 60 - WCDMA Japan 850 band • Bit 61 - WCDMA 1500 band • Bits 62 and 63 - Reserved • Bit to check in ParamPresenceMask - 1
----------------------------	--

<i>is_LteBand-Capability_Available</i> [OUT]	<div>LTE Band Capabilities Availability.<ul style="list-style-type: none">• 0 : Unavailable.• 1 : Available.• Bit to check in ParamPresenceMask - 16</div>
--	---

<i>LteBand-Capability[OUT]</i>	<p>Bitmask of LTE bands supported by the device</p> <ul style="list-style-type: none"> • Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170 • Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990 • Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880 • Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155 • Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894 • Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885 • Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690 • Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960 • Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9 • Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170 • Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9 • Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746 • Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756 • Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768 • Bits 14 and 15 - Reserved • Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746 • Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875 • Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890 • Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821 • Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9 • Bit 21 - Reserved • Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200 • Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559 • Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995 • Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894 • Bit 26 - Reserved • Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803 • Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728 • Bit 29 - LTE EUTRAN Band 30 UL: 2350-2360; DL: 2305-2315 • Bit 30 - Reserved • Bit 31 - LTE EUTRAN Band 32 DL: 9920-10359 • Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920 • Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025 • Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910 • Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990 • Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930 • Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620 • Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920 • Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400 • Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690 • Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600 • Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800 • Bits 43 through 64 - Reserved • Bit to check in ParamPresenceMask - 16
--------------------------------	---

<i>is_TdsBand-Capability_Available</i> [OUT]	TDS Band Capabilities Availability. <ul style="list-style-type: none"> • 0 : Unavailable. • 1 : Available. • Bit to check in ParamPresenceMask - 17
<i>TdsBand-Capability</i> [OUT]	Bitmask of TDS bands supported by the device. <ul style="list-style-type: none"> • Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz • Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz • Bit 2 - TDS Band C 1910 to 1930 MHz • Bit 3 - TDS Band D 2570 to 2620 MHz • Bit 4 - TDS Band E 2300 to 2400 MHz • Bit 5 - TDS Band F 1880 to 1920 MHz • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.709.2 Field Documentation

8.709.2.1 `uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability`

8.709.2.2 `int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available`

8.709.2.3 `int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available`

8.709.2.4 `uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability`

8.709.2.5 `swi_uint256_t unpack_dms_SLQSGetBandCapability_t::ParamPresenceMask`

8.709.2.6 `uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability`

8.710 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`

8.710.1 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"> • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19 • Bit 60 - WCDMA Japan 850 band • Bit 61 - WCDMA 1500 band • Bits 62 and 63 - Reserved • Bit to check in ParamPresenceMask - 1
----------------------------	---

<i>is_LteBand-Capability_Available</i> [OUT]	<div>LTE Band Capabilities Availability.<ul style="list-style-type: none">• 0 : Unavailable.• 1 : Available.• Bit to check in ParamPresenceMask - 16</div>
--	---

<i>LteBand-Capability</i> [OUT]	<p>Bitmask of LTE bands supported by the device</p> <ul style="list-style-type: none"> • Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170 • Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990 • Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880 • Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155 • Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894 • Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885 • Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690 • Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960 • Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9 • Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170 • Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9 • Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746 • Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756 • Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768 • Bits 14 and 15 - Reserved • Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746 • Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875 • Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890 • Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821 • Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9 • Bit 21 - Reserved • Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200 • Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559 • Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995 • Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894 • Bit 26 - Reserved • Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803 • Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728 • Bit 29 - LTE EUTRAN Band 30 UL: 2350-2360; DL: 2305-2315 • Bit 30 - Reserved • Bit 31 - LTE EUTRAN Band 32 DL: 9920-10359 • Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920 • Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025 • Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910 • Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990 • Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930 • Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620 • Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920 • Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400 • Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690 • Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600 • Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800 • Bits 43 through 64 - Reserved • Bit to check in ParamPresenceMask - 16
---------------------------------	---

<i>is_TdsBand-Capability_Available[OUT]</i>	TDS Band Capabilities Availability. <ul style="list-style-type: none"> • 0 : Unavailable. • 1 : Available. • Bit to check in ParamPresenceMask - 17
<i>TdsBand-Capability[OUT]</i>	Bitmask of TDS bands supported by the device. <ul style="list-style-type: none"> • Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz • Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz • Bit 2 - TDS Band C 1910 to 1930 MHz • Bit 3 - TDS Band D 2570 to 2620 MHz • Bit 4 - TDS Band E 2300 to 2400 MHz • Bit 5 - TDS Band F 1880 to 1920 MHz • Bit to check in ParamPresenceMask - 17
<i>LteBands-Support[OUT]</i>	<ul style="list-style-type: none"> • See dms_LteBandsSupport for more information
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.710.2 Field Documentation

8.710.2.1 `uint64_t unpack_dms_SLQSGetBandCapabilityExt_t::bandCapability`

8.710.2.2 `int unpack_dms_SLQSGetBandCapabilityExt_t::is_LteBandCapability_Available`

8.710.2.3 `int unpack_dms_SLQSGetBandCapabilityExt_t::is_TdsBandCapability_Available`

8.710.2.4 `uint64_t unpack_dms_SLQSGetBandCapabilityExt_t::LteBandCapability`

8.710.2.5 `dms_LteBandsSupport unpack_dms_SLQSGetBandCapabilityExt_t::LteBandsSupport`

8.710.2.6 `swi_uint256_t unpack_dms_SLQSGetBandCapabilityExt_t::ParamPresenceMask`

8.710.2.7 `uint64_t unpack_dms_SLQSGetBandCapabilityExt_t::TdsBandCapability`

8.711 unpack_dms_SLQSGetERIFile_t Struct Reference

Data Fields

- [eriDataparams eriFile](#)
- `uint16_t Tlvresult`
- [swi_uint256_t ParamPresenceMask](#)

8.711.1 Detailed Description

This structure contains Get ERI file unpack

Parameters

<i>eriFile</i>	<ul style="list-style-type: none"> • Pointer to structure ERIFileparams • See eriDataparams for more information • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.711.2 Field Documentation

8.711.2.1 [eriDataparams](#) [unpack_dms_SLQSGetERIFile_t::eriFile](#)8.711.2.2 [swi_uint256_t](#) [unpack_dms_SLQSGetERIFile_t::ParamPresenceMask](#)8.711.2.3 [uint16_t](#) [unpack_dms_SLQSGetERIFile_t::Tlvresult](#)8.712 [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](#)

8.712.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

Parameters

<i>pPsmEnableState</i>	[Optional] <ul style="list-style-type: none"> • See dms_PSMEnableStateTlv for more information • Bit to check in ParamPresenceMask - 16
<i>pDurationThreshold</i>	[Optional] <ul style="list-style-type: none"> • See dms_PSMDurationThresholdTlv for more information • Bit to check in ParamPresenceMask - 17
<i>pDurationDueToOOS</i>	[Optional] <ul style="list-style-type: none"> • See dms_PSMDurationDueToOOSTlv for more information • Bit to check in ParamPresenceMask - 18

<i>pRandomization-Window</i>	[Optional] <ul style="list-style-type: none"> See dms_PSMRandomizationWindowTlv for more information Bit to check in ParamPresenceMask - 19
<i>pActiveTimer</i>	[Optional] <ul style="list-style-type: none"> See dms_PSMActiveTimerTlv for more information Bit to check in ParamPresenceMask - 20
<i>pPeriodic-UpdateTimer</i>	[Optional] <ul style="list-style-type: none"> See dms_PSMPeriodicUpdateTimerTlv for more information Bit to check in ParamPresenceMask - 21
<i>pEarlyWakeup-Time</i>	[Optional] <ul style="list-style-type: none"> See dms_PSMEarlyWakeupTimeTlv for more information Bit to check in ParamPresenceMask - 22
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.712.2 Field Documentation

8.712.2.1 `dms_PSMActiveTimerTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pActiveTimer`

8.712.2.2 `swi_uint256_t` `unpack_dms_SLQSSetPowerSaveModeConfig_t::ParamPresenceMask`

8.712.2.3 `dms_PSMDurationDueToOOSTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pDurationDueToOOS`

8.712.2.4 `dms_PSMDurationThresholdTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pDurationThreshold`

8.712.2.5 `dms_PSMEarlyWakeupTimeTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pEarlyWakeupTime`

8.712.2.6 `dms_PSMPeriodicUpdateTimerTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pPeriodicUpdateTimer`

8.712.2.7 `dms_PSMEnableStateTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pPsmEnableState`

8.712.2.8 `dms_PSMRandomizationWindowTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pRandomization-Window`

8.713 unpack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.713.1 Detailed Description

This structure is used to store `unpack_dms_SLQSSetPowerSaveModeConfig` parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.713.2 Field Documentation

8.713.2.1 `swi_uint256_t` `unpack_dms_SLQSSetPowerSaveModeConfig_t::ParamPresenceMask`8.713.2.2 `uint16_t` `unpack_dms_SLQSSetPowerSaveModeConfig_t::Tlvresult`8.714 `unpack_dms_SLQSSwiClearDyingGaspStatistics_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.714.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.714.2 Field Documentation

8.714.2.1 `swi_uint256_t` `unpack_dms_SLQSSwiClearDyingGaspStatistics_t::ParamPresenceMask`8.714.2.2 `uint16_t` `unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult`8.715 `unpack_dms_SLQSSwiGetCrashInfo_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- [crashInfoParams](#) `crashInfoParam`
- `swi_uint256_t` [ParamPresenceMask](#)

8.715.1 Detailed Description

This structure contains SWI get crash information unpack information

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>crashInfoParam[-OUT]</i>	<ul style="list-style-type: none"> • See crashInfoParams • crashStatus <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 1 • crashInfo <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.715.2 Field Documentation

8.715.2.1 [crashInfoParams](#) unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam8.715.2.2 [swi_uint256_t](#) unpack_dms_SLQSSwiGetCrashInfo_t::ParamPresenceMask8.715.2.3 [uint16_t](#) unpack_dms_SLQSSwiGetCrashInfo_t::Tlvresult

8.716 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

- [packgetDyingGaspCfg](#) * [pGetDyingGaspCfg](#)
- [uint16_t](#) [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.716.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

<i>pGetDyingGasp-Cfg</i>	<ul style="list-style-type: none"> • See packgetDyingGaspCfg • pDestSMSNum. <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 16 • pDestSMSContent. <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.716.2 Field Documentation

8.716.2.1 `swi_uint256_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::ParamPresenceMask`

8.716.2.2 `packgetDyingGaspCfg* unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg`

8.716.2.3 `uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult`

8.717 `unpack_dms_SLQSSwiGetDyingGaspStatistics_t` Struct Reference

Data Fields

- [packgetDyingGaspStatistics](#) * [pGetDyingGaspStatistics](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.717.1 Detailed Description

This structure contains Get Dying GASP Statistics unpack.

Parameters

<i>pGetDyingGaspStatistics</i>	<ul style="list-style-type: none"> • See packgetDyingGaspStatistics • <code>pTimeStamp</code>. <ul style="list-style-type: none"> – Bit to check in <code>ParamPresenceMask</code> - 16 • <code>pSMSAttemptedFlag</code>. <ul style="list-style-type: none"> – Bit to check in <code>ParamPresenceMask</code> - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result

8.717.2 Field Documentation

8.717.2.1 `swi_uint256_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::ParamPresenceMask`

8.717.2.2 `packgetDyingGaspStatistics* unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics`

8.717.2.3 `uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

8.718 `unpack_dms_SLQSSwiGetFirmwareCurr_t` Struct Reference

Data Fields

- `uint8_t` [numEntries](#)
- `image_info_t` * [pCurrImgInfo](#)
- `char` [priver](#) [16]
- `char` [pkgver](#) [16]
- `char` [fwvers](#) [16]
- `char` [carrier](#) [16]
- `swi_uint256_t` [ParamPresenceMask](#)

8.718.1 Detailed Description

This structure is used to store unpack_dms_SLQSSwiGetFirmwareCurr parameters

Parameters

<i>numEntries</i> [IN/-OUT]	<ul style="list-style-type: none"> Number of entries in the image list to follow The size of the list pCurrImgInfo must be specified when calling the API <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 1
<i>pCurrImgInfo</i> [OUT]	<ul style="list-style-type: none"> Currently Active Image List <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 1
<i>priver</i> [OUT]	<ul style="list-style-type: none"> PRI version of the currently running firmware Bit to check in ParamPresenceMask - 1
<i>pkgver</i> [OUT]	<ul style="list-style-type: none"> Package version of the currently running firmware Bit to check in ParamPresenceMask - 1
<i>fwvers</i> [OUT]	<ul style="list-style-type: none"> firmware version of the currently running firmware Bit to check in ParamPresenceMask - 1
<i>carrier</i> [OUT]	<ul style="list-style-type: none"> Carrier string of the currently running firmware Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.718.2 Field Documentation

8.718.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.718.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.718.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.718.2.4 swi_uint256_t unpack_dms_SLQSSwiGetFirmwareCurr_t::ParamPresenceMask

8.718.2.5 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.718.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.718.2.7 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.719 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](#)

8.719.1 Detailed Description

This structure is used to store Firmware Update Status

Parameters

<i>ResCode</i>	<ul style="list-style-type: none"> • FW Update Result Code • Values: <ul style="list-style-type: none"> – 0x00000001 - Successful – 0xFFFFFFFF - Unknown (due to power off reset after firmware update) – 0x100000nn - File update errors while nn will be the exact error number: <ul style="list-style-type: none"> * 00 - General error – 0x200000nn - NVUP update errors while nn will be the exact error number: <ul style="list-style-type: none"> * 00 - General error – 0x40000nnn - FOTA update agent errors while nnn will be the exact error number: <ul style="list-style-type: none"> * 000 ~ 0FF - Insignia defined error code * 100 ~ 1FF - Sierra defined error code * See qaGobiApiTableFwDldErrorCodes.h for more detailed information – 0x800000nn - FDT/SSDP reported errors while nn will be the exact error number <ul style="list-style-type: none"> * See qaGobiApiTableFwDldErrorCodes.h for more detailed information • Bit to check in ParamPresenceMask - 16
<i>imgType</i>	<ul style="list-style-type: none"> • Optional parameter • Firmware image type that failed the update • Bit to check in ParamPresenceMask - 17
<i>refData</i>	<ul style="list-style-type: none"> • Optional parameter • Failed image reference data • This is normally the offset of the image that caused the failure • Bit to check in ParamPresenceMask - 18
<i>refString</i>	<ul style="list-style-type: none"> • Optional parameter • Failed image reference string. This is normally the partition name of the image that caused the failure if applicable. • Bit to check in ParamPresenceMask - 19

<i>logString</i>	<ul style="list-style-type: none"> Optional parameter Failed image reference string. This is normally the partition name of the image that caused the failure if applicable. Bit to check in ParamPresenceMask - 20
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.719.2 Field Documentation

8.719.2.1 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType`

8.719.2.2 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]`

8.719.2.3 `swi_uint256_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ParamPresenceMask`

8.719.2.4 `uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData`

8.719.2.5 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]`

8.719.2.6 `uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode`

8.719.2.7 `uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult`

8.720 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`

8.720.1 Detailed Description

This structure contains SWI get host device info unpack information

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Result
------------------	---

<i>manString</i> [OUT]	<ul style="list-style-type: none"> • Host Device Manufacturer Name • Null terminated ASCII String • Bit to check in ParamPresenceMask - 16
<i>modelString</i> [OUT]	<ul style="list-style-type: none"> • Host Device Model String • Null terminated ASCII String • Bit to check in ParamPresenceMask - 17
<i>swVerString</i> [OUT]	<ul style="list-style-type: none"> • Host Device Software Version String • Null terminated ASCII string • Bit to check in ParamPresenceMask - 18
<i>plasmaIDString</i> [OUT]	<ul style="list-style-type: none"> • Host Device Plasma ID String • Null terminated alphanumeric ASCII String. • Bit to check in ParamPresenceMask - 19
<i>hostID</i> [OUT]	<ul style="list-style-type: none"> • Device Host ID String • Null terminated alphanumeric ASCII String. • Bit to check in ParamPresenceMask - 20
<i>ParamPresenceMask</i> [OUT]	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.720.2 Field Documentation

8.720.2.1 `char unpack_dms_SLQSSwiGetHostDevInfo_t::hostID[255]`

8.720.2.2 `char unpack_dms_SLQSSwiGetHostDevInfo_t::manString[255]`

8.720.2.3 `char unpack_dms_SLQSSwiGetHostDevInfo_t::modelString[255]`

8.720.2.4 `swi_uint256_t unpack_dms_SLQSSwiGetHostDevInfo_t::ParamPresenceMask`

8.720.2.5 `char unpack_dms_SLQSSwiGetHostDevInfo_t::plasmaIDString[255]`

8.720.2.6 `char unpack_dms_SLQSSwiGetHostDevInfo_t::swVerString[255]`

8.720.2.7 `uint16_t unpack_dms_SLQSSwiGetHostDevInfo_t::Tlvresult`

8.721 `unpack_dms_SLQSSwiGetOSInfo_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `char nameString [255]`
- `char versionString [255]`

- [swi_uint256_t ParamPresenceMask](#)

8.721.1 Detailed Description

This structure contains SWI get host os info unpack information

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result
<i>nameString[OUT]</i>	<ul style="list-style-type: none">• Host operating system name• Bit to check in ParamPresenceMask - 16
<i>VersionString[OUT]</i>	<ul style="list-style-type: none">• Host operating system version• Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.721.2 Field Documentation

8.721.2.1 `char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]`

8.721.2.2 `swi_uint256_t unpack_dms_SLQSSwiGetOSInfo_t::ParamPresenceMask`

8.721.2.3 `uint16_t unpack_dms_SLQSSwiGetOSInfo_t::Tlvresult`

8.721.2.4 `char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]`

8.722 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `char meidString [8]`
- [swi_uint256_t ParamPresenceMask](#)

8.722.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• pack Result
<i>meidString[OUT]</i>	<ul style="list-style-type: none">• optional parameter, mobile equipment identifier• Bit to check in ParamPresenceMask - 16

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.722.2 Field Documentation

8.722.2.1 `char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]`

8.722.2.2 `swi_uint256_t unpack_dms_SLQSSwiGetSerialNoExt_t::ParamPresenceMask`

8.722.2.3 `uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult`

8.723 `unpack_dms_SLQSSwiSetDyingGaspCfg_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.723.1 Detailed Description

This structure contains set Dying GASP Config unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.723.2 Field Documentation

8.723.2.1 `swi_uint256_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::ParamPresenceMask`

8.723.2.2 `uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult`

8.724 `unpack_dms_SLQSSwiSetHostDevInfo_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.724.1 Detailed Description

This structure contains SWI set host dev info unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.724.2 Field Documentation

8.724.2.1 swi_uint256_t unpack_dms_SLQSSwiSetHostDevInfo_t::ParamPresenceMask

8.724.2.2 uint16_t unpack_dms_SLQSSwiSetHostDevInfo_t::Tlvresult

8.725 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- swi_uint256_t [ParamPresenceMask](#)

8.725.1 Detailed Description

This structure contains SWI set host OS info unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.725.2 Field Documentation

8.725.2.1 swi_uint256_t unpack_dms_SLQSSwiSetOSInfo_t::ParamPresenceMask

8.725.2.2 uint16_t unpack_dms_SLQSSwiSetOSInfo_t::Tlvresult

8.726 unpack_dms_SLQSUIMGetState_t Struct Reference

Data Fields

- uint8_t [state](#)
- uint16_t [Tlvresult](#)
- swi_uint256_t [ParamPresenceMask](#)

8.726.1 Detailed Description

This structure contains UIM get state unpack information

Parameters

<i>state[OUT]</i>	<ul style="list-style-type: none"> • UIM state • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.726.2 Field Documentation

8.726.2.1 `swi_uint256_t unpack_dms_SLQSUIMGetState_t::ParamPresenceMask`8.726.2.2 `uint8_t unpack_dms_SLQSUIMGetState_t::state`8.726.2.3 `uint16_t unpack_dms_SLQSUIMGetState_t::Tlvresult`8.727 `unpack_dms_SwiEventReportCallBack_ind_t` Struct Reference

Data Fields

- [dms_TemperatureTlv](#) TempTlv
- [dms_VoltageTlv](#) VoltTlv
- [dms_UimStatusTlv](#) UimStatusTlv
- [swi_uint256_t](#) ParamPresenceMask

8.727.1 Detailed Description

Structure used to SWI event report indication parameters.

Parameters

<i>TempTlv</i>	<ul style="list-style-type: none"> • Temperature status tlv • See dms_TemperatureTlv for more information
<i>VoltTlv</i>	<ul style="list-style-type: none"> • Voltage status tlv • see dms_VoltageTlv for more information
<i>UimStatusTlv</i>	<ul style="list-style-type: none"> • UIM status tlv • see dms_UimStatusTlv for more information
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

: None

8.727.2 Field Documentation

8.727.2.1 swi_uint256_t unpack_dms_SwiEventReportCallBack_ind_t::ParamPresenceMask

8.727.2.2 dms_TemperatureTlv unpack_dms_SwiEventReportCallBack_ind_t::TempTlv

8.727.2.3 dms_UimStatusTlv unpack_dms_SwiEventReportCallBack_ind_t::UimStatusTlv

8.727.2.4 dms_VoltageTlv unpack_dms_SwiEventReportCallBack_ind_t::VoltTlv

8.728 unpack_dms_SwiSetEventReport_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.728.1 Detailed Description

This structure is used to store unpack_dms_SwiSetEventReport parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack Tlv result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.728.2 Field Documentation

8.728.2.1 swi_uint256_t unpack_dms_SwiSetEventReport_t::ParamPresenceMask

8.728.2.2 uint16_t unpack_dms_SwiSetEventReport_t::Tlvresult

8.729 unpack_dms_SwiUimSelect_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.729.1 Detailed Description

This structure is used to store unpack_dms_SwiUimSelect parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack Tlv result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.729.2 Field Documentation

8.729.2.1 `swi_uint256_t unpack_dms_SwiUimSelect_t::ParamPresenceMask`8.729.2.2 `uint16_t unpack_dms_SwiUimSelect_t::Tlvresult`8.730 `unpack_dms_UIMGetControlKeyStatus_t` Struct Reference

Data Fields

- `uint8_t facilityState`
- `uint8_t verifyRetriesLeft`
- `uint8_t unblockRetriesLeft`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.730.1 Detailed Description

This structure contains PIN retries status

Parameters

<i>facilityState[OUT]</i>	<ul style="list-style-type: none"> Control key status <ul style="list-style-type: none"> 0 - Deactivated 1 - Activated 2 - Blocked Bit to check in ParamPresenceMask - 1
<i>verifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> Optional parameter Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. Bit to check in ParamPresenceMask - 1
<i>unblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> Optional parameter Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.730.2 Field Documentation

8.730.2.1 uint8_t unpack_dms_UIMGetControlKeyStatus_t::facilityState

8.730.2.2 swi_uint256_t unpack_dms_UIMGetControlKeyStatus_t::ParamPresenceMask

8.730.2.3 uint16_t unpack_dms_UIMGetControlKeyStatus_t::Tlvresult

8.730.2.4 uint8_t unpack_dms_UIMGetControlKeyStatus_t::unlockRetriesLeft

8.730.2.5 uint8_t unpack_dms_UIMGetControlKeyStatus_t::verifyRetriesLeft

8.731 unpack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint8_t [stringSize](#)
- uint8_t [String](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.731.1 Detailed Description

This structure is used to store unpack_dms_UIMGetICCID parameters.

Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> • Size of String. • Bit to check in ParamPresenceMask - 1
<i>String</i>	<ul style="list-style-type: none"> • ICCID String. • Maximum Length is 255 Bytes • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unack result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.731.2 Field Documentation

8.731.2.1 swi_uint256_t unpack_dms_UIMGetICCID_t::ParamPresenceMask

8.731.2.2 uint8_t unpack_dms_UIMGetICCID_t::String[255]

8.731.2.3 uint8_t unpack_dms_UIMGetICCID_t::stringSize

8.731.2.4 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.732 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](#)

8.732.1 Detailed Description

This structure contains Get PIN Status unpack information

Parameters

<i>p1Status[OUT]</i>	<ul style="list-style-type: none"> • PIN1 status(0xFFFFFFFF - Unknown) <ul style="list-style-type: none"> – 0 - PIN not initialized – 1 - PIN enabled, not verified – 2 - PIN enabled, verified – 3 - PIN disabled – 4 - PIN blocked – 5 - PIN permanently blocked • Bit to check in ParamPresenceMask - 17
<i>p1VerifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> • Optional parameter • Upon operational failure, this will indicate number of retries left of PIN1, after which PIN will be blocked. • Bit to check in ParamPresenceMask - 17
<i>p1UnblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> • Optional parameter • 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. • Bit to check in ParamPresenceMask - 17
<i>p2Status[OUT]</i>	<ul style="list-style-type: none"> • PIN2 status(0xFFFFFFFF - Unknown) <ul style="list-style-type: none"> – 0 - PIN not initialized – 1 - PIN enabled, not verified – 2 - PIN enabled, verified – 3 - PIN disabled – 4 - PIN blocked – 5 - PIN permanently blocked • Bit to check in ParamPresenceMask - 18

<i>p2VerifyRetries-Left[OUT]</i>	<ul style="list-style-type: none"> • Optional parameter • Upon operational failure, this will indicate number of retries left of PIN2, after which PIN will be blocked. • Bit to check in ParamPresenceMask - 18
<i>p2Unblock-RetriesLeft[OUT]</i>	<ul style="list-style-type: none"> • Optional parameter • 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. • Bit to check in ParamPresenceMask - 18
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.732.2 Field Documentation

8.732.2.1 `uint8_t unpack_dms_UIMGetPINStatus_t::p1Status`

8.732.2.2 `uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft`

8.732.2.3 `uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft`

8.732.2.4 `uint8_t unpack_dms_UIMGetPINStatus_t::p2Status`

8.732.2.5 `uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft`

8.732.2.6 `uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft`

8.732.2.7 `swi_uint256_t unpack_dms_UIMGetPINStatus_t::ParamPresenceMask`

8.732.2.8 `uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult`

8.733 unpack_dms_UIMSetControlKeyProtection_t Struct Reference

Data Fields

- `uint8_t verifyRetriesLeft`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.733.1 Detailed Description

This structure contains UIM Set control key protection unpack information

Parameters

<i>verifyRetries-Left</i> [OUT]	<ul style="list-style-type: none"> Optional parameter number of retries left after which the control key is blocked Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.733.2 Field Documentation

8.733.2.1 `swi_uint256_t unpack_dms_UIMSetControlKeyProtection_t::ParamPresenceMask`8.733.2.2 `uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult`8.733.2.3 `uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft`8.734 `unpack_dms_UIMSetPINProtection_t` Struct Reference

Data Fields

- `uint8_t` [verifyRetriesLeft](#)
- `uint8_t` [unblockRetriesLeft](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.734.1 Detailed Description

This structure contains PIN retries status

Parameters

<i>verifyRetries-Left</i> [OUT]	<ul style="list-style-type: none"> Optional parameter Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. Bit to check in ParamPresenceMask - 16
<i>unblockRetries-Left</i> [OUT]	<ul style="list-style-type: none"> Optional parameter Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i> [OUT]	<ul style="list-style-type: none"> Unpack Result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.734.2 Field Documentation

8.734.2.1 swi_uint256_t unpack_dms_UIMSetPINProtection_t::ParamPresenceMask

8.734.2.2 uint16_t unpack_dms_UIMSetPINProtection_t::Tlvresult

8.734.2.3 uint8_t unpack_dms_UIMSetPINProtection_t::unlockRetriesLeft

8.734.2.4 uint8_t unpack_dms_UIMSetPINProtection_t::verifyRetriesLeft

8.735 unpack_dms_UIMUnlockControlKey_t Struct Reference

Data Fields

- uint8_t [unlockRetriesLeft](#)
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.735.1 Detailed Description

This structure contains UIM Set control key protection unpack information

Parameters

<i>unlockRetriesLeft</i> [OUT]	<ul style="list-style-type: none">• Optional parameter• number of unlock retries left after which the control key is permanently blocked• Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i> [OUT]	<ul style="list-style-type: none">• Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.735.2 Field Documentation

8.735.2.1 swi_uint256_t unpack_dms_UIMUnlockControlKey_t::ParamPresenceMask

8.735.2.2 uint16_t unpack_dms_UIMUnlockControlKey_t::Tlvresult

8.735.2.3 uint8_t unpack_dms_UIMUnlockControlKey_t::unlockRetriesLeft

8.736 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint32_t [ImageListSize](#)
- [FMSPrefImageList](#) * [pImageList](#)
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.736.1 Detailed Description

This structure contains the Get Image Preference information unpack

Parameters

<i>listSize</i>	<ul style="list-style-type: none"> The number of elements in the image list
<i>pListEntries</i>	<ul style="list-style-type: none"> Array of Image entries with size provided by previous field See FMSImageElement Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack result

8.736.2 Field Documentation

8.736.2.1 `uint32_t unpack_fms_GetImagesPreference_t::ImageListSize`

8.736.2.2 `swi_uint256_t unpack_fms_GetImagesPreference_t::ParamPresenceMask`

8.736.2.3 `FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList`

8.736.2.4 `uint16_t unpack_fms_GetImagesPreference_t::Tlvresult`

8.737 unpack_fms_GetStoredImages_t Struct Reference

Data Fields

- `uint32_t` [imagelistSize](#)
- [FMSImageList](#) [imageList](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.737.1 Detailed Description

This structure contains the Get Stored Images unpack

Parameters

<i>listSize</i>	<ul style="list-style-type: none"> The number of elements in the image list
<i>imageList</i>	<ul style="list-style-type: none"> Array of Image entries with size provided by previous field See FMSImageElement Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack result

8.737.2 Field Documentation

8.737.2.1 FMSImageList unpack_fms_GetStoredImages_t::imageList

8.737.2.2 uint32_t unpack_fms_GetStoredImages_t::imagelistSize

8.737.2.3 swi_uint256_t unpack_fms_GetStoredImages_t::ParamPresenceMask

8.737.2.4 uint16_t unpack_fms_GetStoredImages_t::Tlvresult

8.738 unpack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t [ImageTypesSize](#)
- uint8_t [ImageTypes](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.738.1 Detailed Description

This structure contains the Set Images Preference unpack

Parameters

<i>ImageTypesSize</i>	<ul style="list-style-type: none"> • Image Type Size
<i>ImageTypes</i>	<ul style="list-style-type: none"> • Image Type • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result

8.738.2 Field Documentation

8.738.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

8.738.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

8.738.2.3 swi_uint256_t unpack_fms_SetImagesPreference_t::ParamPresenceMask

8.738.2.4 uint16_t unpack_fms_SetImagesPreference_t::Tlvresult

8.739 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

8.739.1 Detailed Description

This structure contains the SLQSGetIMSSMSConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> • Settings Response • Bit to check in ParamPresenceMask - 16
<i>pSMSFormat</i>	<ul style="list-style-type: none"> • SMS format <ul style="list-style-type: none"> – 0 - 3GPP – 1 - 3GPP2 • Bit to check in ParamPresenceMask - 17
<i>pSMSOverIPNwInd</i>	<ul style="list-style-type: none"> • SMS over IP Network Indication Flag <ul style="list-style-type: none"> – TRUE - Turn on mobile-originated SMS – FALSE - Turn off mobile-originated SMS • Bit to check in ParamPresenceMask - 18
<i>pPhoneCtxtURLen[IN/OUT]</i>	<ul style="list-style-type: none"> • Size in bytes assigned to the Phone context Universal Resource Identifier to follow
<i>pPhoneCtxtURI</i>	<ul style="list-style-type: none"> • Phone context universal resource identifier • Length of this string must be specified in pPhoneCtxtURLen parameter • Bit to check in ParamPresenceMask - 19

8.739.2 Field Documentation

8.739.2.1 swi_uint256_t unpack_ims_SLQSGetIMSSMSConfig_t::ParamPresenceMask

8.739.2.2 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pPhoneCtxtURI

8.739.2.3 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pPhoneCtxtURLen

8.739.2.4 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pSettingResp

8.739.2.5 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pSMSFormat

8.739.2.6 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pSMSOverIPNwInd

8.740 unpack_ims_SLQSGetIMSUserConfig_t Struct Reference

Data Fields

- uint8_t * [pSettingResp](#)
- uint8_t * [pIMSDomainLen](#)
- uint8_t * [pIMSDomain](#)
- [swi_uint256_t](#) ParamPresenceMask

8.740.1 Detailed Description

This structure contains the SLQSGetIMSUserConfig pack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> • Settings Response • Bit to check in ParamPresenceMask - 16
<i>pIMSDomainLen</i> [IN/OUT]	<ul style="list-style-type: none"> • Length of IMS Domain Name to follow
<i>pIMSDomain</i>	<ul style="list-style-type: none"> • IMS domain name • Length of this string must be specified in pIMSDomainLen parameter • Bit to check in ParamPresenceMask - 17

8.740.2 Field Documentation

8.740.2.1 [swi_uint256_t](#) unpack_ims_SLQSGetIMSUserConfig_t::ParamPresenceMask

8.740.2.2 [uint8_t*](#) unpack_ims_SLQSGetIMSUserConfig_t::pIMSDomain

8.740.2.3 [uint8_t*](#) unpack_ims_SLQSGetIMSUserConfig_t::pIMSDomainLen

8.740.2.4 [uint8_t*](#) unpack_ims_SLQSGetIMSUserConfig_t::pSettingResp

8.741 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

8.741.1 Detailed Description

This structure contains the SLQSGetIMSVoIPConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings Response. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE Bit to check in ParamPresenceMask - 16
<i>pSessionExpiry-Timer</i>	<ul style="list-style-type: none"> Session duration, in seconds Bit to check in ParamPresenceMask - 17
<i>pMinSession-ExpiryTimer</i>	<ul style="list-style-type: none"> Minimum allowed value for session expiry timer, in seconds Bit to check in ParamPresenceMask - 18
<i>pAmrWbEnable</i>	<ul style="list-style-type: none"> Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio Values: <ul style="list-style-type: none"> True - Enable False - Disable Bit to check in ParamPresenceMask - 19
<i>pScrAmrEnable</i>	<ul style="list-style-type: none"> Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB) Values: <ul style="list-style-type: none"> True - Enable False - Disable Bit to check in ParamPresenceMask - 20
<i>pScrAmrWb-Enable</i>	<ul style="list-style-type: none"> Flag to enable/disable SCR for AMR WB Audio Values: <ul style="list-style-type: none"> True - Enable False - Disable Bit to check in ParamPresenceMask - 21
<i>pAmrMode</i>	<ul style="list-style-type: none"> BitMask for AMR NB modes allowed Values: <ul style="list-style-type: none"> 0x1 - 4.75 kbps 0x2 - 5.15 kbps 0x4 - 5.9 kbps 0x8 - 6.17 kbps 0x10 - 7.4 kbps 0x20 - 7.95 kbps 0x40 - 10.2 kbps 0x80 - 12.2 kbps Bit to check in ParamPresenceMask - 22

<i>pAmrWBMode</i>	<ul style="list-style-type: none"> • BitMask for AMR WB modes allowed • Values: <ul style="list-style-type: none"> – 0x1 - 6.60 kbps – 0x2 - 8.85 kbps – 0x4 - 12.65 kbps – 0x8 - 14.25 kbps – 0x10 - 15.85 kbps – 0x20 - 18.25 kbps – 0x40 - 19.85 kbps – 0x80 - 23.05 kbps – 0x100 - 23.85 kbps • Bit to check in ParamPresenceMask - 23
<i>pAmrOctet-Aligned</i>	<ul style="list-style-type: none"> • Flag to indicate if the octet is aligned for AMR NB Audio • Values: <ul style="list-style-type: none"> – True - Aligned – False - Not aligned, Bandwidth Efficient mode • Bit to check in ParamPresenceMask - 24
<i>pAmrWBOctet-Aligned</i>	<ul style="list-style-type: none"> • Flag to indicate if the octet is aligned for AMR WB Audio • Values: <ul style="list-style-type: none"> – True - Aligned – False - Not aligned, Bandwidth Efficient mode • Bit to check in ParamPresenceMask - 25
<i>pRingingTimer</i>	<ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 26
<i>pRingBackTimer</i>	<ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 27
<i>pRTPRTCP-InactTimer</i>	<ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 28

8.741.2 Field Documentation

8.741.2.1 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pAmrMode

8.741.2.2 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pAmrOctetAligned

8.741.2.3 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pAmrWbEnable

- 8.741.2.4 uint16_t* unpack_ims_SLQSGetIMSVolPConfig_t::pAmrWBMode
- 8.741.2.5 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pAmrWBOctetAligned
- 8.741.2.6 swi_uint256_t unpack_ims_SLQSGetIMSVolPConfig_t::ParamPresenceMask
- 8.741.2.7 uint16_t* unpack_ims_SLQSGetIMSVolPConfig_t::pMinSessionExpiryTimer
- 8.741.2.8 uint16_t* unpack_ims_SLQSGetIMSVolPConfig_t::pRingBackTimer
- 8.741.2.9 uint16_t* unpack_ims_SLQSGetIMSVolPConfig_t::pRingingTimer
- 8.741.2.10 uint16_t* unpack_ims_SLQSGetIMSVolPConfig_t::pRTPRTCPInactTimer
- 8.741.2.11 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pScrAmrEnable
- 8.741.2.12 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pScrAmrWbEnable
- 8.741.2.13 uint16_t* unpack_ims_SLQSGetIMSVolPConfig_t::pSessionExpiryTimer
- 8.741.2.14 uint8_t* unpack_ims_SLQSGetIMSVolPConfig_t::pSettingResp

8.742 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](#)

8.742.1 Detailed Description

This structure contains the SLQSGetRegMgrConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> • Settings Response • Bit to check in ParamPresenceMask - 16
<i>pPCSCFPort</i>	<ul style="list-style-type: none"> • Proxy call session control function port • Bit to check in ParamPresenceMask - 17
<i>pPriCSCFPort-NameLen(IN/OUT)</i>	<ul style="list-style-type: none"> • Size in bytes assigned to the primary CSCF Port name parameter to follow
<i>pPriCSCFPort-Name</i>	<ul style="list-style-type: none"> • Call Session control port, fully qualified domain name • Length of this string must be specified in pPriCSCFPortNameLen parameter • Bit to check in ParamPresenceMask - 18

<i>pIMSTestMode</i>	<ul style="list-style-type: none"> IMS Test mode Enabled. <ul style="list-style-type: none"> TRUE - Enabled FALSE - Disabled Bit to check in ParamPresenceMask - 19
---------------------	---

Note

pPriCSCFPortNameLen must be set to a valid value during API call to retrieve pPriCSCFPortName.

8.742.2 Field Documentation

8.742.2.1 swi_uint256_t unpack_ims_SLQSGetRegMgrConfig_t::ParamPresenceMask

8.742.2.2 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pIMSTestMode

8.742.2.3 uint16_t* unpack_ims_SLQSGetRegMgrConfig_t::pPCSCFPort

8.742.2.4 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pPriCSCFPortName

8.742.2.5 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pPriCSCFPortNameLen

8.742.2.6 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pSettingResp

8.743 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](#)

8.743.1 Detailed Description

This structure contains the SLQSGetSIPConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings Response Bit to check in ParamPresenceMask - 16
<i>pSIPLocalPort</i>	<ul style="list-style-type: none"> Primary call session control function SIP port number Bit to check in ParamPresenceMask - 17

<i>pTimerSIPReg</i>	<ul style="list-style-type: none"> Initial SIP registration duration from the User equipment, in seconds Bit to check in ParamPresenceMask - 18
<i>pSubscribeTimer</i>	<ul style="list-style-type: none"> Duration of the subscription by the UE for IMS registration notifications, in seconds Bit to check in ParamPresenceMask - 19
<i>pTimerT1</i>	<ul style="list-style-type: none"> RTT estimate, in milliseconds Bit to check in ParamPresenceMask - 20
<i>pTimerT2</i>	<ul style="list-style-type: none"> The maximum retransmit interval for non-invite requests and invite responses, in milliseconds Bit to check in ParamPresenceMask - 21
<i>pTimerTf</i>	<ul style="list-style-type: none"> Non-invite transaction timeout timer, in milliseconds Bit to check in ParamPresenceMask - 22
<i>pSigComp-Enabled</i>	<ul style="list-style-type: none"> Sig Comp Status <ul style="list-style-type: none"> TRUE - Sig Comp Enabled FALSE - Sig Comp Disabled Bit to check in ParamPresenceMask - 23

8.743.2 Field Documentation

8.743.2.1 `swi_uint256_t unpack_ims_SLQSGetSIPConfig_t::ParamPresenceMask`

8.743.2.2 `uint8_t* unpack_ims_SLQSGetSIPConfig_t::pSettingResp`

8.743.2.3 `uint8_t* unpack_ims_SLQSGetSIPConfig_t::pSigCompEnabled`

8.743.2.4 `uint16_t* unpack_ims_SLQSGetSIPConfig_t::pSIPLocalPort`

8.743.2.5 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pSubscribeTimer`

8.743.2.6 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerSIPReg`

8.743.2.7 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerT1`

8.743.2.8 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerT2`

8.743.2.9 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerTf`

8.744 `unpack_ims_SLQSRegMgrCfgCallback_ind_t` Struct Reference

Data Fields

- [ims_PCSCFPortInfo PCTlv](#)

- [ims_CSCFPortNameInfo](#) PNTlv
- [ims_IMSTestModelInfo](#) TMTlv
- [swi_uint256_t](#) ParamPresenceMask

8.744.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

Parameters

<i>PCTlv</i>	<ul style="list-style-type: none"> • Primary PCSCF Port info • See ims_PCSCFPortInfo for more information • Bit to check in ParamPresenceMask - 16
<i>PNTlv</i>	<ul style="list-style-type: none"> • CSCF Port info • see ims_CSCFPortNameInfo for more information • Bit to check in ParamPresenceMask - 17
<i>TMTlv</i>	<ul style="list-style-type: none"> • IMS test mode info • see ims_IMSTestModelInfo for more information • Bit to check in ParamPresenceMask - 18

Note

: None

8.744.2 Field Documentation

8.744.2.1 [swi_uint256_t](#) unpack_ims_SLQSRegMgrCfgCallBack_ind_t::ParamPresenceMask

8.744.2.2 [ims_PCSCFPortInfo](#) unpack_ims_SLQSRegMgrCfgCallBack_ind_t::PCTlv

8.744.2.3 [ims_CSCFPortNameInfo](#) unpack_ims_SLQSRegMgrCfgCallBack_ind_t::PNTlv

8.744.2.4 [ims_IMSTestModelInfo](#) unpack_ims_SLQSRegMgrCfgCallBack_ind_t::TMTlv

8.745 unpack_ims_SLQSSetIMSSMSConfig_t Struct Reference

Data Fields

- [uint8_t](#) * [pSettingResp](#)
- [swi_uint256_t](#) ParamPresenceMask

8.745.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE Bit to check in ParamPresenceMask - 16
---------------------	--

8.745.2 Field Documentation

8.745.2.1 `swi_uint256_t` `unpack_ims_SLQSSetIMSSMSConfig_t::ParamPresenceMask`8.745.2.2 `uint8_t*` `unpack_ims_SLQSSetIMSSMSConfig_t::pSettingResp`8.746 `unpack_ims_SLQSSetIMSUserConfig_t` Struct Reference

Data Fields

- `uint8_t *` [pSettingResp](#)
- [swi_uint256_t](#) `ParamPresenceMask`

8.746.1 Detailed Description

This structure contains the `SLQSSetIMSUserConfig` unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE Bit to check in ParamPresenceMask - 16
---------------------	--

8.746.2 Field Documentation

8.746.2.1 `swi_uint256_t` `unpack_ims_SLQSSetIMSUserConfig_t::ParamPresenceMask`8.746.2.2 `uint8_t*` `unpack_ims_SLQSSetIMSUserConfig_t::pSettingResp`8.747 `unpack_ims_SLQSSetIMSVoIPConfig_t` Struct Reference

Data Fields

- `uint8_t *` [pSettingResp](#)
- [swi_uint256_t](#) `ParamPresenceMask`

8.747.1 Detailed Description

This structure contains the `SLQSSetIMSVoIPConfig` unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE Bit to check in ParamPresenceMask - 16
---------------------	--

8.747.2 Field Documentation

8.747.2.1 swi_uint256_t unpack_ims_SLQSSetIMSVolPConfig_t::ParamPresenceMask

8.747.2.2 uint8_t* unpack_ims_SLQSSetIMSVolPConfig_t::pSettingResp

8.748 unpack_ims_SLQSSetRegMgrConfig_t Struct Reference

Data Fields

- uint8_t * [pSettingResp](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.748.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE Bit to check in ParamPresenceMask - 16
---------------------	--

8.748.2 Field Documentation

8.748.2.1 swi_uint256_t unpack_ims_SLQSSetRegMgrConfig_t::ParamPresenceMask

8.748.2.2 uint8_t* unpack_ims_SLQSSetRegMgrConfig_t::pSettingResp

8.749 unpack_ims_SLQSSetSIPConfig_t Struct Reference

Data Fields

- uint8_t * [pSettingResp](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.749.1 Detailed Description

This structure contains the SLQSSetSIPConfig unpack parameters.

Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE Bit to check in ParamPresenceMask - 16
---------------------	--

8.749.2 Field Documentation

8.749.2.1 `swi_uint256_t` `unpack_ims_SLQSSetSIPConfig_t::ParamPresenceMask`8.749.2.2 `uint8_t*` `unpack_ims_SLQSSetSIPConfig_t::pSettingResp`8.750 `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

8.750.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

Parameters

<i>SPTlv</i>	<ul style="list-style-type: none"> SIP port tlv See ims_SIPPortInfo for more information Bit to check in ParamPresenceMask - 16
<i>SRTlv</i>	<ul style="list-style-type: none"> SIP Registration Timer tlv see ims_SIPRegnTmrInfo for more information Bit to check in ParamPresenceMask - 17
<i>STTlv</i>	<ul style="list-style-type: none"> Subscriber timer tlv see ims_SubscrTmrInfo for more information Bit to check in ParamPresenceMask - 18
<i>TT1Tlv</i>	<ul style="list-style-type: none"> Timer T1 tlv see ims_TmrT1Info for more information Bit to check in ParamPresenceMask - 19

<i>TT2Tlv</i>	<ul style="list-style-type: none"> • Timer T2 tlv • see ims_TmrT2Info for more information • Bit to check in ParamPresenceMask - 20
<i>TTfTlv</i>	<ul style="list-style-type: none"> • Timer Tf tlv • see ims_TmrTfInfo for more information • Bit to check in ParamPresenceMask - 21
<i>SCTlv</i>	<ul style="list-style-type: none"> • SigComp Status Tlv • see ims_SigCompEnInfo for more information • Bit to check in ParamPresenceMask - 22

Note

: None

8.750.2 Field Documentation

8.750.2.1 swi_uint256_t unpack_ims_SLQSSIPCfgCallBack_ind_t::ParamPresenceMask

8.750.2.2 ims_SigCompEnInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SCTlv

8.750.2.3 ims_SIPPortInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SPTlv

8.750.2.4 ims_SIPRegnTmrInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SRTlv

8.750.2.5 ims_SubscrTmrInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::STTlv

8.750.2.6 ims_TmrT1Info unpack_ims_SLQSSIPCfgCallBack_ind_t::TT1Tlv

8.750.2.7 ims_TmrT2Info unpack_ims_SLQSSIPCfgCallBack_ind_t::TT2Tlv

8.750.2.8 ims_TmrTfInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::TTfTlv

8.751 unpack_ims_SLQSSMSCfgCallBack_ind_t Struct Reference**Data Fields**

- [ims_SMSFmtInfo](#) SFTlv
- [ims_SMSolPNwInfo](#) SINTlv
- [ims_PhCtxtURIInfo](#) PCURTlv
- [swi_uint256_t](#) ParamPresenceMask

8.751.1 Detailed Description

Structure used to store SMS Config Indication Parameters.

Parameters

<i>SFTIv</i>	<ul style="list-style-type: none"> • SMS format info • See ims_SMSFmtlInfo for more information • Bit to check in ParamPresenceMask - 16
<i>SINTIv</i>	<ul style="list-style-type: none"> • SMS over IP network info • see ims_SMSoIPNwInfo for more information • Bit to check in ParamPresenceMask - 17
<i>PCURTIv</i>	<ul style="list-style-type: none"> • Phone context URI info • see ims_PhCtxtURIInfo for more information • Bit to check in ParamPresenceMask - 18

Note

: None

8.751.2 Field Documentation

8.751.2.1 `swi_uint256_t` `unpack_ims_SLQSSMSCfgCallBack_ind_t::ParamPresenceMask`

8.751.2.2 `ims_PhCtxtURIInfo` `unpack_ims_SLQSSMSCfgCallBack_ind_t::PCURTIv`

8.751.2.3 `ims_SMSFmtlInfo` `unpack_ims_SLQSSMSCfgCallBack_ind_t::SFTIv`

8.751.2.4 `ims_SMSoIPNwInfo` `unpack_ims_SLQSSMSCfgCallBack_ind_t::SINTIv`

8.752 `unpack_ims_SLQSUUserCfgCallBack_ind_t` Struct Reference

Data Fields

- [ims_IMSDomainInfo](#) `IDTIv`
- [swi_uint256_t](#) `ParamPresenceMask`

8.752.1 Detailed Description

Structure used to store SMS Config Indication Parameters.

Parameters

<i>IDTIv</i>	<ul style="list-style-type: none"> • IMS domain info • See ims_IMSDomainInfo for more information • Bit to check in ParamPresenceMask - 16
--------------	--

Note

: None

8.752.2 Field Documentation

8.752.2.1 `ims_IMSDomainInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::IDTlv`

8.752.2.2 `swi_uint256_t` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ParamPresenceMask`

8.753 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`

8.753.1 Detailed Description

Structure used to store VOIP Config Indication Parameters.

Parameters

<i>SDTlv</i>	<ul style="list-style-type: none"> • Session Duration tlv • See ims_SessDurInfo for more information • Bit to check in ParamPresenceMask - 16
<i>MSETlv</i>	<ul style="list-style-type: none"> • Minimum Session Timer tlv • see ims_MinSessExpInfo for more information • Bit to check in ParamPresenceMask - 17
<i>EAWTlv</i>	<ul style="list-style-type: none"> • Enable AMR WB tlv • see ims_EnabAMRWBInfo for more information • Bit to check in ParamPresenceMask - 18
<i>ESATlv</i>	<ul style="list-style-type: none"> • Enable SCR AMR NB tlv • see ims_EnabSCRAMRInfo for more information • Bit to check in ParamPresenceMask - 19

<i>ESAWTlv</i>	<ul style="list-style-type: none"> • Enable SCR AMR WB tlv • see ims_EnabSCRAMRWBInfo for more information • Bit to check in ParamPresenceMask - 20
<i>AMTlv</i>	<ul style="list-style-type: none"> • AMR NB Mode tlv • see ims_AMRModelInfo for more information • Bit to check in ParamPresenceMask - 21
<i>AWMTlv</i>	<ul style="list-style-type: none"> • AMR WB Mode Tlv • see ims_AMRWBModelInfo for more information • Bit to check in ParamPresenceMask - 22
<i>AOATlv</i>	<ul style="list-style-type: none"> • AMR NB Octet Aligned tlv • see ims_AMROctAlgnInfo for more information • Bit to check in ParamPresenceMask - 23
<i>AWOATlv</i>	<ul style="list-style-type: none"> • AMR WB Octet Aligned tlv • see ims_AMRWBOctAlgnInfo for more information • Bit to check in ParamPresenceMask - 24
<i>RTTlv</i>	<ul style="list-style-type: none"> • Ringing Timer Duration tlv • see ims_RngTmrInfo for more information • Bit to check in ParamPresenceMask - 25
<i>RBTTlv</i>	<ul style="list-style-type: none"> • Ringback Duration tlv • see ims_RngBkTmrInfo for more information • Bit to check in ParamPresenceMask - 26
<i>RTIDTlv</i>	<ul style="list-style-type: none"> • RTP/RTCP Inactivity Timer Duration Tlv • see ims_RTPRTCPInactTmrDurInfo for more information • Bit to check in ParamPresenceMask - 27

Note

: None

8.753.2 Field Documentation8.753.2.1 **ims_AMRModelInfo** unpack_ims_SLQSVolPCfgCallBack_ind_t::AMTlv8.753.2.2 **ims_AMROctAlgnInfo** unpack_ims_SLQSVolPCfgCallBack_ind_t::AOATlv8.753.2.3 **ims_AMRWBModelInfo** unpack_ims_SLQSVolPCfgCallBack_ind_t::AWMTlv

- 8.753.2.4 `ims_AMRWBOctAlgnInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::AWOATlv`
- 8.753.2.5 `ims_EnabAMRWBInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::EAWTlv`
- 8.753.2.6 `ims_EnabSCRAMRInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ESATlv`
- 8.753.2.7 `ims_EnabSCRAMRWBInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ESAWTlv`
- 8.753.2.8 `ims_MinSessExpInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::MSETlv`
- 8.753.2.9 `swi_uint256_t` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ParamPresenceMask`
- 8.753.2.10 `ims_RngBkTmrInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::RBTTlv`
- 8.753.2.11 `ims_RTPRTCPlnactTmrDurInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::RTIDTlv`
- 8.753.2.12 `ims_RngTmrInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::RTTlv`
- 8.753.2.13 `ims_SessDurInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::SDTlv`

8.754 unpack_imsa_SLQSGetIMSARegStatus_t Struct Reference

Data Fields

- `uint8_t` [ImsRegStatus](#)
- `uint16_t` [ImsRegErrCode](#)
- `uint32_t` [NewImsRegStatus](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.754.1 Detailed Description

This structure contains response parameters of registration status.

Parameters

<i>ImsRegStatus</i>	<ul style="list-style-type: none"> • IMS Registration Status (Deprecated). • Values <ul style="list-style-type: none"> – TRUE - UE is registered on the IMS network – FALSE - UE is not registered on the IMS network • Bit to check in ParamPresenceMask - 16
<i>ImsRegErrCode</i>	<ul style="list-style-type: none"> • IMS Registration Error Code. • An error code is returned when the IMS registration status is IMSA_STATUS_NOT_REGISTERED. -Values <ul style="list-style-type: none"> – 3xx - Redirection responses – 4xx - Client failure responses – 5xx - Server failure responses – 6xx - Global failure responses • Bit to check in ParamPresenceMask - 17

<i>NewImsReg-Status</i>	<ul style="list-style-type: none"> • New IMS Registration Status • Values <ul style="list-style-type: none"> – 0 - Not registered for IMS – 1 - Registering for IMS – 2 - Registered for IMS • Bit to check in ParamPresenceMask - 18
-------------------------	---

8.754.2 Field Documentation

8.754.2.1 uint16_t unpack_imsa_SLQSGetIMSARegStatus_t::ImsRegErrCode

8.754.2.2 uint8_t unpack_imsa_SLQSGetIMSARegStatus_t::ImsRegStatus

8.754.2.3 uint32_t unpack_imsa_SLQSGetIMSARegStatus_t::NewImsRegStatus

8.754.2.4 swi_uint256_t unpack_imsa_SLQSGetIMSARegStatus_t::ParamPresenceMask

8.755 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](#)

8.755.1 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"> • SMS Service Status. • Values <ul style="list-style-type: none"> – 0 - IMS SMS service is not available – 1 - IMS SMS is in limited service – 2 - IMS SMS is in full service • Bit to check in ParamPresenceMask - 16
--------------------------	---

<i>VoipSeriveStatus</i>	<ul style="list-style-type: none"> • VoIP Service Status. • Values <ul style="list-style-type: none"> – 0 - IMS VoIP service is not available – 2 - IMS VoIP is in full service • Bit to check in ParamPresenceMask - 17
<i>VtServiceStatus</i>	<ul style="list-style-type: none"> • VT Service Status • Values <ul style="list-style-type: none"> – 0 - IMS VT service is not available – 2 - IMS VT is in full service • Bit to check in ParamPresenceMask - 18
<i>SmsServiceRat</i>	<ul style="list-style-type: none"> • SMS service RAT • Values <ul style="list-style-type: none"> – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN • Bit to check in ParamPresenceMask - 19
<i>VoipServiceRat</i>	<ul style="list-style-type: none"> • VoIP service RAT. • Values <ul style="list-style-type: none"> – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN • Bit to check in ParamPresenceMask - 20
<i>VtServiceRat</i>	<ul style="list-style-type: none"> • VT service RAT. • Values <ul style="list-style-type: none"> – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN • Bit to check in ParamPresenceMask - 21
<i>UtServiceStatus</i>	<ul style="list-style-type: none"> • UT service Status. • Values <ul style="list-style-type: none"> – 0 - IMS UT service is not available – 2 - IMS UT is in full service • Bit to check in ParamPresenceMask - 22

<i>UtServiceRat</i>	<ul style="list-style-type: none"> • UT service RAT. • Values <ul style="list-style-type: none"> – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN • Bit to check in ParamPresenceMask - 23
<i>VsServiceStatus</i>	<ul style="list-style-type: none"> • VS service Status. • Values <ul style="list-style-type: none"> – 0 - IMS UT service is not available – 2 - IMS UT is in full service • Bit to check in ParamPresenceMask - 24
<i>VsServiceRat</i>	<ul style="list-style-type: none"> • VS service RAT. • Values <ul style="list-style-type: none"> – 0 - IMS service is registered on WLAN – 1 - IMS service is registered on WWAN – 2 - IMS service is registered on interworking WLAN • Bit to check in ParamPresenceMask - 25

8.755.2 Field Documentation

8.755.2.1 `swi_uint256_t unpack_imsa_SLQSGetIMSAServiceStatus_t::ParamPresenceMask`

8.755.2.2 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::SmsServiceRat`

8.755.2.3 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::SmsServiceStatus`

8.755.2.4 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::UtServiceRat`

8.755.2.5 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::UtServiceStatus`

8.755.2.6 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VoipServiceRat`

8.755.2.7 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VoipServiceStatus`

8.755.2.8 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VsServiceRat`

8.755.2.9 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VsServiceStatus`

8.755.2.10 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VtServiceRat`

8.755.2.11 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VtServiceStatus`

8.756 `unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t` Struct Reference

Data Fields

- uint8_t [PdpConnState](#)
- [imsa_IMSFailErrCodeTlv](#) FailErrCode
- [swi_uint256_t](#) ParamPresenceMask

8.756.1 Detailed Description

Structure used to store IMSA PDP status indication Parameters.

Parameters

<i>PdpConnState</i>	<ul style="list-style-type: none">• Values:<ul style="list-style-type: none">– 0- not connected– 1- connected• Bit to check in ParamPresenceMask - 1
<i>FailErrCode</i>	<ul style="list-style-type: none">• Fail error code when PDP is not connected• See imsa_IMSFailErrCodeTlv for more information• Bit to check in ParamPresenceMask - 16

Note

: None

8.756.2 Field Documentation

8.756.2.1 [imsa_IMSFailErrCodeTlv](#) unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t::FailErrCode

8.756.2.2 [swi_uint256_t](#) unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t::ParamPresenceMask

8.756.2.3 [uint8_t](#) unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t::PdpConnState

8.757 unpack_imsa_SLQSImsaRatStatusCallBack_ind_t Struct Reference

Data Fields

- [imsa_RatHandoverStatusInfo](#) RatHandover
- [swi_uint256_t](#) ParamPresenceMask

8.757.1 Detailed Description

Structure used to store IMSA RAT status indication Parameters.

Parameters

<i>RatHandover</i>	<ul style="list-style-type: none">• RAT handover status info• See imsa_RatHandoverStatusInfo for more information• Bit to check in ParamPresenceMask - 16
--------------------	--

Note

: None

8.757.2 Field Documentation

8.757.2.1 `swi_uint256_t` `unpack_imsa_SLQSImsaRatStatusCallBack_ind_t::ParamPresenceMask`8.757.2.2 `imsa_RatHandoverStatusInfo` `unpack_imsa_SLQSImsaRatStatusCallBack_ind_t::RatHandover`8.758 `unpack_imsa_SLQSImsaRegStatusCallBack_ind_t` Struct Reference

Data Fields

- [imsa_IMSRegStatusInfo](#) IMSRegistration
- [imsa_IMSRegStatusErrorCodeInfo](#) IMSRegistrationError
- [imsa_NewIMSRegStatusInfo](#) NewIMSRegistration
- [swi_uint256_t](#) ParamPresenceMask

8.758.1 Detailed Description

Structure used to store IMSA registration status indication Parameters.

Parameters

<i>IMSRegistration</i>	<ul style="list-style-type: none"> • IMS registration info • See imsa_IMSRegStatusInfo for more information • Bit to check in ParamPresenceMask - 1
<i>IMSRegistration-Error</i>	<ul style="list-style-type: none"> • IMS registration error • see imsa_IMSRegStatusErrorCodeInfo for more information • Bit to check in ParamPresenceMask - 16
<i>NewIMS-Registration</i>	<ul style="list-style-type: none"> • New IMS registration info • see imsa_NewIMSRegStatusInfo for more information • Bit to check in ParamPresenceMask - 17

Note

: None

8.758.2 Field Documentation

8.758.2.1 `imsa_IMSRegStatusInfo` `unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::IMSRegistration`8.758.2.2 `imsa_IMSRegStatusErrorCodeInfo` `unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::IMSRegistrationError`8.758.2.3 `imsa_NewIMSRegStatusInfo` `unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::NewIMSRegistration`

8.758.2.4 swi_uint256_t unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::ParamPresenceMask

8.759 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

8.759.1 Detailed Description

Structure used to store IMSA service status indication Parameters.

Parameters

<i>SmsService</i>	<ul style="list-style-type: none"> • SMS service info • See imsa_SmsSvcStatusInfo for more information • Bit to check in ParamPresenceMask - 16
<i>VoipService</i>	<ul style="list-style-type: none"> • VOIP service info • see imsa_VoipSvcStatusInfo for more information • Bit to check in ParamPresenceMask - 17
<i>VtService</i>	<ul style="list-style-type: none"> • VT service info • see imsa_VtSvcStatusInfo for more information • Bit to check in ParamPresenceMask - 18
<i>SmsRat</i>	<ul style="list-style-type: none"> • SMS RAT info • see imsa_SmsRatInfo for more information • Bit to check in ParamPresenceMask - 19
<i>VoipRat</i>	<ul style="list-style-type: none"> • VOIP RAT info • see imsa_VoipRatInfo for more information • Bit to check in ParamPresenceMask - 20
<i>VtRat</i>	<ul style="list-style-type: none"> • VT RAT info • see imsa_VtRatInfo for more information • Bit to check in ParamPresenceMask - 21

<i>UtService</i>	<ul style="list-style-type: none"> • UT service info • see imsa_UtSvcStatusInfo for more information • Bit to check in ParamPresenceMask - 22
<i>UtRat</i>	<ul style="list-style-type: none"> • UT RAT info • see imsa_UtRatInfo for more information • Bit to check in ParamPresenceMask - 23

Note

: None

8.759.2 Field Documentation8.759.2.1 `swi_uint256_t` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::ParamPresenceMask`8.759.2.2 `imsa_SmsRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::SmsRat`8.759.2.3 `imsa_SmsSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::SmsService`8.759.2.4 `imsa_UtRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::UtRat`8.759.2.5 `imsa_UtSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::UtService`8.759.2.6 `imsa_VoipRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VoipRat`8.759.2.7 `imsa_VoipSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VoipService`8.759.2.8 `imsa_VtRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VtRat`8.759.2.9 `imsa_VtSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VtService`**8.760 `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](#)

8.760.1 Detailed Description

This structure contains Best Available Position

Parameters

<i>status</i>	<ul style="list-style-type: none"> • Valid values: <ul style="list-style-type: none"> – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
<i>xid</i>	<p>Transaction ID that was specified in the Get Best Available Position request.</p> <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 16

<i>pLatitude</i>	<ul style="list-style-type: none"> • Latitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude • Bit to check in ParamPresenceMask - 17
<i>pLongitude</i>	<ul style="list-style-type: none"> • Longitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude • Bit to check in ParamPresenceMask - 18
<i>pHorUncCircular</i>	<ul style="list-style-type: none"> • Horizontal position uncertainty. • Units - Meters • Bit to check in ParamPresenceMask - 19
<i>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> • Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883 • Bit to check in ParamPresenceMask - 20
<i>pVertUnc</i>	<ul style="list-style-type: none"> • Vertical uncertainty. • Units - Meters • Bit to check in ParamPresenceMask - 21
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> • UTC timestamp • Units - Milliseconds since Jan. 1, 1970 • Bit to check in ParamPresenceMask - 22
<i>pTimeUnc</i>	<ul style="list-style-type: none"> • Time uncertainty. • Units - Milliseconds • Bit to check in ParamPresenceMask - 23
<i>pHorUncEllipse-SemiMinor</i>	<ul style="list-style-type: none"> • Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters • Bit to check in ParamPresenceMask - 24

<i>pHorUncEllipse-SemiMajor</i>	<ul style="list-style-type: none"> • Semi-major axis of horizontal elliptical uncertainty. • Units: Meters • Bit to check in ParamPresenceMask - 25
<i>pHorUncEllipse-OrientAzimuth</i>	<ul style="list-style-type: none"> • Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180 • Bit to check in ParamPresenceMask - 26
<i>pHorCirConf</i>	<ul style="list-style-type: none"> • Horizontal circular uncertainty confidence • Units: Precent • Range: 0 to 99 • Bit to check in ParamPresenceMask - 27
<i>pHorEllpConf</i>	<ul style="list-style-type: none"> • Horizontal elliptical uncertainty confidence • Units: Precent • Range: 0 to 99 • Bit to check in ParamPresenceMask - 28
<i>pHorReliability</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Location reliability is not set. – 1 - Location reliability is very low; use it at your own risk – 2 - Location reliability is low; little or no cross-checking is possible. – 3 - Location reliability is medium; limited cross-check passed – 4 - Location reliability is high; strong cross-check passed • Bit to check in ParamPresenceMask - 29
<i>pSpeed-Horizontal</i>	<ul style="list-style-type: none"> • Horizontal speed. • Units - Meters/second • Bit to check in ParamPresenceMask - 30
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> • 3-D Speed uncertainty. • Units - Meters/second. • Bit to check in ParamPresenceMask - 31
<i>pAltitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> • Altitude With Respect to Sea Level. • Units - Meters • Bit to check in ParamPresenceMask - 32

<i>pVertConfidence</i>	<ul style="list-style-type: none"> • Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99 • Bit to check in ParamPresenceMask - 33
<i>pVertReliability</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Location reliability is not set. – 1 - Location reliability is very low; use it at your own risk. – 2 - Location reliability is low; little or no cross-checking is possible – 3 - Location reliability is medium; limited cross-check passed – 4 - Location reliability is high; strong cross-check passed • Bit to check in ParamPresenceMask - 34
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> • Vertical speed. • Units - Meters/second • Bit to check in ParamPresenceMask - 35
<i>pSpeedVertical- Unc</i>	<ul style="list-style-type: none"> • Vertical speed • Units: Meters/second • Bit to check in ParamPresenceMask - 36
<i>pHeading</i>	<ul style="list-style-type: none"> • Heading. • Units - Degree • Range 0 to 359.999 • Bit to check in ParamPresenceMask - 37
<i>pHeadingUnc</i>	<ul style="list-style-type: none"> • Heading uncertainty. • Units - Degree • Range 0 to 359.999 • Bit to check in ParamPresenceMask - 38
<i>pMagnetic- Deviation</i>	<ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 39

<i>pTechnologyMask</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0x00000001 - Satellites were used to generate the fix – 0x00000002 - Cell towers were used to generate the fix – 0x00000004 - Wi-Fi access points were used to generate the fix – 0x00000008 - Sensors were used to generate the fix – 0x00000010 - Reference Location was used to generate the fix – 0x00000020 - Coarse position injected into the location engine was used to generate the fix – 0x00000040 - AFLT was used to generate the fix – 0x00000080 - GNSS and network-provided measurements were used to generate the fix • Bit to check in ParamPresenceMask - 40
<i>-pPrecisionDilution</i>	<ul style="list-style-type: none"> • See loc_precisionDilution for more information • Bit to check in ParamPresenceMask - 41
<i>-pGpsTime</i>	<ul style="list-style-type: none"> • See loc_gpsTime for more information • Bit to check in ParamPresenceMask - 42
<i>pTimeSrc</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites. • Bit to check in ParamPresenceMask - 43
<i>-pSensorDataUsage</i>	<ul style="list-style-type: none"> • See loc_sensorDataUsage for more information • Bit to check in ParamPresenceMask - 44
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> • See loc_svUsedforFix for more information • Bit to check in ParamPresenceMask - 45

8.760.2 Field Documentation

- 8.760.2.1 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid`
- 8.760.2.2 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel`
- 8.760.2.3 `swi_uint256_t unpack_loc_BestAvailPos_Ind_t::ParamPresenceMask`
- 8.760.2.4 `loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime`
- 8.760.2.5 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading`
- 8.760.2.6 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc`
- 8.760.2.7 `uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf`
- 8.760.2.8 `uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf`
- 8.760.2.9 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability`
- 8.760.2.10 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular`
- 8.760.2.11 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.760.2.12 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor`
- 8.760.2.13 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor`
- 8.760.2.14 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude`
- 8.760.2.15 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude`
- 8.760.2.16 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation`
- 8.760.2.17 `loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution`
- 8.760.2.18 `loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage`
- 8.760.2.19 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal`
- 8.760.2.20 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc`
- 8.760.2.21 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical`
- 8.760.2.22 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc`
- 8.760.2.23 `loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix`
- 8.760.2.24 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask`
- 8.760.2.25 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc`
- 8.760.2.26 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc`
- 8.760.2.27 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc`

8.760.2.28 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence

8.760.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability

8.760.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc

8.760.2.31 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid

8.760.2.32 uint32_t unpack_loc_BestAvailPos_Ind_t::status

8.760.2.33 uint16_t unpack_loc_BestAvailPos_Ind_t::Tlvresult

8.761 unpack_loc_CradleMountCallback_Ind_t Struct Reference

Data Fields

- uint32_t [cradleMountConfigStatus](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.761.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

Parameters

<i>cradleMount-ConfigStatus</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
---------------------------------	--

8.761.2 Field Documentation

8.761.2.1 uint32_t unpack_loc_CradleMountCallback_Ind_t::cradleMountConfigStatus

8.761.2.2 [swi_uint256_t](#) unpack_loc_CradleMountCallback_Ind_t::ParamPresenceMask

8.762 unpack_loc_Delete_Assist_Data_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

- [swi_uint256_t ParamPresenceMask](#)

8.762.1 Detailed Description

This structure contains LOC delete assist data unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.• Bit to check in ParamPresenceMask - 2
------------------	--

8.762.2 Field Documentation

8.762.2.1 [swi_uint256_t unpack_loc_Delete_Assist_Data_t::ParamPresenceMask](#)

8.762.2.2 [uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult](#)

8.763 unpack_loc_DeleteAssistData_Ind_t Struct Reference

Data Fields

- [uint32_t status](#)
- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.763.1 Detailed Description

Contain the parameters passed for SetLocDeleteAssistDataCallback by the device.

Parameters

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

8.763.2 Field Documentation

8.763.2.1 swi_uint256_t unpack_loc_DeleteAssistData_Ind_t::ParamPresenceMask

8.763.2.2 uint32_t unpack_loc_DeleteAssistData_Ind_t::status

8.763.2.3 uint16_t unpack_loc_DeleteAssistData_Ind_t::Tlvresult

8.764 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

- uint32_t [engineState](#)
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.764.1 Detailed Description

This structure contains LOC Engine State field.

Parameters

<i>engineState</i>	<ul style="list-style-type: none"> • Location engine state. • Valid values <ul style="list-style-type: none"> – 1 - Location engine is on – 2 - Location engine is off • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.764.2 Field Documentation

8.764.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.764.2.2 swi_uint256_t unpack_loc_EngineState_Ind_t::ParamPresenceMask

8.764.2.3 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

8.765 unpack_loc_EventNMEA_Ind_t Struct Reference

Data Fields

- char [NMEADData](#) [201]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.765.1 Detailed Description

Contain the parameters passed for SetLocEventMaskNMEACallback by the device.

Parameters

<i>NMEADData</i>	<ul style="list-style-type: none"> • NMEA string • Type - NULL terminated string • Maximum string length (including NULL terminator) - 201 • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.765.2 Field Documentation

8.765.2.1 char unpack_loc_EventNMEA_Ind_t::NMEADData[201]

8.765.2.2 swi_uint256_t unpack_loc_EventNMEA_Ind_t::ParamPresenceMask

8.765.2.3 uint16_t unpack_loc_EventNMEA_Ind_t::Tlvresult

8.766 unpack_loc_EventRegister_t Struct Reference

Data Fields

- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.766.1 Detailed Description

This structure contains Event Register unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.• Bit to check in ParamPresenceMask - 2
------------------	--

8.766.2 Field Documentation

8.766.2.1 [swi_uint256_t unpack_loc_EventRegister_t::ParamPresenceMask](#)

8.766.2.2 [uint16_t unpack_loc_EventRegister_t::Tlvresult](#)

8.767 unpack_loc_EventTimeSyncCallback_Ind_t Struct Reference

Data Fields

- [uint32_t timeSyncRefCounter](#)
- [swi_uint256_t ParamPresenceMask](#)

8.767.1 Detailed Description

This structure contains LOC Event Time Sync Reference COUNTER

Parameters

<i>timeSyncRef-Counter</i>	<ul style="list-style-type: none">• Sent by the location engine when it needs to synchronize location engine and control point (sensor processor) times.• Bit to check in ParamPresenceMask - 1
----------------------------	--

8.767.2 Field Documentation

8.767.2.1 [swi_uint256_t unpack_loc_EventTimeSyncCallback_Ind_t::ParamPresenceMask](#)

8.767.2.2 [uint32_t unpack_loc_EventTimeSyncCallback_Ind_t::timeSyncRefCounter](#)

8.768 unpack_loc_GetOpMode_Ind_t Struct Reference

Data Fields

- uint32_t [Status](#)
- uint32_t * [pMode](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.768.1 Detailed Description

Contain the parameters passed for SetLocGetOpModeCallback by the device.

Parameters

<i>Status</i>	<ul style="list-style-type: none"> • Status of the Get Server request. • Valid values <ul style="list-style-type: none"> – eQMI_LOC_SUCCESS (0) - Request was completed successfully – eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure – eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported – eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters – eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy – eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline – eQMI_LOC_TIMEOUT (6) - Request failed because it timed out – eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested – eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request – eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed – eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure – 0xffffffff - Invalid data. • Bit to check in ParamPresenceMask - 1
<i>pMode</i>	<ul style="list-style-type: none"> • Current operation mode. • Valid values: <ul style="list-style-type: none"> – eQMI_LOC_OPER_MODE_DEFAULT(1) - Use the default engine mode – eQMI_LOC_OPER_MODE_MSB (2)- Use the MS-based mode – eQMI_LOC_OPER_MODE_MSA (3)- Use the MS-assisted mode – eQMI_LOC_OPER_MODE_STANDALONE (4) - Use Standalone mode – eQMI_LOC_OPER_MODE_CELL_ID (5) - Use cell ID; this mode is only valid for GSM/UMTS networks – 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 • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.768.2 Field Documentation

8.768.2.1 `swi_uint256_t` `unpack_loc_GetOpMode_Ind_t::ParamPresenceMask`

8.768.2.2 `uint32_t*` `unpack_loc_GetOpMode_Ind_t::pMode`

8.768.2.3 `uint32_t` `unpack_loc_GetOpMode_Ind_t::Status`

8.769 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](#)

8.769.1 Detailed Description

Contain the parameters passed for SetLocGetServerCallback by the device.

Parameters

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

<i>serverType</i>	<ul style="list-style-type: none"> • Type of server. • Valid values <ul style="list-style-type: none"> – eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE – eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC – eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP – eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) - Server type is custom PDE • Bit to check in ParamPresenceMask - 2
<i>plPv4AddrInfo</i>	<ul style="list-style-type: none"> • See loc_IPv4Info for more information. • Bit to check in ParamPresenceMask - 16
<i>plPv6AddrInfo</i>	<ul style="list-style-type: none"> • See loc_IPv6Info for more information. • Bit to check in ParamPresenceMask - 17
<i>pURL</i>	<ul style="list-style-type: none"> • See loc_urlAddr for more information. • Bit to check in ParamPresenceMask - 18

8.769.2 Field Documentation

8.769.2.1 `swi_uint256_t unpack_loc_GetServer_Ind_t::ParamPresenceMask`

8.769.2.2 `loc_IPv4Info* unpack_loc_GetServer_Ind_t::plPv4AddrInfo`

8.769.2.3 `loc_IPv6Info* unpack_loc_GetServer_Ind_t::plPv6AddrInfo`

8.769.2.4 `loc_urlAddr* unpack_loc_GetServer_Ind_t::pURL`

8.769.2.5 `uint32_t unpack_loc_GetServer_Ind_t::serverStatus`

8.769.2.6 `uint32_t unpack_loc_GetServer_Ind_t::serverType`

8.769.2.7 `uint16_t unpack_loc_GetServer_Ind_t::Tlvresult`

8.770 unpack_loc_GnssSvInfo_Ind_t Struct Reference

Data Fields

- `uint8_t altitudeAssumed`
- `loc_satelliteInfo * pSatelliteInfo`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.770.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

Parameters

<i>altitudeAssumed</i>	<ul style="list-style-type: none">Indicates whether altitude is assumed or calculated<ul style="list-style-type: none">0x00 (FALSE) - Valid altitude is calculated0x01 (TRUE) - Valid altitude is assumed; there may not be enough satellites to determine precise altitudeBit to check in ParamPresenceMask - 1
<i>pSatelliteInfo</i>	<ul style="list-style-type: none">See loc_satelliteInfo for more information.Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none">unpack result

8.770.2 Field Documentation

8.770.2.1 `uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed`

8.770.2.2 `swi_uint256_t unpack_loc_GnssSvInfo_Ind_t::ParamPresenceMask`

8.770.2.3 `loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo`

8.770.2.4 `uint16_t unpack_loc_GnssSvInfo_Ind_t::Tlvresult`

8.771 unpack_loc_InjectPositionCallback_Ind_t Struct Reference

Data Fields

- `uint32_t` [status](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.771.1 Detailed Description

Contain the parameters passed for LOC Inject Position indication by the device.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • Position Injection Status • Valid values: <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
---------------	--

Note

None

8.771.2 Field Documentation

8.771.2.1 `swi_uint256_t unpack_loc_InjectPositionCallback_Ind_t::ParamPresenceMask`8.771.2.2 `uint32_t unpack_loc_InjectPositionCallback_Ind_t::status`8.772 `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`

8.772.1 Detailed Description

This structure contains LOC Inject Sensor Data

Parameters

<i>injectSensorDataStatus</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
<i>pOpaqueIdentifier</i> [OPTIONAL]	<ul style="list-style-type: none"> • Sent in by the client echoed so the client can relate the indication to the request. • NULL when this TLV not present • Bit to check in ParamPresenceMask - 16
<i>pAccelSamplesAccepted</i> [OPTIONAL]	<ul style="list-style-type: none"> • Lets the client know how many 3-axis accelerometer samples were accepted. • This field is present only if the accelerometer samples were sent in the request. • NULL when this TLV not present • Bit to check in ParamPresenceMask - 17
<i>pGyroSamplesAccepted</i> [OPTIONAL]	<ul style="list-style-type: none"> • Lets the client know how many 3-axis gyroscope samples were accepted. • This field is present only if the gyroscope samples were sent in the request. • NULL when this TLV not present • Bit to check in ParamPresenceMask - 18
<i>pAccelTempSamplesAccepted</i> [OPTIONAL]	<ul style="list-style-type: none"> • Lets the client know how many accelerometer temperature samples were accepted. • This field is present only if the accelerometer temperature samples were sent in the request. • NULL when this TLV not present • Bit to check in ParamPresenceMask - 19
<i>pGyroTempSamplesAccepted</i> [OPTIONAL]	<ul style="list-style-type: none"> • Lets the client know how many gyroscope temperature samples were accepted. • This field is present only if the gyroscope temperature samples were sent in the request. • NULL when this TLV not present • Bit to check in ParamPresenceMask - 20

8.772.2 Field Documentation

8.772.2.1 uint32_t unpack_loc_InjectSensorDataCallback_Ind_t::injectSensorDataStatus

8.772.2.2 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pAccelSamplesAccepted

8.772.2.3 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pAccelTempSamplesAccepted

8.772.2.4 swi_uint256_t unpack_loc_InjectSensorDataCallback_Ind_t::ParamPresenceMask

8.772.2.5 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pGyroSamplesAccepted

8.772.2.6 uint8_t* unpack_loc_InjectSensorDataCallback_Ind_t::pGyroTempSamplesAccepted

8.772.2.7 uint32_t* unpack_loc_InjectSensorDataCallback_Ind_t::pOpaqueldentifier

8.773 unpack_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference

Data Fields

- uint32_t [injectTimeSyncStatus](#)
- swi_uint256_t [ParamPresenceMask](#)

8.773.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

Parameters

<i>injectTimeSync-Status</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure
------------------------------	---

- * - Bit to check in ParamPresenceMask - 1

8.773.2 Field Documentation

8.773.2.1 uint32_t unpack_loc_InjectTimeSyncDataCallback_Ind_t::injectTimeSyncStatus

8.773.2.2 swi_uint256_t unpack_loc_InjectTimeSyncDataCallback_Ind_t::ParamPresenceMask

8.774 unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference

Data Fields

- uint32_t [status](#)
- swi_uint256_t [ParamPresenceMask](#)

8.774.1 Detailed Description

Contain the parameters passed for Inject UTC time indication by the device.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • Status of the UTC Time Injection request • Valid values: <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
---------------	---

Note

None

8.774.2 Field Documentation

8.774.2.1 `swi_uint256_t unpack_loc_InjectUTCtimeCallback_Ind_t::ParamPresenceMask`

8.774.2.2 `uint32_t unpack_loc_InjectUTCtimeCallback_Ind_t::status`

8.775 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

8.775.1 Detailed Description

This structure contains Event Position Report Indication unpack

Parameters

<i>sessionStatus</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Session was successful – 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. – 2 - Session failed.. – 3 - Fix request failed because the session timed out. – 4 - Fix request failed because the session was ended by the user. – 5 - Fix request failed due to bad parameters in the request. – 6 - Fix request failed because the phone is offline. – 7 - Fix request failed because the engine is locked • Bit to check in ParamPresenceMask - 1
<i>sessionId</i>	<ul style="list-style-type: none"> • ID of the session that was specified in the Start request • Range - 0 to 255 • Bit to check in ParamPresenceMask - 2
<i>pLatitude</i>	<ul style="list-style-type: none"> • Latitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -90.0 to 90.0 • Positive values indicate northern latitude • Negative values indicate southern latitude • Bit to check in ParamPresenceMask - 16

<i>pLongitude</i>	<ul style="list-style-type: none"> • Longitude (specified in WGS84 datum) • Type - Floating point • Units - Degrees • Range - -180.0 to 180.0 • Positive values indicate eastern latitude • Negative values indicate western latitude • Bit to check in ParamPresenceMask - 17
<i>pHorUncCircular</i>	<ul style="list-style-type: none"> • Horizontal position uncertainty. • Units - Meters • Bit to check in ParamPresenceMask - 18
<i>pHorUncEllipse-SemiMinor</i>	<ul style="list-style-type: none"> • Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters • Bit to check in ParamPresenceMask - 19
<i>pHorUncEllipse-SemiMajor</i>	<ul style="list-style-type: none"> • Semi-major axis of horizontal elliptical uncertainty. • Units: Meters • Bit to check in ParamPresenceMask - 20
<i>pHorUncEllipse-OrientAzimuth</i>	<ul style="list-style-type: none"> • Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180 • Bit to check in ParamPresenceMask - 21
<i>pHorConfidence</i>	<ul style="list-style-type: none"> • Horizontal uncertainty confidence. • If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty. • Units - Percentage • Range 0-99 • Bit to check in ParamPresenceMask - 22
<i>pHorReliability</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Location reliability is not set. – 1 - Location reliability is very low; use it at your own risk – 2 - Location reliability is low; little or no cross-checking is possible. – 3 - Location reliability is medium; limited cross-check passed – 4 - Location reliability is high; strong cross-check passed

- Bit to check in ParamPresenceMask - **23**

Parameters

<i>pSpeed-Horizontal</i>	<ul style="list-style-type: none"> • Horizontal speed. • Units - Meters/second • Bit to check in ParamPresenceMask - 24
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> • 3-D Speed uncertainty. • Units - Meters/second. • Bit to check in ParamPresenceMask - 25
<i>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> • Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883 • Bit to check in ParamPresenceMask - 26
<i>pAltitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> • Altitude With Respect to Sea Level. • Units - Meters • Bit to check in ParamPresenceMask - 27
<i>pVertUnc</i>	<ul style="list-style-type: none"> • Vertical uncertainty. • Units - Meters • Bit to check in ParamPresenceMask - 28
<i>pVertConfidence</i>	<ul style="list-style-type: none"> • Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99 • Bit to check in ParamPresenceMask - 29
<i>pVertReliability</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Location reliability is not set. – 1 - Location reliability is very low; use it at your own risk. – 2 - Location reliability is low; little or no cross-checking is possible – 3 - Location reliability is medium; limited cross-check passed – 4 - Location reliability is high; strong cross-check passed – Bit to check in ParamPresenceMask - 30
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> • Vertical speed. • Units - Meters/second • Bit to check in ParamPresenceMask - 31

<i>pHeading</i>	<ul style="list-style-type: none"> • Heading. • Units - Degree • Range 0 to 359.999 • Bit to check in ParamPresenceMask - 32
<i>pHeadingUnc</i>	<ul style="list-style-type: none"> • Heading uncertainty. • Units - Degree • Range 0 to 359.999 • Bit to check in ParamPresenceMask - 33
<i>pMagnetic-Deviation</i>	<ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 34
<i>pTechnology-Mask</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0x00000001 - Satellites were used to generate the fix – 0x00000002 - Cell towers were used to generate the fix – 0x00000004 - Wi-Fi access points were used to generate the fix – 0x00000008 - Sensors were used to generate the fix – 0x00000010 - Reference Location was used to generate the fix – 0x00000020 - Coarse position injected into the location engine was used to generate the fix – 0x00000040 - AFLT was used to generate the fix – 0x00000080 - GNSS and network-provided measurements were used to generate the fix • Bit to check in ParamPresenceMask - 35
<i>pPrecision-Dilution</i>	<ul style="list-style-type: none"> • See loc_precisionDilution for more information • Bit to check in ParamPresenceMask - 36
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> • UTC timestamp • Units - Milliseconds since Jan. 1, 1970 • Bit to check in ParamPresenceMask - 37
<i>pLeapSeconds</i>	<ul style="list-style-type: none"> • Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds. • Units - Seconds • Bit to check in ParamPresenceMask - 38
<i>pGpsTime</i>	<ul style="list-style-type: none"> • See loc_gpsTime for more information • Bit to check in ParamPresenceMask - 39

<i>pTimeUnc</i>	<ul style="list-style-type: none"> • Time uncertainty. • Units - Milliseconds • Bit to check in ParamPresenceMask - 40
<i>pTimeSrc</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.

- Bit to check in ParamPresenceMask - **41**

Parameters

<i>pSensorData-Usage</i>	<ul style="list-style-type: none"> • See loc_sensorDataUsage for more information • Bit to check in ParamPresenceMask - 42
<i>pFixId</i>	<ul style="list-style-type: none"> • Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session. • Bit to check in ParamPresenceMask - 43
<i>pSvUsedforFix</i>	<ul style="list-style-type: none"> • See loc_svUsedforFix for more information • Bit to check in ParamPresenceMask - 44
<i>pAltitude-Assumed</i>	<ul style="list-style-type: none"> • Indicates whether altitude is assumed or calculated. • Value <ul style="list-style-type: none"> – 0x00 - Altitude is calculated – 0x01 - Altitude is assumed • Bit to check in ParamPresenceMask - 45

8.775.2 Field Documentation

- 8.775.2.1 `uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed`
- 8.775.2.2 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid`
- 8.775.2.3 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel`
- 8.775.2.4 `swi_uint256_t unpack_loc_PositionRpt_Ind_t::ParamPresenceMask`
- 8.775.2.5 `uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId`
- 8.775.2.6 `loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime`
- 8.775.2.7 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading`
- 8.775.2.8 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc`
- 8.775.2.9 `uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence`
- 8.775.2.10 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability`
- 8.775.2.11 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular`
- 8.775.2.12 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.775.2.13 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor`
- 8.775.2.14 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor`
- 8.775.2.15 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude`
- 8.775.2.16 `uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds`
- 8.775.2.17 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude`
- 8.775.2.18 `uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation`
- 8.775.2.19 `loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution`
- 8.775.2.20 `loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage`
- 8.775.2.21 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal`
- 8.775.2.22 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc`
- 8.775.2.23 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical`
- 8.775.2.24 `loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix`
- 8.775.2.25 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask`
- 8.775.2.26 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc`
- 8.775.2.27 `uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc`

- 8.775.2.28 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc`
- 8.775.2.29 `uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence`
- 8.775.2.30 `uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability`
- 8.775.2.31 `uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc`
- 8.775.2.32 `uint8_t unpack_loc_PositionRpt_Ind_t::sessionId`
- 8.775.2.33 `uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus`
- 8.775.2.34 `uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult`

8.776 `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](#)

8.776.1 Detailed Description

This structure contains LOC Event Sensor Streaming Ready Status

Parameters

<code>-pAccelAcceptReady[OPTIONAL]</code>	<ul style="list-style-type: none"> • See loc_accelAcceptReady for more information • Bit to check in ParamPresenceMask - 16
<code>-pGyroAcceptReady[OPTIONAL]</code>	<ul style="list-style-type: none"> • See loc_gyroAcceptReady for more information • Bit to check in ParamPresenceMask - 17
<code>-pAccelTempAcceptReady[OPTIONAL]</code>	<ul style="list-style-type: none"> • See loc_accelTempAcceptReady for more information • Bit to check in ParamPresenceMask - 18
<code>-pGyroTempAcceptReady[OPTIONAL]</code>	<ul style="list-style-type: none"> • See loc_gyroTempAcceptReady for more information • Bit to check in ParamPresenceMask - 19

8.776.2 Field Documentation

- 8.776.2.1 `loc_accelAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pAccelAcceptReady`
- 8.776.2.2 `loc_accelTempAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pAccelTempAcceptReady`

8.776.2.3 `swi_uint256_t` `unpack_loc_SensorStreamingCallback_Ind_t::ParamPresenceMask`

8.776.2.4 `loc_gyroAcceptReady*` `unpack_loc_SensorStreamingCallback_Ind_t::pGyroAcceptReady`

8.776.2.5 `loc_gyroTempAcceptReady*` `unpack_loc_SensorStreamingCallback_Ind_t::pGyroTempAcceptReady`

8.777 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

Data Fields

- `uint32_t` [status](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.777.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • Valid values <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.777.2 Field Documentation

8.777.2.1 `swi_uint256_t` `unpack_loc_SetExtPowerConfig_Ind_t::ParamPresenceMask`

8.777.2.2 `uint32_t` `unpack_loc_SetExtPowerConfig_Ind_t::status`

8.777.2.3 `uint16_t` `unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult`

8.778 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

- [swi_uint256_t ParamPresenceMask](#)

8.778.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result. • Bit to check in ParamPresenceMask - 2
------------------	---

8.778.2 Field Documentation

8.778.2.1 [swi_uint256_t unpack_loc_SetExtPowerState_t::ParamPresenceMask](#)

8.778.2.2 [uint16_t unpack_loc_SetExtPowerState_t::Tlvresult](#)

8.779 unpack_loc_SetOperationMode_Ind_t Struct Reference

Data Fields

- [uint32_t status](#)
- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.779.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • Status of the Set Operation Mode request. • Valid values: <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure – 2 - Request failed because it is not supported – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - Request failed because the engine could not allocate sufficient memory for the request – 9 - Request failed because the maximum number of Geofences are already programmed – 10 - Location service failed because of an XTRA version-based file format check failure • Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.779.2 Field Documentation

8.779.2.1 `swi_uint256_t unpack_loc_SetOperationMode_Ind_t::ParamPresenceMask`

8.779.2.2 `uint32_t unpack_loc_SetOperationMode_Ind_t::status`

8.779.2.3 `uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult`

8.780 unpack_loc_SetOperationMode_t Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.780.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.• Bit to check in ParamPresenceMask - 2
------------------	--

8.780.2 Field Documentation

8.780.2.1 `swi_uint256_t unpack_loc_SetOperationMode_t::ParamPresenceMask`

8.780.2.2 `uint16_t unpack_loc_SetOperationMode_t::Tlvresult`

8.781 unpack_loc_SetServer_Ind_t Struct Reference

Data Fields

- `uint32_t serverStatus`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.781.1 Detailed Description

Contain the parameters passed for SetLocSetServerCallback by the device.

Parameters

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

8.781.2 Field Documentation

8.781.2.1 `swi_uint256_t` `unpack_loc_SetServer_Ind_t::ParamPresenceMask`8.781.2.2 `uint32_t` `unpack_loc_SetServer_Ind_t::serverStatus`8.781.2.3 `uint16_t` `unpack_loc_SetServer_Ind_t::Tlvresult`8.782 `unpack_loc_SLQSLOCGetBestAvailPos_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.782.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result. • Bit to check in ParamPresenceMask - 2
------------------	---

8.782.2 Field Documentation

8.782.2.1 `swi_uint256_t unpack_loc_SLQSLOCGetBestAvailPos_t::ParamPresenceMask`

8.782.2.2 `uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult`

8.783 unpack_loc_SLQSLOCGetOpMode_t Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.783.1 Detailed Description

This structure contains Start LOC unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.• Bit to check in ParamPresenceMask - 2
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.783.2 Field Documentation

8.783.2.1 `swi_uint256_t unpack_loc_SLQSLOCGetOpMode_t::ParamPresenceMask`

8.783.2.2 `uint16_t unpack_loc_SLQSLOCGetOpMode_t::Tlvresult`

8.784 unpack_loc_Start_t Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.784.1 Detailed Description

This structure contains Start LOC unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.• Bit to check in ParamPresenceMask - 2
------------------	--

8.784.2 Field Documentation

8.784.2.1 `swi_uint256_t unpack_loc_Start_t::ParamPresenceMask`

8.784.2.2 `uint16_t unpack_loc_Start_t::Tlvresult`

8.785 `unpack_loc_Stop_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.785.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result. • Bit to check in ParamPresenceMask - 2
------------------	---

8.785.2 Field Documentation

8.785.2.1 `swi_uint256_t unpack_loc_Stop_t::ParamPresenceMask`

8.785.2.2 `uint16_t unpack_loc_Stop_t::Tlvresult`

8.786 `unpack_nas_GetACCOLC_t` Struct Reference

Data Fields

- `uint8_t * pAccolc`
- `swi_uint256_t ParamPresenceMask`

8.786.1 Detailed Description

Retrieves information about the access overload class unpack structure

Parameters

<i>pAccolc</i>	<ul style="list-style-type: none"> • An 8-bit integer representation of the ACCOLC. <ul style="list-style-type: none"> – Range: 0 to 15 (0x00 to 0x0F). • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.786.2 Field Documentation

8.786.2.1 `uint8_t*` `unpack_nas_GetACCOLC_t::pAccolc`

8.786.2.2 `swi_uint256_t` `unpack_nas_GetACCOLC_t::ParamPresenceMask`

8.787 unpack_nas_GetANAAAAuthenticationStatus_t Struct Reference

Data Fields

- `uint32_t *` `pAuthStatus`
- `swi_uint256_t` `ParamPresenceMask`

8.787.1 Detailed Description

AN-AAA authentication status of the device

Parameters

<i>AN-AAA</i>	Authentication Status. <ul style="list-style-type: none"> • Status of the last AN-AAA authentication request, if any, for the current 1xEV-DO session. <ul style="list-style-type: none"> – 0 - AAA_STATUS_FAILED - Authentication failed – 1 - AAA_STATUS_SUCCESS - Authentication success – 2 - AAA_STATUS_NO_REQUEST - No authentication requested • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.787.2 Field Documentation

8.787.2.1 `swi_uint256_t` `unpack_nas_GetANAAAAuthenticationStatus_t::ParamPresenceMask`

8.787.2.2 `uint32_t*` `unpack_nas_GetANAAAAuthenticationStatus_t::pAuthStatus`

8.788 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`

8.788.1 Detailed Description

Gets the current CDMA network parameters

Parameters

<i>SCI</i>	<ul style="list-style-type: none"> • Slot cycle index • Bit to check in ParamPresenceMask - 17
<i>SCM</i>	<ul style="list-style-type: none"> • Station class mark • Bit to check in ParamPresenceMask - 18
<i>RegHomeSID</i>	<ul style="list-style-type: none"> • Register on home SID <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 19
<i>RegForeignSID</i>	<ul style="list-style-type: none"> • Register on foreign SID <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 19
<i>RegForeignNID</i>	<ul style="list-style-type: none"> • Register on foreign NID <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 19
<i>ForceRev0</i>	<ul style="list-style-type: none"> • Force CDMA 1x-EV-DO Rev. 0 mode <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 20
<i>CustomSCP</i>	<ul style="list-style-type: none"> • Use a custom config for CDMA 1x-EV-DO SCP <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 21

<i>Protocol</i>	<ul style="list-style-type: none"> • Protocol mask for custom SCP config <ul style="list-style-type: none"> – 0x00000001 - Subtype 2 Physical Layer – 0x00000002 - Enhanced CCMAC – 0x00000004 - Enhanced ACMAC – 0x00000008 - Enhanced FTCMAC – 0x00000010 - Subtype 3 RTCMAC – 0x00000020 - Subsystem 1 RTCMAC – 0x00000040 - Enhanced Idle – 0x00000080 - Generic Multimode Capable Disc Port • Bit to check in ParamPresenceMask - 21
<i>Broadcast</i>	<ul style="list-style-type: none"> • Broadcast mask for custom SCP config <ul style="list-style-type: none"> – 0x00000001 - Generic broadcast enabled • Bit to check in ParamPresenceMask - 21
<i>Application</i>	<ul style="list-style-type: none"> • Application mask for custom SCP config <ul style="list-style-type: none"> – 0x00000001 - SN Multiflow Packet Application – 0x00000002 - SN Enhanced Multiflow Packet Application • Bit to check in ParamPresenceMask - 21
<i>Roaming</i>	<ul style="list-style-type: none"> • Roaming preference <ul style="list-style-type: none"> – 0 - Automatic – 1 - Home Only – 2 - Affiliated Roaming Only – 3 - Home and Affiliated Roaming • Bit to check in ParamPresenceMask - 22
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.788.2 Field Documentation

8.788.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.788.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.788.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.788.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.788.2.5 swi_uint256_t unpack_nas_GetCDMANetworkParameters_t::ParamPresenceMask

8.788.2.6 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.788.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

8.788.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

8.788.2.9 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

8.788.2.10 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

8.788.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

8.788.2.12 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

8.789 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

8.789.1 Detailed Description

Structure for HomeNetwork3GPP2 unpack.

Parameters

<i>pMCC</i>	<ul style="list-style-type: none"> • Mobile country code (UMTS only). • Bit to check in ParamPresenceMask - 1
<i>pMNC</i>	<ul style="list-style-type: none"> • Mobile network code (UMTS only). • Bit to check in ParamPresenceMask - 1
<i>nameSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that network name array can contain (UMTS only). • Bit to check in ParamPresenceMask - 1
<i>pName</i>	<ul style="list-style-type: none"> • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only). • Bit to check in ParamPresenceMask - 1

<i>pSID</i>	<ul style="list-style-type: none"> • Home network system ID <ul style="list-style-type: none"> – 0xFFFF - Unknown. – Only applies to cdma2000 • Bit to check in ParamPresenceMask - 16
<i>pNID</i>	<ul style="list-style-type: none"> • Home network ID <ul style="list-style-type: none"> – 0xFFFF - Unknown. – Only applies to cdma2000 • Bit to check in ParamPresenceMask - 16
<i>pNw2MCC</i>	<ul style="list-style-type: none"> • Mobile country code (3GPP2 only). • Range : 0 to 999 • Bit to check in ParamPresenceMask - 17
<i>pNw2MNC</i>	<ul style="list-style-type: none"> • Mobile network code (3GPP2 only). • Range : 0 to 999 • Bit to check in ParamPresenceMask - 17
<i>pNw2DescDisp</i>	<ul style="list-style-type: none"> • Network Name Display (3GPP2 only). -Valid Value <ul style="list-style-type: none"> – 0x00 - Do not display – 0x01 - Display – 0xFF - Unknown • Bit to check in ParamPresenceMask - 17
<i>pNw2DescEnc</i>	<ul style="list-style-type: none"> • Encoding of the network description (3GPP2 only). • Valid Value <ul style="list-style-type: none"> – 0x00 - Octet, unspecified – 0x02 - 7-bit ASCII (liteqmi_helper_decode7bitAsciiEncString) – 0x04 - Unicode – 0x09 - GSM 7-bit default • Bit to check in ParamPresenceMask - 17
<i>pNw2DescLen</i>	<ul style="list-style-type: none"> • Network Description Length (3GPP2 only). • Bit to check in ParamPresenceMask - 17
<i>pNw2Name</i>	<ul style="list-style-type: none"> • Network Name (3GPP2 only). • Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.789.2 Field Documentation

8.789.2.1 `uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize`

8.789.2.2 `swi_uint256_t unpack_nas_GetHomeNetwork3GPP2_t::ParamPresenceMask`

8.789.2.3 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC`

8.789.2.4 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC`

8.789.2.5 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName`

8.789.2.6 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID`

8.789.2.7 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp`

8.789.2.8 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc`

8.789.2.9 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen`

8.789.2.10 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC`

8.789.2.11 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC`

8.789.2.12 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name`

8.789.2.13 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID`

8.790 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`

8.790.1 Detailed Description

This structure contains unpack get get home network parameters.

Parameters

<i>mcc</i>	<ul style="list-style-type: none"> • Mobile country code (UMTS only). • A 16-bit integer representation of MCC. Range: 0 to 999. • Bit to check in ParamPresenceMask - 1
<i>mnc</i>	<ul style="list-style-type: none"> • Mobile network code (UMTS only). • A 16-bit integer representation of MNC. Range: 0 to 999. • Bit to check in ParamPresenceMask - 1

<i>name</i>	<ul style="list-style-type: none"> • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only). • Bit to check in ParamPresenceMask - 1
<i>sid</i>	<ul style="list-style-type: none"> • Home network system ID <ul style="list-style-type: none"> – 0xFFFF - Unknown. – Only applies to cdma2000 • Bit to check in ParamPresenceMask - 16
<i>nid</i>	<ul style="list-style-type: none"> • Home network ID <ul style="list-style-type: none"> – 0xFFFF - Unknown. – Only applies to cdma2000 • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.790.2 Field Documentation

8.790.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

8.790.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

8.790.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

8.790.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

8.790.2.5 swi_uint256_t unpack_nas_GetHomeNetwork_t::ParamPresenceMask

8.790.2.6 uint16_t unpack_nas_GetHomeNetwork_t::sid

8.791 unpack_nas_GetNetworkPreference_t Struct Reference

Data Fields

- uint32_t [ActiveTechPref](#)
- uint32_t [Duration](#)
- uint32_t [PersistentTechPref](#)
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.791.1 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"> • Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: <ul style="list-style-type: none"> – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: <ul style="list-style-type: none"> – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved • Bit to check in ParamPresenceMask - 1
<i>Duration[OUT]</i>	<ul style="list-style-type: none"> • Duration of active preference <ul style="list-style-type: none"> – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call • Bit to check in ParamPresenceMask - 1
<i>Persistent-TechnologyPref[OUT]</i>	<ul style="list-style-type: none"> • Bit field representing persistent radio technology preference <ul style="list-style-type: none"> – Same representation as the pTechnologyPref parameter • Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.791.2 Field Documentation

8.791.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.791.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.791.2.3 swi_uint256_t unpack_nas_GetNetworkPreference_t::ParamPresenceMask

8.791.2.4 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.791.2.5 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.792 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

- [uint8_t instancesSize](#)
- [RFBandInfoElements](#) [RFBandInfoElements](#) [255]
- [swi_uint256_t ParamPresenceMask](#)

8.792.1 Detailed Description

Structur to store the RFInfoList

Parameters

<i>instanceSize</i>	<ul style="list-style-type: none"> • Upon input, maximum number of elements that the RF info instances array can contain. • Upon successful output, actual number of elements in RF info instances array.
<i>RFBandInfo-Elements</i>	<ul style="list-style-type: none"> • RF info instances array <ul style="list-style-type: none"> – See RFBandInfoElements for more information • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.792.2 Field Documentation

8.792.2.1 [uint8_t](#) `unpack_nas_GetRFInfo_t::instancesSize`

8.792.2.2 [swi_uint256_t](#) `unpack_nas_GetRFInfo_t::ParamPresenceMask`

8.792.2.3 [RFBandInfoElements](#) `unpack_nas_GetRFInfo_t::RFBandInfoElements[255]`

8.793 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](#)

8.793.1 Detailed Description

This structure contains unpack get get serving network parameters.

Parameters

<i>Registration-State</i>	<ul style="list-style-type: none"> Registration state: <ul style="list-style-type: none"> 0 - Not registered 1 - Registered 2 - Searching/Not Registered 3 - Registration Denied 4 - Unknown Bit to check in ParamPresenceMask - 1
<i>CSDomain</i>	<ul style="list-style-type: none"> Circuit switch domain status: <ul style="list-style-type: none"> 0 - Unknown/Not Applicable 1 - Attached 2 - Detached Bit to check in ParamPresenceMask - 1
<i>PSDomain</i>	<ul style="list-style-type: none"> Packet switch domain status <ul style="list-style-type: none"> 0 - Unknown/Not Applicable 1 - Attached 2 - Detached Bit to check in ParamPresenceMask - 1
<i>RAN</i>	<ul style="list-style-type: none"> Type of radio access network on which mobile is registered: <ul style="list-style-type: none"> 0 - Unknown 1 - cdma2000 network 2 - UMTS network Bit to check in ParamPresenceMask - 1
<i>RadiolfacesSize</i>	<ul style="list-style-type: none"> Upon input, maximum number of elements that the radio interface array contain. Upon successful output, actual number of elements in the radio interface array. Bit to check in ParamPresenceMask - 1
<i>Radiolfaces</i>	<ul style="list-style-type: none"> An array of Radio Interface Technology <ul style="list-style-type: none"> See qaGobiApiTableRadioInterfaces.h for the Radio Interface Technologies Bit to check in ParamPresenceMask - 1
<i>Roaming</i>	<ul style="list-style-type: none"> Roaming indicator Bit to check in ParamPresenceMask - 16
<i>MCC</i>	<ul style="list-style-type: none"> Mobile country code Bit to check in ParamPresenceMask - 18

<i>MNC</i>	<ul style="list-style-type: none"> • Mobile network code • Bit to check in ParamPresenceMask - 18
<i>nameSize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks • Bit to check in ParamPresenceMask - 18
<i>Name</i>	<ul style="list-style-type: none"> • Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks • Bit to check in ParamPresenceMask - 18
<i>DataCapsLen[IN/OUT]</i>	<ul style="list-style-type: none"> • Upon input, the maximum number of elements the data capabilities array can contain. • Upon output, the actual number of elements in the data capabilities array. • Bit to check in ParamPresenceMask - 17
<i>DataCaps[OUT]</i>	<ul style="list-style-type: none"> • Data capabilities array of unsigned long type <ul style="list-style-type: none"> – 1 - GPRS – 2 - EDGE – 3 - HSDPA – 4 - HSUPA – 5 - WCDMA – 6 - CDMA 1xRTT – 7 - CDMA 1xEV-DO Rev 0 – 8 - CDMA 1xEV-DO Rev. A – 9 - GSM – 10 - EVDO Rev. B – 11 - LTE – 12 - HSDPA Plus – 13 - Dual Carrier HSDPA Plus • Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.793.2 Field Documentation

8.793.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.793.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.793.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.793.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.793.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

- 8.793.2.6 `uint8_t unpack_nas_GetServingNetwork_t::Name[255]`
- 8.793.2.7 `uint8_t unpack_nas_GetServingNetwork_t::nameSize`
- 8.793.2.8 `swi_uint256_t unpack_nas_GetServingNetwork_t::ParamPresenceMask`
- 8.793.2.9 `uint32_t unpack_nas_GetServingNetwork_t::PSDomain`
- 8.793.2.10 `uint8_t unpack_nas_GetServingNetwork_t::Radiolfaces[255]`
- 8.793.2.11 `uint8_t unpack_nas_GetServingNetwork_t::RadiolfacesSize`
- 8.793.2.12 `uint32_t unpack_nas_GetServingNetwork_t::RAN`
- 8.793.2.13 `uint32_t unpack_nas_GetServingNetwork_t::RegistrationState`
- 8.793.2.14 `uint32_t unpack_nas_GetServingNetwork_t::Roaming`

8.794 `unpack_nas_GetServingNetworkCapabilities_t` Struct Reference

Data Fields

- `uint8_t DataCapsLen`
- `uint8_t DataCaps [255]`
- `swi_uint256_t ParamPresenceMask`

8.794.1 Detailed Description

This structure contains unpack get serving network capabilities parameters.

Parameters

<i>DataCapsLen</i>	<ul style="list-style-type: none"> • Upon input, the maximum number of elements the data capabilities array can contain. • Upon output, the actual number of elements in the data capabilities array. • Bit to check in ParamPresenceMask - 17
<i>DataCaps[OUT]</i>	<ul style="list-style-type: none"> • Data capabilities array of unsigned long type <ul style="list-style-type: none"> – 1 - GPRS – 2 - EDGE – 3 - HSDPA – 4 - HSUPA – 5 - WCDMA – 6 - CDMA 1xRTT – 7 - CDMA 1xEV-DO Rev 0 – 8 - CDMA 1xEV-DO Rev. A – 9 - GSM – 10 - EVDO Rev. B – 11 - LTE – 12 - HSDPA Plus – 13 - Dual Carrier HSDPA Plus • Bit to check in ParamPresenceMask - 17

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.794.2 Field Documentation

8.794.2.1 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]`

8.794.2.2 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen`

8.794.2.3 `swi_uint256_t unpack_nas_GetServingNetworkCapabilities_t::ParamPresenceMask`

8.795 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

- `uint32_t len`
- signed char `rsi` [8]
- `uint32_t radio` [8]
- `swi_uint256_t ParamPresenceMask`

8.795.1 Detailed Description

This structure contains unpack get signal strengths parameters.

Parameters

<i>len</i>	<ul style="list-style-type: none"> number of rssi & radio items following
<i>rsi</i>	<ul style="list-style-type: none"> signal strength array Bit to check in ParamPresenceMask - 16
<i>radio</i>	<ul style="list-style-type: none"> radio interface array Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.795.2 Field Documentation

8.795.2.1 `uint32_t unpack_nas_GetSignalStrengths_t::len`

8.795.2.2 `swi_uint256_t unpack_nas_GetSignalStrengths_t::ParamPresenceMask`

8.795.2.3 `uint32_t unpack_nas_GetSignalStrengths_t::radio[8]`

8.795.2.4 `signed char unpack_nas_GetSignalStrengths_t::rsi[8]`

8.796 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](#)

8.796.1 Detailed Description

Contain the network scan information.

Parameters

<i>p3GppNetwork-InstanceSize</i>	<ul style="list-style-type: none"> • Upon input, maximum number of elements that the network info instance array can contain. • Upon successful output, the actual number of elements in the network info instance array. • Bit to check in ParamPresenceMask - 16
<i>p3GppNetwork-InfoInstances</i>	<ul style="list-style-type: none"> • Network info instance array <ul style="list-style-type: none"> – See nas_QmiNas3GppNetworkInfo for more information • Bit to check in ParamPresenceMask - 16
<i>pRATInstance-Size</i>	<ul style="list-style-type: none"> • Upon input, maximum number of elements that the RAT info instance array can contain. • Upon successful output, the actual number of elements in the RAT info instance array. • Bit to check in ParamPresenceMask - 17
<i>pRATInstance</i>	<ul style="list-style-type: none"> • RAT info instance array <ul style="list-style-type: none"> – See nas_QmiNas3GppNetworkRAT for more information • Bit to check in ParamPresenceMask - 17
<i>pPCSInstance-Size</i>	<ul style="list-style-type: none"> • Upon input, maximum number of elements that the PCS Digit info instance array can contain. • Upon successful output, the actual number of elements in the PCS Digit info instance array. • Bit to check in ParamPresenceMask - 18
<i>pPCSInstance</i>	<ul style="list-style-type: none"> • PCS Digit info instance array <ul style="list-style-type: none"> – See nas_QmisNasPcsDigit for more information • Bit to check in ParamPresenceMask - 18

<i>pScanResult</i>	<ul style="list-style-type: none"> • status of network scan • 0x00 - scan successful • 0x01 - scan was aborted • 0x02 - scan did not complete due to a radio link failure recovery in progress • Bit to check in ParamPresenceMask - 19
<i>pPCIInfo</i>	<ul style="list-style-type: none"> • PCI Information <ul style="list-style-type: none"> – See nas_QmisNasSlqsNasPCIInfo for more information • Bit to check in ParamPresenceMask - 23
<i>pLteOpModeTlv[OUT]</i>	<ul style="list-style-type: none"> • LTE Operational Mode. • See nas_LteOpModeTlv for more information. • Bit to check in ParamPresenceMask - 26
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.796.2 Field Documentation

8.796.2.1 [nas_QmiNas3GppNetworkInfo](#)* [unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances](#)

8.796.2.2 [uint8_t](#)* [unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize](#)

8.796.2.3 [swi_uint256_t](#) [unpack_nas_PerformNetworkScan_t::ParamPresenceMask](#)

8.796.2.4 [nas_LteOpModeTlv](#)* [unpack_nas_PerformNetworkScan_t::pLteOpModeTlv](#)

8.796.2.5 [nas_QmisNasSlqsNasPCIInfo](#)* [unpack_nas_PerformNetworkScan_t::pPCIInfo](#)

8.796.2.6 [nas_QmisNasPcsDigit](#)* [unpack_nas_PerformNetworkScan_t::pPCSInstance](#)

8.796.2.7 [uint8_t](#)* [unpack_nas_PerformNetworkScan_t::pPCSInstanceSize](#)

8.796.2.8 [nas_QmiNas3GppNetworkRAT](#)* [unpack_nas_PerformNetworkScan_t::pRATInstance](#)

8.796.2.9 [uint8_t](#)* [unpack_nas_PerformNetworkScan_t::pRATInstanceSize](#)

8.796.2.10 [uint32_t](#)* [unpack_nas_PerformNetworkScan_t::pScanResult](#)

8.797 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

Data Fields

- [uint8_t dataCapsSize](#)
- [uint8_t dataCaps](#) [255]
- [swi_uint256_t ParamPresenceMask](#)

8.797.1 Detailed Description

This structure contains unpack set data capabilities callback indication parameters.

Parameters

<i>dataCapsSize</i>	<ul style="list-style-type: none"> • Number of Data Capabilities • Bit to check in ParamPresenceMask - 17
<i>dataCaps</i>	<ul style="list-style-type: none"> • Data Capabilities • Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.797.2 Field Documentation

8.797.2.1 `uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]`

8.797.2.2 `uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize`

8.797.2.3 `swi_uint256_t unpack_nas_SetDataCapabilitiesCallback_ind_t::ParamPresenceMask`

8.798 unpack_nas_SetEventReportInd_t Struct Reference

Data Fields

- [nas_SignalStrengthTlv SSTlv](#)
- [nas_RFInfoTlv RFTlv](#)
- [nas_RejectReasonTlv RRTlv](#)
- [nas_SLQSSignalStrengthsTlv SLQSSSTlv](#)
- [swi_uint256_t ParamPresenceMask](#)

8.798.1 Detailed Description

This structure contains unpack set event report indication parameters.

Parameters

<i>SSTlv</i>	<ul style="list-style-type: none"> • signal strength tlv • Bit to check in ParamPresenceMask - 16
<i>RFTlv</i>	<ul style="list-style-type: none"> • RF tlv • Bit to check in ParamPresenceMask - 17
<i>RRTlv</i>	<ul style="list-style-type: none"> • RR tlv • Bit to check in ParamPresenceMask - 18

<i>SLQSSSTlv</i>	<ul style="list-style-type: none"> • signal strength complete info tlv • Bit to check in ParamPresenceMask - 19 • Bit to check in ParamPresenceMask - 20 • Bit to check in ParamPresenceMask - 21 • Bit to check in ParamPresenceMask - 22 • Bit to check in ParamPresenceMask - 23 • Bit to check in ParamPresenceMask - 24 • Bit to check in ParamPresenceMask - 25 • Bit to check in ParamPresenceMask - 26
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.798.2 Field Documentation

8.798.2.1 `swi_uint256_t unpack_nas_SetEventReportInd_t::ParamPresenceMask`

8.798.2.2 `nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv`

8.798.2.3 `nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv`

8.798.2.4 `nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv`

8.798.2.5 `nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv`

8.799 unpack_nas_SetNasLTECphyCalndCallback_ind_t Struct Reference

Data Fields

- [nas_PhyCaAggScellIndType](#) `sPhyCaAggScellIndType`
- [nas_PhyCaAggScellIDIBw](#) `sPhyCaAggScellIDIBw`
- [nas_PhyCaAggScellInfo](#) `sPhyCaAggScellInfo`
- [nas_PhyCaAggPcellInfo](#) `sPhyCaAggPcellInfo`
- [nas_PhyCaAggScellIndex](#) `sPhyCaAggScellIndex`
- [swi_uint256_t](#) `ParamPresenceMask`

8.799.1 Detailed Description

Structure for storing the LTE PHY CA indication parameters.

Parameters

<i>pPhyCaAggScellIndType</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellIndType for more information. • Bit to check in ParamPresenceMask - 1
<i>sPhyCaAggScellIDIBw</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellIDIBw for more information. • Bit to check in ParamPresenceMask - 16

<i>sPhyCaAgg-ScellInfo</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellInfo for more information. • Bit to check in ParamPresenceMask - 17
<i>sPhyCaAgg-PcellInfo</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggPcellInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>sPhyCaAgg-ScellIndex</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellIndex for more information. • Bit to check in ParamPresenceMask - 19
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.799.2 Field Documentation

8.799.2.1 `swi_uint256_t` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::ParamPresenceMask`

8.799.2.2 `nas_PhyCaAggPcellInfo` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggPcellInfo`

8.799.2.3 `nas_PhyCaAggScellDIBw` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellDIBw`

8.799.2.4 `nas_PhyCaAggScellIndex` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndex`

8.799.2.5 `nas_PhyCaAggScellIndType` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndType`

8.799.2.6 `nas_PhyCaAggScellInfo` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellInfo`

8.800 `unpack_nas_SetNetworkPreference_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.800.1 Detailed Description

This structure contains unpack set network preference parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.800.2 Field Documentation

8.800.2.1 `swi_uint256_t` unpack_nas_SetNetworkPreference_t::ParamPresenceMask

8.800.2.2 `uint16_t` unpack_nas_SetNetworkPreference_t::Tlvresult

8.801 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- `uint8_t` [roaming](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.801.1 Detailed Description

This structure contains unpack set roaming indicator callback indication parameters.

Parameters

<i>roaming</i>	<ul style="list-style-type: none">• Roaming Indication<ul style="list-style-type: none">– 0 - Roaming– 1 - Home– 2 - Roaming partner– >2 - Operator defined values• Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.801.2 Field Documentation

8.801.2.1 `swi_uint256_t` unpack_nas_SetRoamingIndicatorCallback_ind_t::ParamPresenceMask

8.801.2.2 `uint8_t` unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.802 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

- [NAServingSystemInfo](#) SSInfo
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.802.1 Detailed Description

This structure contains unpack set serving system callback indication parameters.

Parameters

<i>SSInfo</i>	<ul style="list-style-type: none"> Serving system parameters information <ul style="list-style-type: none"> See NASServingSystemInfo for more details Bit to check in ParamPresenceMask - 1
<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack result Bit to check in ParamPresenceMask - 2
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.802.2 Field Documentation

8.802.2.1 `swi_uint256_t` `unpack_nas_SetServingSystemCallback_ind_t::ParamPresenceMask`8.802.2.2 `NASServingSystemInfo` `unpack_nas_SetServingSystemCallback_ind_t::SSInfo`8.802.2.3 `uint16_t` `unpack_nas_SetServingSystemCallback_ind_t::Tlvresult`8.803 `unpack_nas_SLQSGetErrorRate_t` Struct Reference

Data Fields

- `uint16_t *` [pCDMAFrameErrRate](#)
- `uint16_t *` [pHDRPackErrRate](#)
- `uint8_t *` [pGSMBER](#)
- `uint8_t *` [pWCDMABER](#)
- `swi_uint256_t` `ParamPresenceMask`

8.803.1 Detailed Description

Structure for Error Rate unpack parameters.

Parameters

<i>pCDMAFrameErrRate</i>	<ul style="list-style-type: none"> CDMA Frame Error Rate 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%. A value of 0xFFFF indicates that the error rate is unknown/unavailable. Bit to check in ParamPresenceMask - 16
<i>pHDRPackErrRate</i>	<ul style="list-style-type: none"> HDR Packet Error Rate 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%. A value of 0xFFFF indicates that the error rate is unknown/unavailable. Bit to check in ParamPresenceMask - 17

<i>pGSMBER</i>	<ul style="list-style-type: none"> • GSM Bit Error Rate • Valid error rate values between 1 and 100 are returned to indicate the percentage value. • A 0% block error rate (BLER) indicates No Data. • Bit to check in ParamPresenceMask - 18
<i>pWCDMABER</i>	<ul style="list-style-type: none"> • WCDMA Block Error Rate • Valid error rate values between 1 and 100 are returned to indicate the percentage value. • A value of 0xFF indicates that the error rate is unknown/unavailable. • Bit to check in ParamPresenceMask - 19
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.803.2 Field Documentation

8.803.2.1 `swi_uint256_t unpack_nas_SLQSGetErrorRate_t::ParamPresenceMask`

8.803.2.2 `uint16_t* unpack_nas_SLQSGetErrorRate_t::pCDMAFrameErrRate`

8.803.2.3 `uint8_t* unpack_nas_SLQSGetErrorRate_t::pGSMBER`

8.803.2.4 `uint16_t* unpack_nas_SLQSGetErrorRate_t::pHDRPackErrRate`

8.803.2.5 `uint8_t* unpack_nas_SLQSGetErrorRate_t::pWCDMABER`

8.804 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- [NasGetLTECphyCAInfo LTECphyCAInfo](#)
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.804.1 Detailed Description

This structure contains unpack get carrier aggregation event information parameters.

Parameters

<i>LTECphyCa</i>	<ul style="list-style-type: none"> • Carrier aggregation event information <ul style="list-style-type: none"> – See NasGetLTECphyCAInfo for more details
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result • Bit to check in ParamPresenceMask - 2
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.804.2 Field Documentation

8.804.2.1 `NasGetLTECphyCAInfo unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCAInfo`

8.804.2.2 `swi_uint256_t unpack_nas_SlqsGetLTECphyCAInfo_t::ParamPresenceMask`

8.804.2.3 `uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult`

8.805 `unpack_nas_SLQSGetNetworkTime_t` Struct Reference

Data Fields

- [nas_timeInfo](#) * [p3GPP2TimeInfo](#)
- [nas_timeInfo](#) * [p3GPPTimeInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.805.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

Parameters

<i>p3GPP2Time-Info</i>	[Optional] • See nas_timeInfo for more information
------------------------	---

- * - Bit to check in ParamPresenceMask - 16

Parameters

<i>p3GPPTimeInfo</i>	[Optional] – See nas_timeInfo for more information
----------------------	---

- * - Bit to check in ParamPresenceMask - 17

Parameters

<i>ParamPresence-Mask</i>	– bitmask representation to indicate valid parameters.
---------------------------	--

8.805.2 Field Documentation

8.805.2.1 `nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo`

8.805.2.2 `nas_timeInfo* unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo`

8.805.2.3 `swi_uint256_t unpack_nas_SLQSGetNetworkTime_t::ParamPresenceMask`

8.806 `unpack_nas_SLQSGetOperatorNameData_t` Struct Reference

Data Fields

- [nas_serviceProviderName](#) * [pSvcProviderName](#)
- [nas_operatorPLMNList](#) * [pOperatorPLMNList](#)

- [nas_PLMNNetworkName](#) * [pPLMNNetworkName](#)
- [nas_operatorNameString](#) * [pOperatorNameString](#)
- [nas_PLMNNetworkNameData](#) * [pNITZInformation](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.806.1 Detailed Description

Structure for Operator Name unpack.

Parameters

<i>pSvcProvider- Name</i>	<ul style="list-style-type: none"> • Refer nas_serviceProviderName for details (Optional). • Can provide NULL if this parameter is not required. • Bit to check in ParamPresenceMask - 16
<i>pOperatorPLM- NList</i>	<ul style="list-style-type: none"> • Refer nas_operatorPLMNList for details (Optional). • Can provide NULL if this parameter is not required. • Bit to check in ParamPresenceMask - 17
<i>pPLMNNetwork- Name</i>	<ul style="list-style-type: none"> • Refer nas_PLMNNetworkName for details (Optional). • Can provide NULL if this parameter is not required. • Bit to check in ParamPresenceMask - 18
<i>pOperatorName- String</i>	<ul style="list-style-type: none"> • Refer nas_operatorNameString for details (Optional). • Can provide NULL if this parameter is not required. • Bit to check in ParamPresenceMask - 19
<i>pNITZ- Information</i>	<ul style="list-style-type: none"> • Refer nas_PLMNNetworkNameData for details (Optional). • Can provide NULL if this parameter is not required. • Bit to check in ParamPresenceMask - 20
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.806.2 Field Documentation

8.806.2.1 [swi_uint256_t](#) [unpack_nas_SLQSGetOperatorNameData_t::ParamPresenceMask](#)

8.806.2.2 [nas_PLMNNetworkNameData*](#) [unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation](#)

8.806.2.3 [nas_operatorNameString*](#) [unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString](#)

8.806.2.4 [nas_operatorPLMNList*](#) [unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList](#)

8.806.2.5 [nas_PLMNNetworkName*](#) [unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName](#)

8.806.2.6 [nas_serviceProviderName*](#) [unpack_nas_SLQSGetOperatorNameData_t::pSvcProviderName](#)

8.807 unpack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint8_t [spnEncoding](#)
- uint8_t [spnLength](#)
- char [spn](#) [255]
- uint8_t [shortNameEn](#)
- uint8_t [shortNameCl](#)
- uint8_t [shortNameSB](#)
- char [shortNameLen](#)
- uint8_t [shortName](#) [255]
- uint8_t [longNameEn](#)
- uint8_t [longNameCl](#)
- uint8_t [longNameSB](#)
- uint8_t [longNameLen](#)
- char [longName](#) [255]
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.807.1 Detailed Description

This structure contains unpack get operator name for specified network parameters.

Parameters

<i>spnEncoding</i>	<ul style="list-style-type: none"> • Coding scheme used for service provider name. This value is ignored if spn_len is zero Values: <ul style="list-style-type: none"> – 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0 – 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038 – Note: This value is ignored if spnLength is zero. • Bit to check in ParamPresenceMask - 16
<i>spnLength</i>	<ul style="list-style-type: none"> • Length of SPN which follows • Note: This SPN value is ignored if spnLength is zero and spnEncoding is 0xff. • Bit to check in ParamPresenceMask - 16
<i>spn</i>	<ul style="list-style-type: none"> • Service Provider name string • Bit to check in ParamPresenceMask - 16
<i>shortNameEn</i>	<ul style="list-style-type: none"> • 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"> – 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0 – 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038 – Note: This value is ignored if shortNameLen is zero. • Bit to check in ParamPresenceMask - 16

<i>shortNameCI</i>	<ul style="list-style-type: none"> 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"> 0x00 - Do not add the letters for the countrys initials to the name 0x01 - Add the countrys initials and a text string to the name 0xFF - Not specified Note: This value is ignored if shortNameLen is zero. Bit to check in ParamPresenceMask - 16
<i>shortNameSB</i>	<ul style="list-style-type: none"> PLMN short name spare bits. This value is ignored if shortNameLen is zero. Values: <ul style="list-style-type: none"> 0x01 - Bit 8 is spare and set to 0 in octet 0x02 - Bits 7 and 8 are spare and set to 0 in octet n 0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n 0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n 0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n 0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n 0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n 0x00 - Carries no information about the number of spare bits in octet n Note: This value is ignored if shortNameLen is zero. Bit to check in ParamPresenceMask - 16
<i>shortNameLen</i>	<ul style="list-style-type: none"> Length of shortName which follows Note: This shortName value is ignored if shortNameLen is zero and shortNameEn is 0xff. Bit to check in ParamPresenceMask - 16
<i>shortName</i>	<ul style="list-style-type: none"> PLMN short name Bit to check in ParamPresenceMask - 16
<i>longNameEn</i>	<ul style="list-style-type: none"> 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"> 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038 Note: This value is ignored if shortNameLen is zero. Bit to check in ParamPresenceMask - 16
<i>longNameCI</i>	<ul style="list-style-type: none"> 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"> 0x00 - Do not add the letters for the countrys initials to the name 0x01 - Add the countrys initials and a text string to the name 0xFF - Not specified Note: This value is ignored if shortNameLen is zero. Bit to check in ParamPresenceMask - 16

<i>longNameSB</i>	<ul style="list-style-type: none"> • PLMN long name spare bits. This value is ignored if longNameLen is zero. Values: <ul style="list-style-type: none"> – 0x01 - Bit 8 is spare and set to 0 in octet – 0x02 - Bits 7 and 8 are spare and set to 0 in octet n – 0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n – 0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n – 0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n – 0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n – 0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n – 0x00 - Carries no information about the number of spare bits in octet n – Note: This value is ignored if shortNameLen is zero. • Bit to check in ParamPresenceMask - 16
<i>longNameLen</i>	<ul style="list-style-type: none"> • Length of longName which follows • Note: This longName value is ignored if longNameLen is zero and longNameEn is 0xff. • Bit to check in ParamPresenceMask - 16
<i>longName</i>	<ul style="list-style-type: none"> • PLMN long name • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.807.2 Field Documentation

8.807.2.1 char unpack_nas_SLQSGetPLMNName_t::longName[255]

8.807.2.2 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameCI

8.807.2.3 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameEn

8.807.2.4 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameLen

8.807.2.5 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameSB

8.807.2.6 swi_uint256_t unpack_nas_SLQSGetPLMNName_t::ParamPresenceMask

8.807.2.7 uint8_t unpack_nas_SLQSGetPLMNName_t::shortName[255]

8.807.2.8 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameCI

8.807.2.9 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameEn

8.807.2.10 char unpack_nas_SLQSGetPLMNName_t::shortNameLen

8.807.2.11 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameSB

8.807.2.12 char unpack_nas_SLQSGetPLMNName_t::spn[255]

8.807.2.13 uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding

8.807.2.14 uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength

8.808 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_CDMASysInfoExt](#) CDMASystemInfoExt
- uint8_t HdrPersonality
- uint16_t TrackAreaCode
- [nas_callBarStatus](#) CallBarStatus
- [swi_uint256_t](#) ParamPresenceMask

8.808.1 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">• Serving System• See nas_servSystem for more information• Bit to check in ParamPresenceMask - 1
----------------------	--

<i>RoamIndicator-Val</i>	<ul style="list-style-type: none"> • Optional parameter indicating Roaming Indicator value • Values: <ul style="list-style-type: none"> – 0x00 - Roaming – 0x01 - Home – 0x02 - Flashing – 0x03 and above - Operator defined values • Bit to check in ParamPresenceMask - 16
<i>DataSrv-Capabilities</i>	<ul style="list-style-type: none"> • Optional parameter indicating Data services capability • See nas_dataSrvCapabilities for more information • Bit to check in ParamPresenceMask - 17
<i>CurrentPLMN</i>	<ul style="list-style-type: none"> • Optional parameter indicating Current PLMN • See nas_currentPLMN for more information • Bit to check in ParamPresenceMask - 18
<i>SystemID</i>	<ul style="list-style-type: none"> • Optional parameter indicating System ID • Bit to check in ParamPresenceMask - 19
<i>NetworkID</i>	<ul style="list-style-type: none"> • Optional parameter indicating Network ID • Bit to check in ParamPresenceMask - 19
<i>BaseStationID</i>	<ul style="list-style-type: none"> • Optional parameter indicating Base Station Identification Number • Bit to check in ParamPresenceMask - 20
<i>BaseStation-Latitude</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 20
<i>Basestation-Longitude</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 20
<i>Roaming-IndicatorList</i>	<ul style="list-style-type: none"> • Optional parameter indicating Roaming Indicator List • See nas_roamIndList for more information • Bit to check in ParamPresenceMask - 21

<i>DefaultRoamInd</i>	<ul style="list-style-type: none"> • Optional parameter indicating Default Roaming Indicator • Values: <ul style="list-style-type: none"> – 0x00 - Roaming – 0x01 - Home • Bit to check in ParamPresenceMask - 22
<i>Gpp2TimeZone</i>	<ul style="list-style-type: none"> • Optional parameter indicating 3GPP2 Time Zone • See nas_qaQmi3Gpp2TimeZone for more information • Bit to check in ParamPresenceMask - 23
<i>CDMA_P_Rev</i>	<ul style="list-style-type: none"> • Optional parameter indicating CDMA P_Rev in use • Bit to check in ParamPresenceMask - 24
<i>GppTimeZone</i>	<ul style="list-style-type: none"> • Optional parameter indicating Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min. (signed value). • Bit to check in ParamPresenceMask - 26
<i>GppNetworkDS- TAdjustment</i>	<ul style="list-style-type: none"> • Optional parameter indicating 3GPP network daylight saving adjustment • Values: <ul style="list-style-type: none"> – 0x00 - No adjustment for Daylight Saving Time – 0x01 - 1 hr adjustment for Daylight Saving Time – 0x02 - 2 hr adjustment for Daylight Saving Time • Bit to check in ParamPresenceMask - 27
<i>Lac</i>	<ul style="list-style-type: none"> • Optional parameter indicating 3GPP Location Area Code • Bit to check in ParamPresenceMask - 28
<i>CellID</i>	<ul style="list-style-type: none"> • Optional parameter indicating 3GPP Cell ID • Bit to check in ParamPresenceMask - 29
<i>ConcSvcInfo</i>	<ul style="list-style-type: none"> • Optional parameter indicating 3GPP2 concurrent service Info • Values: <ul style="list-style-type: none"> – 0x00 - Concurrent service not available – 0x01 - Concurrent service available • Bit to check in ParamPresenceMask - 30
<i>PRLInd</i>	<ul style="list-style-type: none"> • Optional parameter indicating 3GPP2 PRL Indicator • Values: <ul style="list-style-type: none"> – 0x00 - System not in PRL – 0x01 - System is in PRL • Bit to check in ParamPresenceMask - 31

<i>DTMInd</i>	<ul style="list-style-type: none"> • Optional parameter indicating Dual Transfer Mode Indication(GSM Only) • Values: <ul style="list-style-type: none"> – 0x00 - DTM not supported – 0x01 - DTM supported • Bit to check in ParamPresenceMask - 32
<i>DetailedSvcInfo</i>	<ul style="list-style-type: none"> • Optional parameter indicating Detailed service information • See nas_detailSvcInfo for more information • Bit to check in ParamPresenceMask - 33
<i>CDMASystem-InfoExt</i>	<ul style="list-style-type: none"> • Optional parameter indicating CDMA System Info Ext • See nas_CDMASysInfoExt for more information • Bit to check in ParamPresenceMask - 34
<i>HdrPersonality</i>	<ul style="list-style-type: none"> • Optional parameter indicating HDR Personality Information • Values: <ul style="list-style-type: none"> – 0x00 - Unknown – 0x01 - HRPD – 0x02 - eHRPD • Bit to check in ParamPresenceMask - 35
<i>TrackAreaCode</i>	<ul style="list-style-type: none"> • Optional parameter indicating Tracking area code information for LTE • Bit to check in ParamPresenceMask - 36
<i>CallBarStatus</i>	<ul style="list-style-type: none"> • Optional parameter indicating Call Barring Status • See nas_callBarStatus for more information • Bit to check in ParamPresenceMask - 37
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.808.2 Field Documentation

8.808.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID

8.808.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude

8.808.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude

8.808.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus

8.808.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev

8.808.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt

- 8.808.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID
- 8.808.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo
- 8.808.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN
- 8.808.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities
- 8.808.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd
- 8.808.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo
- 8.808.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd
- 8.808.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone
- 8.808.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment
- 8.808.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone
- 8.808.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality
- 8.808.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac
- 8.808.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID
- 8.808.2.20 swi_uint256_t unpack_nas_SLQSGetServingSystem_t::ParamPresenceMask
- 8.808.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd
- 8.808.2.22 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal
- 8.808.2.23 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList
- 8.808.2.24 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem
- 8.808.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID
- 8.808.2.26 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

8.809 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](#)

8.809.1 Detailed Description

This structure contains the Signal Strength Information

Parameters

<i>signalStrength-ReqMask</i>	<ul style="list-style-type: none"> • Request Mask <ul style="list-style-type: none"> – 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 Bit 1 - ECIO Information bit Valid values are: 0 - Do Not Request Additional Info for ECIO 1 - Request Additional Info for ECIO Bit 2 - IO Information bit Valid values are: 0 - Do Not Request Additional Info for IO 1 - Request Additional Info for IO Bit 3 - SINR Information bit Valid values are: 0 - Do Not Request Additional Info for SINR 1 - Request Additional Info for SINR 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 Bit 5 - RSRQ Information bit Valid values are: 0 - Do Not Request Additional Info for RSRQ 1 - Request Additional Info for RSRQ 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 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 • Bit to check in ParamPresenceMask - 1
<i>rxSignal-StrengthListLen</i>	<ul style="list-style-type: none"> • Number of elements in Receive Signal Strength List • Bit to check in ParamPresenceMask - 17
<i>rxSignal-StrengthList</i>	<ul style="list-style-type: none"> • See nas_rxSignalStrengthListElement for more information • Bit to check in ParamPresenceMask - 17
<i>ecioListLen</i>	<ul style="list-style-type: none"> • Number of elements in ECIO List • Bit to check in ParamPresenceMask - 18

<i>ecioList</i>	<ul style="list-style-type: none"> • See nas_ecioListElement for more information • Bit to check in ParamPresenceMask - 18
<i>lo</i>	<ul style="list-style-type: none"> • Received lo in dBm; IO is only applicable for 1xEV-DO • Bit to check in ParamPresenceMask - 19
<i>sinr</i>	<ul style="list-style-type: none"> • SINR level <ul style="list-style-type: none"> – 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 • Bit to check in ParamPresenceMask - 20
<i>errorRateListLen</i>	<ul style="list-style-type: none"> • Number of elements in Error Rate List • Bit to check in ParamPresenceMask - 21
<i>errorRateList</i>	<ul style="list-style-type: none"> • See nas_errorRateListElement for more information • Bit to check in ParamPresenceMask - 21
<i>rsrqInfo</i>	<ul style="list-style-type: none"> • See nas_rsrqInformation for more information • Bit to check in ParamPresenceMask - 22
<i>lteSNR</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 23
<i>lteSRP</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 24
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.809.2 Field Documentation

8.809.2.1 `nas_ecioListElement unpack_nas_SLQSGetSignalStrength_t::ecioList[18]`

8.809.2.2 `uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen`

8.809.2.3 `nas_errorRateListElement unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]`

8.809.2.4 `uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen`

- 8.809.2.5 `int32_t unpack_nas_SLQSGetSignalStrength_t::lo`
- 8.809.2.6 `int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp`
- 8.809.2.7 `int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr`
- 8.809.2.8 `swi_uint256_t unpack_nas_SLQSGetSignalStrength_t::ParamPresenceMask`
- 8.809.2.9 `nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo`
- 8.809.2.10 `nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]`
- 8.809.2.11 `uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen`
- 8.809.2.12 `uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask`
- 8.809.2.13 `uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr`

8.810 `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` * `pCampedCiotLteOpMode`
- `swi_uint256_t` `ParamPresenceMask`

8.810.1 Detailed Description

This structure contains unpack get system information parameters.

Parameters

<i>pCDMASrv- StatusInfo</i>	<ul style="list-style-type: none"> • See nas_SrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 16
<i>pHDRSrvStatus- Info</i>	<ul style="list-style-type: none"> • See nas_SrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 17
<i>pGSMSrvStatus- Info</i>	<ul style="list-style-type: none"> • See nas_GSMSrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>pWCDMASrv- StatusInfo</i>	<ul style="list-style-type: none"> • See nas_GSMSrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 19
<i>pLTESrvStatus- Info</i>	<ul style="list-style-type: none"> • See nas_GSMSrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 20
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> • See nas_CDMASysInfo for more information. • Bit to check in ParamPresenceMask - 21
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> • See nas_HDRSysInfo for more information. • Bit to check in ParamPresenceMask - 22
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> • See nas_GSMSysInfo for more information. • Bit to check in ParamPresenceMask - 23
<i>pWCDMASys- Info</i>	<ul style="list-style-type: none"> • See nas_WCDMASysInfo for more information. • Bit to check in ParamPresenceMask - 24
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> • See nas_LTESysInfo for more information. • Bit to check in ParamPresenceMask - 25
<i>pAddCDMASys- Info</i>	<ul style="list-style-type: none"> • See nas_AddCDMASysInfo for more information. • Bit to check in ParamPresenceMask - 26
<i>pAddHDRSys- Info</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used. • Bit to check in ParamPresenceMask - 27

<i>pAddGSM Sys-Info</i>	<ul style="list-style-type: none"> • See nas_AddSysInfo for more information. • Bit to check in ParamPresenceMask - 28
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> • See nas_AddSysInfo for more information. • Bit to check in ParamPresenceMask - 29
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used. • Bit to check in ParamPresenceMask - 30
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See nas_CallBarringSysInfo for more information. • Bit to check in ParamPresenceMask - 31
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See nas_CallBarringSysInfo for more information. • Bit to check in ParamPresenceMask - 32
<i>pLTEVoice-SupportSysInfo</i>	<ul style="list-style-type: none"> • Indicates voice support status on LTE. <ul style="list-style-type: none"> – 0x00 - Voice is not supported – 0x01 - Voice is supported • Bit to check in ParamPresenceMask - 33
<i>pGSMCipher-DomainSysInfo</i>	<ul style="list-style-type: none"> • Ciphering on the service domain. <ul style="list-style-type: none"> – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched • Bit to check in ParamPresenceMask - 34
<i>pWCDMA-CipherDomain-SysInfo</i>	<ul style="list-style-type: none"> • Ciphering on the service domain. <ul style="list-style-type: none"> – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched • Bit to check in ParamPresenceMask - 35

<i>pCampedCiot-LteOpMode</i>	<ul style="list-style-type: none"> • Camped CIOT LTE Operational Mode. • Values <ul style="list-style-type: none"> – NAS_CIOT_SYS_MODE_NO_SRV (0x00) - No service – NAS_CIOT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband – NAS_CIOT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1 – NAS_CIOT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1 • Bit to check in ParamPresenceMask - 73
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.810.2 Field Documentation

8.810.2.1 **nas_AddCDMASysInfo*** unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo

8.810.2.2 **nas_AddSysInfo*** unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo

8.810.2.3 **uint16_t*** unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo

8.810.2.4 **uint16_t*** unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo

8.810.2.5 **nas_AddSysInfo*** unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo

8.810.2.6 **swi_uint256_t** unpack_nas_SLQSGetSysInfo_t::ParamPresenceMask

8.810.2.7 **uint32_t*** unpack_nas_SLQSGetSysInfo_t::pCampedCiotLteOpMode

8.810.2.8 **nas_SrvStatusInfo*** unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo

8.810.2.9 **nas_CDMASysInfo*** unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo

8.810.2.10 **nas_CallBarringSysInfo*** unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo

8.810.2.11 **uint8_t*** unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo

8.810.2.12 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo

8.810.2.13 **nas_GSMSysInfo*** unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo

8.810.2.14 **nas_SrvStatusInfo*** unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo

8.810.2.15 **nas_HDRSysInfo*** unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo

8.810.2.16 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo

8.810.2.17 **nas_LTESysInfo*** unpack_nas_SLQSGetSysInfo_t::pLTESysInfo

8.810.2.18 **uint8_t*** unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo

8.810.2.19 **nas_CallBarringSysInfo*** unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo

8.810.2.20 uint8_t* unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

8.810.2.21 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

8.810.2.22 nas_WCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

8.811 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](#)

8.811.1 Detailed Description

This structure contains unpack get get system selection preferences parameters.

Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none"> • Optional parameter specifying the emergency Mode • Values: <ul style="list-style-type: none"> – 0 - OFF (normal) – 1 - ON (Emergency)
------------------	---

- * - Bit to check in ParamPresenceMask - **16**

Parameters

<i>pModePref</i>	<ul style="list-style-type: none"> – Optional parameter – Bit Mask indicating the radio technology mode preference – Bit values: <ul style="list-style-type: none"> * Bit 0 - cdma2000 1x * Bit 1 - cdma2000 HRPD(1xEV-DO) * Bit 2 - GSM * Bit 3 - UMTS * Bit 4 - LTE
------------------	--

- * - Bit to check in ParamPresenceMask - **17**

Parameters

<i>pBandPref</i>	<ul style="list-style-type: none"> – Optional parameter – Bit mask representing the band preference – Bit values: <ul style="list-style-type: none"> * Bit 0 - Band Class 0, A-System * Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band * Bit 2 - Band Class 1, all blocks * Bit 3 - Band Class 2 place holder * Bit 4 - Band Class 3, A-System * Bit 5 - Band Class 4, all blocks * Bit 6 - Band Class 5, all blocks * Bit 7 - GSM_DCS_1800 band * Bit 8 - GSM Extended GSM (E-GSM) 900 band * Bit 9 - GSM Primary GSM (P-GSM) 900 band * Bit 10 - Band Class 6 * Bit 11 - Band Class 7 * Bit 12 - Band Class 8 * Bit 13 - Band Class 9 * Bit 14 - Band Class 10 * Bit 15 - Band Class 11 * Bit 16 - GSM 450 band * Bit 17 - GSM 480 band * Bit 18 - GSM 750 band * Bit 19 - GSM 850 band * Bit 20 - GSM Railways GSM 900 Band * Bit 21 - GSM PCS 1900 band * Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band * Bit 23 - WCDMA U.S. PCS 1900 band * Bit 24 - WCDMA Europe and China DCS 1800 band * Bit 25 - WCDMA U.S. 1700 band * Bit 26 - WCDMA U.S. 850 band * Bit 27 - WCDMA Japan 800 band * Bit 28 - Band Class 12 * Bit 29 - Band Class 14 * Bit 30 - Reserved * Bit 31 - Band Class 15 * Bit 32 to 47 - Reserved * Bit 48 - WCDMA Europe 2600 band * Bit 49 - WCDMA Europe and Japan 900 band * Bit 50 - WCDMA Japan 1700 band * Bit 51 to 55 - Reserved * Bit 56 - Band Class 16 * Bit 57 - Band Class 17 * Bit 58 - Band Class 18 * Bit 59 - Band Class 19 * Bit 60 to 64 - Reserved
------------------	--

- * - Bit to check in ParamPresenceMask - **18**

Parameters

<i>pPRLPref</i>	<ul style="list-style-type: none">– Optional parameter indicating the CDMA PRL Preference– Values:<ul style="list-style-type: none">* 0x0001 - Acquire available system only on the A side* 0x0002 - Acquire available system only on the B side* 0x3FFF - Acquire any available systems
-----------------	---

- * - Bit to check in ParamPresenceMask - **19**

Parameters

<i>pRoamPref</i>	<ul style="list-style-type: none">– Optional parameter indicating the roaming Preference– Values:<ul style="list-style-type: none">* 0x01 - Acquire only systems for which the roaming indicator is off* 0x02 - Acquire a system as long as its roaming indicator is not off* 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only* 0xFF - Acquire systems, regardless of their roaming indicator
------------------	---

- * - Bit to check in ParamPresenceMask - **20**

Parameters

<i>pLTEBandPref</i>	<ul style="list-style-type: none"> – Optional parameter – Bit mask representing the LTE band preference – Bit Values <ul style="list-style-type: none"> * Bit 0 - E-UTRA Operating Band 1 * Bit 1 - E-UTRA Operating Band 2 * Bit 2 - E-UTRA Operating Band 3 * Bit 3 - E-UTRA Operating Band 4 * Bit 4 - E-UTRA Operating Band 5 * Bit 5 - E-UTRA Operating Band 6 * Bit 6 - E-UTRA Operating Band 7 * Bit 7 - E-UTRA Operating Band 8 * Bit 8 - E-UTRA Operating Band 9 * Bit 9 - E-UTRA Operating Band 10 * Bit 10 - E-UTRA Operating Band 11 * Bit 11 - E-UTRA Operating Band 12 * Bit 12 - E-UTRA Operating Band 13 * Bit 13 - E-UTRA Operating Band 14 * Bit 16 - E-UTRA Operating Band 17 * Bit 17 - E-UTRA Operating Band 18 * Bit 18 - E-UTRA Operating Band 19 * Bit 19 - E-UTRA Operating Band 20 * Bit 20 - E-UTRA Operating Band 21 * Bit 32 - E-UTRA Operating Band 33 * Bit 33 - E-UTRA Operating Band 34 * Bit 34 - E-UTRA Operating Band 35 * Bit 35 - E-UTRA Operating Band 36 * Bit 36 - E-UTRA Operating Band 37 * Bit 37 - E-UTRA Operating Band 38 * Bit 38 - E-UTRA Operating Band 39 * Bit 39 - E-UTRA Operating Band 40 * All other bits are reserved
---------------------	--

- * - Bit to check in ParamPresenceMask - **21**

Parameters

<i>pNetSelPref</i>	<ul style="list-style-type: none"> – Optional parameter indicating network selection preference – Values: <ul style="list-style-type: none"> * 0x00 - Automatic network selection * 0x01 - Manual network selection.
--------------------	---

- * - Bit to check in ParamPresenceMask - **22**

Parameters

<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> – Optional parameter indicating Service domain preference – Values: <ul style="list-style-type: none"> * 0x00 - Circuit switched only * 0x01 - Packet switched only * 0x02 - Circuit switched and packet switched * 0x03 - Packet switched attach * 0x04 - Packet switched detach
-----------------------	--

- * - Bit to check in ParamPresenceMask - 24

Parameters

<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> – Optional parameter indicating GSM/WCDMA Acquisition order Preference – Values: <ul style="list-style-type: none"> * 0x00 - Automatic * 0x01 - GSM then WCDMA * 0x02 - WCDMA then GSM
-------------------------	---

- * - Bit to check in ParamPresenceMask - 25

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> – bitmask representation to indicate valid parameters.
---------------------------	--

8.811.2 Field Documentation

8.811.2.1 `swi_uint256_t unpack_nas_SLQSGetSysSelectionPref_t::ParamPresenceMask`

8.811.2.2 `uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref`

8.811.2.3 `uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode`

8.811.2.4 `uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref`

8.811.2.5 `uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref`

8.811.2.6 `uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref`

8.811.2.7 `uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref`

8.811.2.8 `uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref`

8.811.2.9 `uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref`

8.811.2.10 `uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref`

8.812 `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](#)

8.812.1 Detailed Description

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

Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none"> • Optional parameter specifying the emergency Mode • See nas_EmerModeTlv for more information
<i>pModePref</i>	<ul style="list-style-type: none"> • Optional parameter indicating the radio technology mode preference • See nas_ModePrefTlv for more information
<i>pBandPref</i>	<ul style="list-style-type: none"> • Optional parameter representing the band preference • See nas_BandPrefTlv for more information
<i>pPRLPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating the CDMA PRL Preference • See nas_PRLPrefTlv for more information
<i>pRoamPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating the roaming Preference • See nas_RoamPrefTlv for more information
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> • Optional parameter representing the LTE band preference • See nas_LTEBandPrefTlv for more information
<i>pNetSelPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating network selection preference • See nas_NetSelPrefTlv for more information

<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating Service domain preference • See nas_SrvDomainPrefTlv for more information
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> • Optional parameter indicating GSM/WCDMA Acquisition order Preference • See nas_GWAcqOrderPrefTlv for more information
<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> • Optional parameter • Acquisition Order Preference • See nas_AcqOrderPrefTlv for more information
<i>pRatDisabled-Mask</i>	<ul style="list-style-type: none"> • Optional parameter representing the radio technologies that are disabled. • See nas_RatDisabledMaskTlv for more information
<i>pCiotLteOp-ModePref</i>	<ul style="list-style-type: none"> • Optional parameter • CIOT LTE Operational Mode Preference • See nas_CiotLteOpModePrefTlv for more information
<i>pLteM1BandPref</i>	<ul style="list-style-type: none"> • Optional parameter representing the LTE M1 band preference • See nas_LteM1BandPrefTlv for more information
<i>pLteNb1Band-Pref</i>	<ul style="list-style-type: none"> • Optional parameter representing the LTE NB1 band preference. • See nas_LteNb1BandPrefTlv for more information
<i>pCiotAcqOrder-Pref</i>	<ul style="list-style-type: none"> • Optional parameter indicating CIOT Acquisition Order Preference • See nas_CiotAcqOrderPrefTlv for more information
<i>pNr5gBandPref</i>	<ul style="list-style-type: none"> • Optional parameter indicating NR5G Band Preference • See nas_NR5gBandPrefTlv for more information
<i>pLTEBandPref-Ext</i>	<ul style="list-style-type: none"> • Optional parameter indicating LTE Band Preference Extended • See nas_LTEBandPrefExtTlv for more information

8.812.2 Field Documentation

8.812.2.1 `nas_AcqOrderPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pAcqOrderPref`

8.812.2.2 `nas_BandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pBandPref`

8.812.2.3 `nas_CiotAcqOrderPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pCiotAcqOrderPref`

8.812.2.4 nas_CiotLteOpModePrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pCiotLteOpModePref

8.812.2.5 nas_EmerModeTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pEmerMode

8.812.2.6 nas_GWAcqOrderPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pGWAcqOrderPref

8.812.2.7 nas_LTEBandPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pLTEBandPref

8.812.2.8 nas_LTEBandPrefExtTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pLTEBandPrefExt

8.812.2.9 nas_LteM1BandPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pLteM1BandPref

8.812.2.10 nas_LteNb1BandPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pLteNb1BandPref

8.812.2.11 nas_ModePrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pModePref

8.812.2.12 nas_NetSelPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pNetSelPref

8.812.2.13 nas_NR5gBandPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pNr5gBandPref

8.812.2.14 nas_PRLPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pPRLPref

8.812.2.15 nas_RatDisabledMaskTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pRatDisabledMask

8.812.2.16 nas_RoamPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pRoamPref

8.812.2.17 nas_SrvDomainPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pSrvDomainPref

8.813 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

8.813.1 Detailed Description

Structure for storing the eDRX Change Info indication parameters.

Parameters

<i>edrxEnableType-Tlv</i>	<ul style="list-style-type: none"> • See nas_EdrxEnableType for more information. • Bit to check in ParamPresenceMask - 16
<i>edrxCycleLengthTlv</i>	<ul style="list-style-type: none"> • See nas_EdrxCycleLength for more information. • Bit to check in ParamPresenceMask - 17

<i>edrxPagingTimeWindowTlv</i>	<ul style="list-style-type: none"> • See nas_EdrxPagingTimeWindow for more information. • Bit to check in ParamPresenceMask - 18
<i>edrxRatTypeTlv</i>	<ul style="list-style-type: none"> • See nas_EdrxRatType for more information. • Bit to check in ParamPresenceMask - 19
<i>edrxCiotLteModeTlv</i>	<ul style="list-style-type: none"> • See nas_EdrxCiotLteMode for more information. • Bit to check in ParamPresenceMask - 20
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.813.2 Field Documentation

8.813.2.1 `nas_EdrxCiotLteMode` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxCiotLteModeTlv`

8.813.2.2 `nas_EdrxCycleLength` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxCycleLengthTlv`

8.813.2.3 `nas_EdrxEnableType` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxEnableTypeTlv`

8.813.2.4 `nas_EdrxPagingTimeWindow` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxPagingTimeWindowTlv`

8.813.2.5 `nas_EdrxRatType` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxRatTypeTlv`

8.813.2.6 `swi_uint256_t` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::ParamPresenceMask`

8.814 `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`
- `swi_uint256_t` `ParamPresenceMask`

8.814.1 Detailed Description

Structure for 3GPP2Subscription unpack.

Parameters

<i>pNAMNameInfo</i>	[Optional] <ul style="list-style-type: none"> • See nas_namName for more information • Bit to check in ParamPresenceMask - 16
---------------------	--

<i>pDirNum</i>	[Optional] <ul style="list-style-type: none"> See nas_dirNum for more information Bit to check in ParamPresenceMask - 17
<i>pHomeSIDNID</i>	[Optional] <ul style="list-style-type: none"> See nas_homeSIDNID for more information Bit to check in ParamPresenceMask - 18
<i>pMinBasedIMSI</i>	[Optional] <ul style="list-style-type: none"> See nas_minBasedIMSI for more information Bit to check in ParamPresenceMask - 19
<i>pTrueIMSI</i>	[Optional] <ul style="list-style-type: none"> See nas_trueIMSI for more information Bit to check in ParamPresenceMask - 20
<i>pCDMAChannel</i>	[Optional] <ul style="list-style-type: none"> See nas_CDMAChannel for more information Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.814.2 Field Documentation

8.814.2.1 `swi_uint256_t unpack_nas_SLQSNasGet3GPP2Subscription_t::ParamPresenceMask`

8.814.2.2 `nas_CDMAChannel* unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel`

8.814.2.3 `nas_dirNum* unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum`

8.814.2.4 `nas_homeSIDNID* unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID`

8.814.2.5 `nas_minBasedIMSI* unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI`

8.814.2.6 `nas_namName* unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo`

8.814.2.7 `nas_trueIMSI* unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI`

8.815 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

8.815.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

Parameters

<i>pGERANInfo</i>	<ul style="list-style-type: none"> • See nas_GERANInfo for more information. • Bit to check in ParamPresenceMask - 16
<i>pUMTSInfo</i>	<ul style="list-style-type: none"> • See nas_UMTSInfo for more information. • Bit to check in ParamPresenceMask - 17
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> • See nas_CDMAInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>pLTEInfo-Intrafreq</i>	<ul style="list-style-type: none"> • See nas_LTEInfoIntrafreq for more information. • Bit to check in ParamPresenceMask - 19
<i>pLTEInfo-Interfreq</i>	<ul style="list-style-type: none"> • See nas_LTEInfoInterfreq for more information. • Bit to check in ParamPresenceMask - 20
<i>pLTEInfo-NeighboringGSM</i>	<ul style="list-style-type: none"> • See nas_LTEInfoNeighboringGSM for more information. • Bit to check in ParamPresenceMask - 21
<i>pLTEInfo-NeighboringWCDMA</i>	<ul style="list-style-type: none"> • See nas_LTEInfoNeighboringWCDMA for more information. • Bit to check in ParamPresenceMask - 22
<i>pUMTSCellID</i>	<ul style="list-style-type: none"> • Cell ID. • 0xFFFFFFFF indicates cell ID information is not present. • Bit to check in ParamPresenceMask - 23
<i>pWCDMAInfoLT-ENeighborCell</i>	<ul style="list-style-type: none"> • See nas_WCDMAInfoLTENeighborCell for more information. • Bit to check in ParamPresenceMask - 24
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.815.2 Field Documentation

- 8.815.2.1 swi_uint256_t unpack_nas_SLQSNasGetCellLocationInfo_t::ParamPresenceMask
- 8.815.2.2 nas_CDMAInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo
- 8.815.2.3 nas_GERANInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo
- 8.815.2.4 nas_LTEInfoInterfreq* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq
- 8.815.2.5 nas_LTEInfoIntrafreq* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq
- 8.815.2.6 nas_LTEInfoNeighboringGSM* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM
- 8.815.2.7 nas_LTEInfoNeighboringWCDMA* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA
- 8.815.2.8 uint32_t* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID
- 8.815.2.9 nas_UMTSInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo
- 8.815.2.10 nas_WCDMAInfoLTENeighborCell* unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

8.816 unpack_nas_SLQSNASGeteDRXParams_t Struct Reference

Data Fields

- uint8_t * pCycleLen
- uint8_t * pPagingTimeWindow
- uint8_t * pEdrxEnable
- swi_uint256_t ParamPresenceMask

8.816.1 Detailed Description

This structure contains the SLQSNASGeteDRXParams unpack parameters.

Parameters

<i>pCycleLen</i>	[Optional] <ul style="list-style-type: none"> • eDRX cycle length • Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 • Bit to check in ParamPresenceMask - 16
<i>pPagingTimeWindow</i>	[Optional] <ul style="list-style-type: none"> • eDRX paging time window • Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 • Bit to check in ParamPresenceMask - 17
<i>pEdrxEnable</i>	[Optional] <ul style="list-style-type: none"> • eDRX Enable • Values <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable • Bit to check in ParamPresenceMask - 18

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.816.2 Field Documentation

8.816.2.1 `swi_uint256_t unpack_nas_SLQSNASGeteDRXParams_t::ParamPresenceMask`

8.816.2.2 `uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pCycleLen`

8.816.2.3 `uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pEdrxEnable`

8.816.2.4 `uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pPagingTimeWindow`

8.817 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`

8.817.1 Detailed Description

This structure contains the SLQSNASGeteDRXParams unpack parameters.

Parameters

<i>pCycleLen</i>	[Optional] <ul style="list-style-type: none"> eDRX cycle length Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 Bit to check in ParamPresenceMask - 16
<i>pPagingTimeWindow</i>	[Optional] <ul style="list-style-type: none"> eDRX paging time window Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 Bit to check in ParamPresenceMask - 17
<i>pEdrxEnable</i>	[Optional] <ul style="list-style-type: none"> eDRX Enable Values <ul style="list-style-type: none"> 0x00 - Disable 0x01 - Enable Bit to check in ParamPresenceMask - 18

<i>pEdrxRAT</i>	[Optional] <ul style="list-style-type: none"> • eDRX Radio Access Technology • Values <ul style="list-style-type: none"> – NAS_RADIO_IF_NO_SVC (0x00) - None (no service) – NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X – NAS_RADIO_IF_CDMA_1XEVD0 (0x02) - cdma2000 @ HRPD (1xEV-DO) – NAS_RADIO_IF_AMPS (0x03) - AMPS – NAS_RADIO_IF_GSM (0x04) - GSM – NAS_RADIO_IF_UMTS (0x05) - UMTS – NAS_RADIO_IF_WLAN (0x06) - WLAN – NAS_RADIO_IF_GPS (0x07) - GPS – NAS_RADIO_IF_LTE (0x08) - LTE – NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA – NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1 – NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1 – NAS_RADIO_IF_NO_CHANGE (-1) - No change • Bit to check in ParamPresenceMask - 19
<i>pLteOpMode</i>	[Optional] <ul style="list-style-type: none"> • LTE Operational Mode • Values <ul style="list-style-type: none"> – NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service – NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband – NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1 – NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1 • Bit to check in ParamPresenceMask - 20
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.817.2 Field Documentation

8.817.2.1 swi_uint256_t unpack_nas_SLQSNASGetDRXParamsExt_t::ParamPresenceMask

8.817.2.2 uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pCycleLen

8.817.2.3 uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pEdrxEnable

8.817.2.4 uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pEdrxRAT

8.817.2.5 uint32_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pLteOpMode

8.817.2.6 uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pPagingTimeWindow

8.818 unpack_nas_SLQSNASGetForbiddenNetworks_t Struct Reference

Data Fields

- [nas_ForbiddenNetworks3GPP](#) * [pForbiddenNetworks3GPP](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.818.1 Detailed Description

This structure contains the SLQSNASGetDRXParams request parameters.

Parameters

<i>pForbidden-Networks3GPP</i>	[Optional] <ul style="list-style-type: none"> See nas_ForbiddenNetworks3GPP for more information. Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.818.2 Field Documentation

8.818.2.1 `swi_uint256_t unpack_nas_SLQSNASGetForbiddenNetworks_t::ParamPresenceMask`

8.818.2.2 `nas_ForbiddenNetworks3GPP* unpack_nas_SLQSNASGetForbiddenNetworks_t::pForbiddenNetworks3GPP`

8.819 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference

Data Fields

- `uint8_t * pColorCode`
- `swi_uint256_t ParamPresenceMask`

8.819.1 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"> Color code value Color code corresponding to the sector to which the AT is sending the access probe See 3GPP2 C.S0024-B V3.0, Section 7.11.6.2.1 for more information. <ul style="list-style-type: none"> 0xFF - Not Available Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.819.2 Field Documentation

8.819.2.1 `swi_uint256_t unpack_nas_SLQSNasGetHDRColorCode_t::ParamPresenceMask`

8.819.2.2 `uint8_t* unpack_nas_SLQSNasGetHDRColorCode_t::pColorCode`

8.820 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](#)

8.820.1 Detailed Description

This structure contains the SLQSNasGetRFInfo response parameters.

Parameters

<i>rfbandInfoList</i>	<ul style="list-style-type: none"> • See nas_RfBandInfoList for more information
<i>pRfDedicated-BandInfo</i>	[Optional] <ul style="list-style-type: none"> • See nas_RfDedicatedBandInfo for more information
<i>pRfBandInfoExt-Format</i>	[Optional] <ul style="list-style-type: none"> • See nas_RfBandInfoExtFormat for more information
<i>pRfBandwidth-Info</i>	[Optional] <ul style="list-style-type: none"> • See nas_RfBandwidthInfo for more information
<i>pLTEOperation-Mode</i>	[Optional] <ul style="list-style-type: none"> • See nas_LTEOperationMode for more information

8.820.2 Field Documentation

8.820.2.1 `nas_LTEOperationMode*` `unpack_nas_SLQSNasGetRFInfo_t::pLTEOperationMode`

8.820.2.2 `nas_RfBandInfoExtFormat*` `unpack_nas_SLQSNasGetRFInfo_t::pRfBandInfoExtFormat`

8.820.2.3 `nas_RfBandwidthInfo*` `unpack_nas_SLQSNasGetRFInfo_t::pRfBandwidthInfo`

8.820.2.4 `nas_RfDedicatedBandInfo*` `unpack_nas_SLQSNasGetRFInfo_t::pRfDedicatedBandInfo`

8.820.2.5 `nas_RfBandInfoList` `unpack_nas_SLQSNasGetRFInfo_t::rfbandInfoList`

8.820.2.6 `uint16_t` `unpack_nas_SLQSNasGetRFInfo_t::Tlvresult`

8.821 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

- [cdmaSSInfo](#) `CDMASSInfo`
- [hdrSSInfo](#) `HDRSSInfo`
- `int8_t` [GSMSSInfo](#)
- [cdmaSSInfo](#) `WCDMASSInfo`
- [lteSSInfo](#) `LTESSInfo`

- [swi_uint256_t ParamPresenceMask](#)

8.821.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

Parameters

<i>CDMASSInfo</i>	<ul style="list-style-type: none"> • See cdmaSSInfo for more information • Bit to check in ParamPresenceMask - 16
<i>HDRSSInfo</i>	<ul style="list-style-type: none"> • See hdrSSInfo for more information • Bit to check in ParamPresenceMask - 17
<i>GSMSSInfo</i>	<ul style="list-style-type: none"> • GSM signal strength is the RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal. • Bit to check in ParamPresenceMask - 18
<i>WCDMASSInfo</i>	<ul style="list-style-type: none"> • See cdmaSSInfo for more information • Bit to check in ParamPresenceMask - 19
<i>LTESSInfo</i>	<ul style="list-style-type: none"> • See lteSSInfo for more information • Bit to check in ParamPresenceMask - 20
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.821.2 Field Documentation

8.821.2.1 `cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo`

8.821.2.2 `int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo`

8.821.2.3 `hdrSSInfo unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo`

8.821.2.4 `lteSSInfo unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo`

8.821.2.5 `swi_uint256_t unpack_nas_SLQSNasGetSigInfo_t::ParamPresenceMask`

8.821.2.6 `cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo`

8.822 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference

Data Fields

- `nas_rxInfo` * [pRXChain0Info](#)
- `nas_rxInfo` * [pRXChain1Info](#)

- [nas_txInfo](#) * [pTXInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.822.1 Detailed Description

Structure for TxRx Info unpack

Parameters

<i>pRXChain0Info</i>	[Optional] <ul style="list-style-type: none"> • See nas_rxInfo for more information. • Bit to check in ParamPresenceMask - 16
<i>pRXChain1Info</i>	[Optional] <ul style="list-style-type: none"> • See nas_rxInfo for more information. • Bit to check in ParamPresenceMask - 17
<i>pTXInfo</i>	[Optional] <ul style="list-style-type: none"> • See nas_txInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.822.2 Field Documentation

8.822.2.1 [swi_uint256_t](#) [unpack_nas_SLQSNasGetTxRxInfo_t::ParamPresenceMask](#)

8.822.2.2 [nas_rxInfo*](#) [unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info](#)

8.822.2.3 [nas_rxInfo*](#) [unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info](#)

8.822.2.4 [nas_txInfo*](#) [unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo](#)

8.823 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](#)

8.823.1 Detailed Description

Structure for storing the network reject indication parameters.

Parameters

<i>radioIrf</i>	<ul style="list-style-type: none">• Radio interface from which to get the information. Values:<ul style="list-style-type: none">– 0x04 - NAS_RADIO_IF_GSM - GSM– 0x05 - NAS_RADIO_IF_UMTS - UMTS– 0x08 - NAS_RADIO_IF_LTE - LTE– 0x09 - NAS_RADIO_IF_TDSCDMA - TDSCDMA• Bit to check in ParamPresenceMask - 1
<i>serviceDomain</i>	<ul style="list-style-type: none">• Type of service domain in which the registration is rejected. Values:<ul style="list-style-type: none">– 0x00 - LITE_SYS_SRV_DOMAIN_NO_SRV - No service– 0x01 - LITE_SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only– 0x02 - LITE_SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only– 0x03 - LITE_SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched– 0x04 - LITE_SYS_SRV_DOMAIN_CAMPED - Camped• Bit to check in ParamPresenceMask - 2

<i>rejectCause</i>	<ul style="list-style-type: none"> • Reject cause values. <ul style="list-style-type: none"> – 2 - IMSI unknown in HLR – 3 - Illegal MS – 4 - IMSI unknown in VLR – 5 - IMEI not accepted – 6 - Illegal ME – 7 - EPS services not allowed – 8 - EPS services and non-EPS services not allowed – 9 - UE identity cannot be derived by the network – 10 - Implicitly detached – 11 - PLMN not allowed – 12 - Location Area not allowed – 13 - Roaming not allowed in this location area – 14 - EPS services not allowed in this PLMN – 15 - No Suitable Cells In Location Area – 16 - MSC temporarily not reachable – 17 - Network failure – 18 - CS domain not available – 19 - ESM failure – 20 - MAC failure – 21 - Synch failure – 22 - Congestion – 23 - GSM authentication unacceptable – 24 - Security mode rejected, unspecified – 25 - Not authorized for this CSG – 26 - Non-EPS authentication unacceptable – 32 - Service option not supported – 33 - Requested service option not subscribed – 34 - Service option temporarily out of order – 38 - Call cannot be identified – 48 to 63 - retry upon entry into a new cell – 95 - Semantically incorrect message – 96 - Invalid mandatory information – 97 - Message type non-existent or not implemented – 98 - Message type not compatible with the protocol state – 99 - Information element non-existent or not implemented – 100 - Conditional IE error – 101 - Message not compatible with the protocol state – 111 - Protocol error, unspecified – Note - Any other value received by the mobile station shall be treated as 34, 'Service option temporarily out of order'. – * Any other value received by the network shall be treated as 111, 'Protocol error, unspecified'. <p>See 3GPP TS 24.008, Section 4.4.4.7 and Section 10.5.3.6 See qaGobiApiTableCall-EndReasons.h for Call End reasons</p> <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 3
--------------------	---

<i>pPlmnId</i>	(optional) <ul style="list-style-type: none"> • See nas_PlmnID for more information • Bit to check in ParamPresenceMask - 16
<i>pCsgId</i>	(optional) <ul style="list-style-type: none"> • See nas_CsgId for more information. • Bit to check in ParamPresenceMask - 17
<i>pLteOpMode</i>	(optional) <ul style="list-style-type: none"> • See nas_LteOpMode for more information. • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.823.2 Field Documentation

8.823.2.1 `swi_uint256_t unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::ParamPresenceMask`

8.823.2.2 `nas_CsgId* unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::pCsgId`

8.823.2.3 `nas_LteOpMode* unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::pLteOpMode`

8.823.2.4 `nas_PlmnID* unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::pPlmnId`

8.823.2.5 `uint8_t unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::radioIlf`

8.823.2.6 `uint8_t unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::rejectCause`

8.823.2.7 `LITE_TYPE_OF_SERVICE_DOMAIN unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::serviceDomain`

8.824 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`

8.824.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

Parameters

<i>universalTime</i>	<ul style="list-style-type: none"> • See nas_UniversalTime for more information. • Bit to check in ParamPresenceMask - 1
----------------------	---

<i>pTimeZone</i>	<ul style="list-style-type: none"> • Time Zone. • Offset from Universal time, i.e., the difference between local time and Universal time, in increments of 15 min (signed value). • Bit to check in ParamPresenceMask - 16
<i>pDayltSavAdj</i>	<ul style="list-style-type: none"> • Daylight Saving Adjustment. • Daylight saving adjustment in hr. <ul style="list-style-type: none"> – Possible values: 0, 1, and 2. • Bit to check in ParamPresenceMask - 17
<i>pRadioInterface</i>	<ul style="list-style-type: none"> • Radio interface from which the information comes • Values <ul style="list-style-type: none"> – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.824.2 Field Documentation

8.824.2.1 swi_uint256_t unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::ParamPresenceMask

8.824.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

8.824.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

8.824.2.4 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

8.824.2.5 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

8.825 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](#)

8.825.1 Detailed Description

Structure for storing the nasSigInfo indication parameters.

Parameters

<i>pCDMASigInfo</i>	<ul style="list-style-type: none"> • See cdmaSSInfo for more information. • Bit to check in ParamPresenceMask - 16
<i>pHDRSigInfo</i>	<ul style="list-style-type: none"> • See hdrSSInfo for more information. • Bit to check in ParamPresenceMask - 17
<i>pGSMSigInfo</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 18
<i>pWCDMASigInfo</i>	<ul style="list-style-type: none"> • See cdmaSSInfo for more information. • Bit to check in ParamPresenceMask - 19
<i>pLTESigInfo</i>	<ul style="list-style-type: none"> • See lteSSInfo for more information. • Bit to check in ParamPresenceMask - 20
<i>pRscp</i>	<ul style="list-style-type: none"> • RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. Measurement range: -120 dBm to -25 dBm. • Bit to check in ParamPresenceMask - 21
<i>pTDSCDMASig-InfoExt</i>	<ul style="list-style-type: none"> • See tdscdmaSigInfoExt for more information. • Bit to check in ParamPresenceMask - 22
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.825.2 Field Documentation

8.825.2.1 `swi_uint256_t unpack_nas_SLQSNasSigInfoCallback_ind_t::ParamPresenceMask`

8.825.2.2 `cdmaSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo`

8.825.2.3 `int8_t* unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo`

8.825.2.4 `hdrSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo`

8.825.2.5 `lteSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo`

8.825.2.6 `int8_t* unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp`

8.825.2.7 tdsdmaSigInfoExt* unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt

8.825.2.8 cdmaSSInfo* unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

8.826 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference

Data Fields

- [nas_wcdmaUARFCN](#) * [pWcdmaUARFCN](#)
- [nas_lteEARFCN](#) * [pLteEARFCN](#)
- [nas_ltePCI](#) * [pLtePCI](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.826.1 Detailed Description

Structure for Get Channel Lock unpack.

Parameters

<i>pWcdmaUARFCN</i>	[Optional] <ul style="list-style-type: none"> • See nas_wcdmaUARFCN for more information • Bit to check in ParamPresenceMask - 16
<i>pLteEARFCN</i>	[Optional] <ul style="list-style-type: none"> • See nas_lteEARFCN for more information • Bit to check in ParamPresenceMask - 17
<i>pLtePCI</i>	[Optional] <ul style="list-style-type: none"> • See nas_ltePCI for more information • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.826.2 Field Documentation

8.826.2.1 [swi_uint256_t](#) [unpack_nas_SLQSNASSwiGetChannelLock_t::ParamPresenceMask](#)

8.826.2.2 [nas_lteEARFCN](#)* [unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN](#)

8.826.2.3 [nas_ltePCI](#)* [unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI](#)

8.826.2.4 [nas_wcdmaUARFCN](#)* [unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN](#)

8.827 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

Data Fields

- [nas_CommInfo](#) [commonInfo](#)
- [nas_LTEInfo](#) * [pLTEInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.827.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

Parameters

<i>commonInfo</i>	(mandatory) <ul style="list-style-type: none"> See nas_CommInfo for more information Bit to check in ParamPresenceMask - 1
<i>pLTEInfo</i>	(optional) <ul style="list-style-type: none"> See nas_LTEInfo for more information Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.827.2 Field Documentation

8.827.2.1 `nas_CommInfo` `unpack_nas_SLQSNasSwiModemStatus_t::commonInfo`

8.827.2.2 `swi_uint256_t` `unpack_nas_SLQSNasSwiModemStatus_t::ParamPresenceMask`

8.827.2.3 `nas_LTEInfo*` `unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo`

8.828 `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t` Struct Reference

Data Fields

- [NASQmiCbkJasSwiOTAMessageInd Info](#)
- `uint16_t` `Tlvresult`
- `swi_uint256_t` `ParamPresenceMask`

8.828.1 Detailed Description

This structure contains unpack OTA message indication parameters.

Parameters

<i>Info</i>	<ul style="list-style-type: none"> Structure used to store all QMI Notification Info. <ul style="list-style-type: none"> See NASQmiCbkJasSwiOTAMessageInd for more details
<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack result Bit to check in ParamPresenceMask - 2
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.828.2 Field Documentation

8.828.2.1 NASQmiCbkNasSwiOTAMessageInd unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

8.828.2.2 swi_uint256_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::ParamPresenceMask

8.828.2.3 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult

8.829 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](#)

8.829.1 Detailed Description

Structure for Network Timer indication parameters.

Parameters

<i>t3396_apn</i>	<ul style="list-style-type: none"> • apn • Bit to check in ParamPresenceMask - 16
<i>t3396_plmn_id</i>	<ul style="list-style-type: none"> • plmn id • Bit to check in ParamPresenceMask - 16
<i>t3396_val</i>	<ul style="list-style-type: none"> • timer value • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.829.2 Field Documentation

8.829.2.1 swi_uint256_t unpack_nas_SLQSNasTimerCallback_ind_t::ParamPresenceMask

8.829.2.2 char unpack_nas_SLQSNasTimerCallback_ind_t::t3396_apn[101]

8.829.2.3 uint8_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_plmn_id[3]

8.829.2.4 uint32_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_val

8.830 unpack_nas_SLQSSetSysSelectionPrefCallback_ind_t Struct Reference

Data Fields

- [NASQmiCbKnasSystemSelPrefInd Info](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.830.1 Detailed Description

This structure contains unpack set system selection preference callback indication parameters.

Parameters

<i>Info</i>	<ul style="list-style-type: none"> • Structure used to store all QMI Notification Info. <ul style="list-style-type: none"> – See NASQmiCbKnasSystemSelPrefInd for more details
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result • Bit to check in ParamPresenceMask - 2
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.830.2 Field Documentation

8.830.2.1 `NASQmiCbKnasSystemSelPrefInd unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info`

8.830.2.2 `swi_uint256_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::ParamPresenceMask`

8.830.2.3 `uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult`

8.831 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference

Data Fields

- `uint16_t *` [pCurrentPersonality](#)
- `uint8_t *` [pPersonalityListLength](#)
- `nas_protocolSubtypeElement *` [pProtocolSubtypeElement](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.831.1 Detailed Description

Structure for HDRPersonality unpack.

Parameters

<i>pCurrent-Personality[Out]</i>	<ul style="list-style-type: none"> • Current active personality index. • Bit to check in ParamPresenceMask - 16
----------------------------------	--

<i>pPersonalityListLength</i> [In/Out]	<ul style="list-style-type: none"> Number of Personality Protocol Subtype contains in this response. maximum input value is 3 Bit to check in ParamPresenceMask - 17
<i>pProtocolSubtypeElement</i> [Out]	<ul style="list-style-type: none"> See nas_protocolSubtypeElement for more information. Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.831.2 Field Documentation

8.831.2.1 swi_uint256_t unpack_nas_SLQSSwiGetHDRPersonality_t::ParamPresenceMask

8.831.2.2 uint16_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pCurrentPersonality

8.831.2.3 uint8_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pPersonalityListLength

8.831.2.4 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRPersonality_t::pProtocolSubtypeElement

8.832 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference

Data Fields

- uint16_t * [pCurrentPrsnlty](#)
- uint8_t * [pPersonalityListLength](#)
- [nas_protocolSubtypeElement](#) * [pProtoSubTypElmnt](#)
- uint64_t * [pAppSubType](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.832.1 Detailed Description

Structure for HDRProtSubtype unpack.

Parameters

<i>pCurrentPersonality</i>	<ul style="list-style-type: none"> Current active personality index. Bit to check in ParamPresenceMask - 16
<i>pPersonalityListLength</i>	<ul style="list-style-type: none"> Number of Personality Protocol Subtype contains in this response. maximum input value is 4 Bit to check in ParamPresenceMask - 17
<i>pProtocolSubtypeElement</i>	<ul style="list-style-type: none"> See nas_protocolSubtypeElement for more information. Bit to check in ParamPresenceMask - 17

<i>pAppSubType</i>	<ul style="list-style-type: none"> Stream application subtype Application subtype for each stream Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.832.2 Field Documentation

8.832.2.1 `uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType`

8.832.2.2 `swi_uint256_t unpack_nas_SLQSSwiGetHDRProtSubtype_t::ParamPresenceMask`

8.832.2.3 `uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty`

8.832.2.4 `uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength`

8.832.2.5 `nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypElmnt`

8.833 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference

Data Fields

- [nas_DRCParams](#) * [pDRCParams](#)
- `uint8_t` * [pUATI](#)
- [nas_PilotSetData](#) * [pPilotSetData](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.833.1 Detailed Description

Structure for HRPD Stats unpack.

Parameters

<i>pDRCParams</i>	<ul style="list-style-type: none"> See nas_DRCParams for more information. Bit to check in ParamPresenceMask - 16
<i>pUATI</i>	<ul style="list-style-type: none"> A 128-bit address that includes the access terminal identifier and subnet ID Size must be 16 bytes Bit to check in ParamPresenceMask - 17
<i>pPilotSetData</i>	<ul style="list-style-type: none"> See nas_PilotSetData for more information. Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.833.2 Field Documentation

8.833.2.1 swi_uint256_t unpack_nas_SLQSSwiGetHRPDStats_t::ParamPresenceMask

8.833.2.2 nas_DRCParams* unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams

8.833.2.3 nas_PilotSetData* unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData

8.833.2.4 uint8_t* unpack_nas_SLQSSwiGetHRPDStats_t::pUATI

8.834 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

Data Fields

- uint8_t [ValidityCW0](#)
- uint8_t [CQIValueCW0](#)
- uint8_t [ValidityCW1](#)
- uint8_t [CQIValueCW1](#)
- swi_uint256_t [ParamPresenceMask](#)

8.834.1 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"> • Values <ul style="list-style-type: none"> – 0- Invalid. – 1- Valid. • Bit to check in ParamPresenceMask - 1
<i>CQIValueCW0[OUT]</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – Range 0~15 • Bit to check in ParamPresenceMask - 1
<i>ValidityCW1[OUT]</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0- Invalid. – 1- Valid. • Bit to check in ParamPresenceMask - 1
<i>CQIValueCW1[OUT]</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – Range 0~15 • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.834.2 Field Documentation

8.834.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

8.834.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

8.834.2.3 swi_uint256_t unpack_nas_SLQSSwiGetLteCQI_t::ParamPresenceMask

8.834.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

8.834.2.5 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

8.835 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

Data Fields

- [nas_SccRxInfo](#) * [pSccRxInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.835.1 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"> • See nas_SccRxInfo <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.835.2 Field Documentation

8.835.2.1 swi_uint256_t unpack_nas_SLQSSwiGetLteSccRxInfo_t::ParamPresenceMask

8.835.2.2 nas_SccRxInfo* unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

8.836 unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference

Data Fields

- [nas_HDRPersonality_Ind_Data](#) * [pHDRPersInd](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.836.1 Detailed Description

Structure for HDR Personality indication unpack.

Parameters

<i>pHDRPersInd</i>	<ul style="list-style-type: none"> See nas_HDRPersonality_Ind_Data for more information.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.836.2 Field Documentation

8.836.2.1 swi_uint256_t unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::ParamPresenceMask

8.836.2.2 nas_HDRPersonality_Ind_Data* unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::pHDRPersInd

8.837 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](#)

8.837.1 Detailed Description

Structure for SwiNetworkDebug unpack.

Parameters

<i>pObjectVer</i>	<ul style="list-style-type: none"> Object's version number for the host to handle <ul style="list-style-type: none"> 0xFF - NA Others - shows in decimal Bit to check in ParamPresenceMask - 16
<i>pNetworkStat1x</i>	<ul style="list-style-type: none"> See nas_NetworkStat1x for more information Bit to check in ParamPresenceMask - 17
<i>pNetworkStatEVDO</i>	<ul style="list-style-type: none"> See nas_NetworkStatEVDO for more information. Bit to check in ParamPresenceMask - 18
<i>pDeviceConfigDetail</i>	<ul style="list-style-type: none"> See nas_DeviceConfigDetail for more information. Bit to check in ParamPresenceMask - 19
<i>pDataStatusDetail</i>	<ul style="list-style-type: none"> See nas_DataStatusDetail for more information. Bit to check in ParamPresenceMask - 20

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.837.2 Field Documentation

8.837.2.1 `swi_uint256_t` `unpack_nas_SLQSSwiNetworkDebug_t::ParamPresenceMask`

8.837.2.2 `nas_DataStatusDetail*` `unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail`

8.837.2.3 `nas_DeviceConfigDetail*` `unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail`

8.837.2.4 `nas_NetworkStat1x*` `unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x`

8.837.2.5 `nas_NetworkStatEVDO*` `unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO`

8.837.2.6 `uint8_t*` `unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer`

8.838 `unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t` Struct Reference

Data Fields

- [nas_RankIndicatorTlv](#) `rankIndicatorTlv`
- [swi_uint256_t](#) `ParamPresenceMask`

8.838.1 Detailed Description

Structure for Rank Indicator indication unpack.

Parameters

<i>rankIndicatorTlv</i>	<ul style="list-style-type: none"> See nas_RankIndicatorTlv for more information. Bit to check in <code>ParamPresenceMask</code> - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.838.2 Field Documentation

8.838.2.1 `swi_uint256_t` `unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::ParamPresenceMask`

8.838.2.2 `nas_RankIndicatorTlv` `unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::rankIndicatorTlv`

8.839 `unpack_nas_SLQSSysInfoCallback_ind_t` Struct Reference

Data Fields

- `nas_SrvStatusInfo` * `pCDMASrvStatusInfo`
- `nas_SrvStatusInfo` * `pHDRSrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pGSMSrvStatusInfo`

- [nas_GSMsSrvStatusInfo](#) * [pWCDMASrvStatusInfo](#)
- [nas_GSMsSrvStatusInfo](#) * [pLTESrvStatusInfo](#)
- [nas_CDMASysInfo](#) * [pCDMASysInfo](#)
- [nas_HDRSysInfo](#) * [pHDRSysInfo](#)
- [nas_GSMsSrvStatusInfo](#) * [pGSMsSrvStatusInfo](#)
- [nas_WCDMASysInfo](#) * [pWCDMASysInfo](#)
- [nas_LTESysInfo](#) * [pLTESysInfo](#)
- [nas_AddCDMASysInfo](#) * [pAddCDMASysInfo](#)
- [uint16_t](#) * [pAddHDRSysInfo](#)
- [nas_AddSysInfo](#) * [pAddGSMsSrvStatusInfo](#)
- [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](#)

8.839.1 Detailed Description

This structure contains unpack system information callback indication parameters.

Parameters

<i>pCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> • See nas_SrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 16
<i>pHDRSrvStatusInfo</i>	<ul style="list-style-type: none"> • See nas_SrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 17
<i>pGSMsSrvStatusInfo</i>	<ul style="list-style-type: none"> • See nas_GSMsSrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>pWCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> • See nas_GSMsSrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 19
<i>pLTESrvStatusInfo</i>	<ul style="list-style-type: none"> • See nas_GSMsSrvStatusInfo for more information. • Bit to check in ParamPresenceMask - 20
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> • See nas_CDMASysInfo for more information. • Bit to check in ParamPresenceMask - 21

<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> • See nas_HDRSysInfo for more information. • Bit to check in ParamPresenceMask - 22
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> • See nas_GSMSysInfo for more information. • Bit to check in ParamPresenceMask - 23
<i>pWCDMASys-Info</i>	<ul style="list-style-type: none"> • See nas_WCDMASysInfo for more information. • Bit to check in ParamPresenceMask - 24
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> • See nas_LTESysInfo for more information. • Bit to check in ParamPresenceMask - 25
<i>pAddCDMASys-Info</i>	<ul style="list-style-type: none"> • See nas_AddCDMASysInfo for more information. • Bit to check in ParamPresenceMask - 26
<i>pAddHDRSys-Info</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used. • Bit to check in ParamPresenceMask - 27
<i>pAddGSMSys-Info</i>	<ul style="list-style-type: none"> • See nas_AddSysInfo for more information. • Bit to check in ParamPresenceMask - 28
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> • See nas_AddSysInfo for more information. • Bit to check in ParamPresenceMask - 29
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used. • Bit to check in ParamPresenceMask - 30
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See nas_CallBarringSysInfo for more information. • Bit to check in ParamPresenceMask - 31
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See nas_CallBarringSysInfo for more information. • Bit to check in ParamPresenceMask - 32

<i>pLTEVoice-SupportSysInfo</i>	<ul style="list-style-type: none"> Indicates voice support status on LTE. <ul style="list-style-type: none"> 0x00 - Voice is not supported 0x01 - Voice is supported Bit to check in ParamPresenceMask - 33
<i>pGSMCipher-DomainSysInfo</i>	<ul style="list-style-type: none"> Ciphering on the service domain. <ul style="list-style-type: none"> 0x00 - No service 0x01 - Circuit-switched only 0x02 - Packet-switched only 0x03 - Circuit-switched and packet-switched Bit to check in ParamPresenceMask - 34
<i>pWCDMA-CipherDomain-SysInfo</i>	<ul style="list-style-type: none"> Ciphering on the service domain. <ul style="list-style-type: none"> 0x00 - No service 0x01 - Circuit-switched only 0x02 - Packet-switched only 0x03 - Circuit-switched and packet-switched Bit to check in ParamPresenceMask - 35
<i>pSysInfoNo-Change</i>	<ul style="list-style-type: none"> System Info No Change. 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"> 0x01 - No change in system information Bit to check in ParamPresenceMask - 36
<i>pLteCiotOp-ModeTlv</i>	<ul style="list-style-type: none"> See nas_LteCiotOpModeTlv for more information. Bit to check in ParamPresenceMask - 75
<i>pNR5GSerStat-Tlv</i>	<ul style="list-style-type: none"> See nas_NR5GSerStatTlv for more information. Bit to check in ParamPresenceMask - 76
<i>pNR5GSystem-InfoTlv</i>	<ul style="list-style-type: none"> See nas_NR5GSystemInfoTlv for more information. Bit to check in ParamPresenceMask - 77
<i>pNR5GCell-StatusTlv</i>	<ul style="list-style-type: none"> See nas_NR5GCellStatusTlv for more information. Bit to check in ParamPresenceMask - 78
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.839.2 Field Documentation

- 8.839.2.1 `nas_AddCDMASysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo`
- 8.839.2.2 `nas_AddSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSM SysInfo`
- 8.839.2.3 `uint16_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDR SysInfo`
- 8.839.2.4 `uint16_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTE SysInfo`
- 8.839.2.5 `nas_AddSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMA SysInfo`
- 8.839.2.6 `swi_uint256_t` `unpack_nas_SLQSSysInfoCallback_ind_t::ParamPresenceMask`
- 8.839.2.7 `nas_SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pCDMA SrvStatusInfo`
- 8.839.2.8 `nas_CDMA SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pCDMA SysInfo`
- 8.839.2.9 `nas_CallBarring SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSM CallBarring SysInfo`
- 8.839.2.10 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSM CipherDomain SysInfo`
- 8.839.2.11 `nas_GSM SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSM SrvStatusInfo`
- 8.839.2.12 `nas_GSM SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSM SysInfo`
- 8.839.2.13 `nas_SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pHDR SrvStatusInfo`
- 8.839.2.14 `nas_HDR SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pHDR SysInfo`
- 8.839.2.15 `nas_LteCiotOpModeTlv*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLteCiotOpModeTlv`
- 8.839.2.16 `nas_GSM SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTE SrvStatusInfo`
- 8.839.2.17 `nas_LTE SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTE SysInfo`
- 8.839.2.18 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTE VoiceSupport SysInfo`
- 8.839.2.19 `nas_NR5GCellStatusTlv*` `unpack_nas_SLQSSysInfoCallback_ind_t::pNR5GCellStatus`
- 8.839.2.20 `nas_NR5GSerStatTlv*` `unpack_nas_SLQSSysInfoCallback_ind_t::pNR5GSerStatTlv`
- 8.839.2.21 `nas_NR5GSystemInfoTlv*` `unpack_nas_SLQSSysInfoCallback_ind_t::pNR5GSystemInfoTlv`
- 8.839.2.22 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange`
- 8.839.2.23 `nas_CallBarring SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMA CallBarring SysInfo`
- 8.839.2.24 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMA CipherDomain SysInfo`
- 8.839.2.25 `nas_GSM SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMA SrvStatusInfo`
- 8.839.2.26 `nas_WCDMA SysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMA SysInfo`

8.840 `unpack_omaDmConfigTlv_t` Struct Reference

Data Fields

- uint8_t [state](#)
- uint8_t [userInputReq](#)
- uint16_t [userInputTimeout](#)
- uint16_t [alertmsglength](#)
- uint8_t [alertmsg](#) [256]

8.840.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

Parameters

<i>state</i>	<ul style="list-style-type: none"> • 0x01 - OMA-DM Read Request • 0x02 - OMA-DM Change Request • 0x03 - OMA-DM Config Complete
<i>userInputReq</i>	- Bit mask of available user inputs <ul style="list-style-type: none"> • 0x00 - No user input required. Informational indication • 0x01 - Accept • 0x02 - Reject
<i>userInput-Timeout</i>	<ul style="list-style-type: none"> • Timeout for user input in minutes. A value of 0 means no time-out
<i>alertmsglength</i>	<ul style="list-style-type: none"> • Length of Alert message string in bytes
<i>alertmsg</i>	<ul style="list-style-type: none"> • Alert message in UCS2 (Max 256 characters)

8.840.2 Field Documentation

8.840.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

8.840.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

8.840.2.3 uint8_t unpack_omaDmConfigTlv_t::state

8.840.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

8.840.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

8.841 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](#)

8.841.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

Parameters

<i>state</i>	<ul style="list-style-type: none"> • 0x01 - No Firmware available • 0x02 - Query Firmware Download • 0x03 - Firmware Downloading • 0x04 - Firmware downloaded • 0x05 - Query Firmware Update • 0x06 - Firmware updating • 0x07 - Firmware updated
<i>userInputReq</i>	<ul style="list-style-type: none"> - Bit mask of available user inputs • 0x00 - No user input required. Informational indication • 0x01 - Accept • 0x02 - Reject
<i>userInput-Timeout</i>	<ul style="list-style-type: none"> • Timeout for user input in minutes. A value of 0 means no time-out
<i>fwloadsize</i>	<ul style="list-style-type: none"> • The size (in bytes) of the firmware update package
<i>fwloadComplete</i>	<ul style="list-style-type: none"> • The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.
<i>update-CompleteStatus</i>	<ul style="list-style-type: none"> • This field should be looked at only when the OMADM session is complete. • See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h for update complete status.
<i>severity</i>	<ul style="list-style-type: none"> • 0x01 - Mandatory • 0x02 - Optional
<i>versionlength</i>	<ul style="list-style-type: none"> • Length of FW Version string in bytes
<i>version</i>	<ul style="list-style-type: none"> • FW Version string in ASCII (Max 256 characters)

<i>namelength</i>	<ul style="list-style-type: none"> Length Package Name string in bytes
<i>package_name</i>	<ul style="list-style-type: none"> Package Name in UCS2 (Max 256 characters)
<i>descriptionlength</i>	<ul style="list-style-type: none"> Length of description in bytes
<i>description</i>	<ul style="list-style-type: none"> Description of Update Package in USC2 (Max 256 characters)
<i>sessionType</i>	<ul style="list-style-type: none"> 0x00 - Client initiated 0x01 - Network initiated

8.841.2 Field Documentation

- 8.841.2.1 `uint8_t unpack_omaDmFotaTlv_t::description[256]`
- 8.841.2.2 `uint16_t unpack_omaDmFotaTlv_t::descriptionlength`
- 8.841.2.3 `uint32_t unpack_omaDmFotaTlv_t::fwdloadsize`
- 8.841.2.4 `uint32_t unpack_omaDmFotaTlv_t::fwloadComplete`
- 8.841.2.5 `uint16_t unpack_omaDmFotaTlv_t::namelength`
- 8.841.2.6 `uint8_t unpack_omaDmFotaTlv_t::package_name[256]`
- 8.841.2.7 `uint8_t unpack_omaDmFotaTlv_t::sessionType`
- 8.841.2.8 `uint8_t unpack_omaDmFotaTlv_t::severity`
- 8.841.2.9 `uint8_t unpack_omaDmFotaTlv_t::state`
- 8.841.2.10 `uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus`
- 8.841.2.11 `uint8_t unpack_omaDmFotaTlv_t::userInputReq`
- 8.841.2.12 `uint16_t unpack_omaDmFotaTlv_t::userInputTimeout`
- 8.841.2.13 `uint8_t unpack_omaDmFotaTlv_t::version[256]`
- 8.841.2.14 `uint16_t unpack_omaDmFotaTlv_t::versionlength`

8.842 unpack_omaDmNotificationsTlv_t Struct Reference

Data Fields

- `uint8_t` [notification](#)
- `uint16_t` [sessionStatus](#)

8.842.1 Detailed Description

This structure will hold the SwiOmaDmConfig session notification parameters information.

Parameters

<i>notification</i>	<ul style="list-style-type: none"> • 0x01 - GPS settings change • 0x02 - Device reset • 0x03 - Device factory reset • 0x04 - CI-DC Session start • 0x05 - CI-DC Session end • 0x06 - CI-PRL Session start • 0x07 - CI-PRL Session end • 0x08 - CI-FUMO Session start • 0x09 - CI-FUMO session end • 0x0A - HFA-DC Session start • 0x0B - HFA-DC Session end • 0x0C - HFA-PRL Session start • 0x0D - HFA-PRL Session end • 0x0E - HFA-FUMO Session start • 0x0F - HFA-FUMO session end • 0x10 - NI Session start • 0x11 - NI session end
<i>sessionStatus</i>	<ul style="list-style-type: none"> • This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications • See qaGobiApiTableSwiOMADMSessionStatus.h for session completion code.

8.842.2 Field Documentation

8.842.2.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

8.842.2.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

8.843 unpack_pds_GetPDSDefaults_t Struct Reference

Data Fields

- uint32_t * [pOperation](#)
- uint8_t * [pTimeout](#)
- uint32_t * [pInterval](#)
- uint32_t * [pAccuracy](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.843.1 Detailed Description

Structure contain session configuration parameters.

Parameters

<i>pOperation</i>	<ul style="list-style-type: none"> Current session operating mode <ul style="list-style-type: none"> 0 - Standalone 1 - MS based 2 - MS assisted Bit to check in ParamPresenceMask - 1
<i>pTimeout</i>	<ul style="list-style-type: none"> Maximum amount of time (seconds) to work on each fix, maximum is 255 Bit to check in ParamPresenceMask - 1
<i>pInterval</i>	<ul style="list-style-type: none"> Interval (seconds) between fix requests Bit to check in ParamPresenceMask - 1
<i>pAccuracy</i>	<ul style="list-style-type: none"> Preferred accuracy threshold (meters) Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.843.2 Field Documentation

8.843.2.1 uint32_t* unpack_pds_GetPDSDDefaults_t::pAccuracy

8.843.2.2 swi_uint256_t unpack_pds_GetPDSDDefaults_t::ParamPresenceMask

8.843.2.3 uint32_t* unpack_pds_GetPDSDDefaults_t::pInterval

8.843.2.4 uint32_t* unpack_pds_GetPDSDDefaults_t::pOperation

8.843.2.5 uint8_t* unpack_pds_GetPDSDDefaults_t::pTimeout

8.844 unpack_pds_GetPDSSState_t Struct Reference

Data Fields

- uint32_t * [pEnabledStatus](#)
- uint32_t * [pTrackingStatus](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.844.1 Detailed Description

Structure contain parameters of current PDS state.

Parameters

<i>pEnabledStatus</i>	<ul style="list-style-type: none"> Current PDS state <ul style="list-style-type: none"> 0 - disable 1 - enable Bit to check in ParamPresenceMask - 1
<i>pTrackingStatus</i>	<ul style="list-style-type: none"> Current PDS tracking session state Values: <ul style="list-style-type: none"> 0x00 - Unknown 0x01 - Inactive 0x02 - Active Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.844.2 Field Documentation

8.844.2.1 `swi_uint256_t unpack_pds_GetPDSSState_t::ParamPresenceMask`8.844.2.2 `uint32_t* unpack_pds_GetPDSSState_t::pEnabledStatus`8.844.2.3 `uint32_t* unpack_pds_GetPDSSState_t::pTrackingStatus`8.845 `unpack_pds_GetPortAutomaticTracking_t` Struct Reference

Data Fields

- `uint32_t * pbAuto`
- `swi_uint256_t ParamPresenceMask`

8.845.1 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"> Automatic tracking enabled for NMEA COM port <ul style="list-style-type: none"> 0x00 - Disabled 0x01 - Enabled Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.845.2 Field Documentation

8.845.2.1 `swi_uint256_t` unpack_pds_GetPortAutomaticTracking_t::ParamPresenceMask

8.845.2.2 `uint32_t*` unpack_pds_GetPortAutomaticTracking_t::pbAuto

8.846 unpack_pds_GetServiceAutomaticTracking_t Struct Reference

Data Fields

- `uint32_t *` [pbAuto](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.846.1 Detailed Description

Structure contain the parameter for automatic tracking state.

Parameters

<i>pbAuto</i>	<ul style="list-style-type: none">• Automatic tracking session started for service<ul style="list-style-type: none">– 0x00 - Disabled– 0x01 - Enabled• Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.846.2 Field Documentation

8.846.2.1 `swi_uint256_t` unpack_pds_GetServiceAutomaticTracking_t::ParamPresenceMask

8.846.2.2 `uint32_t*` unpack_pds_GetServiceAutomaticTracking_t::pbAuto

8.847 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference

Data Fields

- `uint32_t *` [pbEnabled](#)
- `uint16_t *` [pInterval](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.847.1 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">• Automatic XTRA download status<ul style="list-style-type: none">– 0 - Disabled– 1 - Enabled• Bit to check in ParamPresenceMask - 16
------------------	--

<i>pInterval</i>	<ul style="list-style-type: none"> Interval (hours) between XTRA downloads Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.847.2 Field Documentation

8.847.2.1 `swi_uint256_t unpack_pds_GetXTRAAutomaticDownload_t::ParamPresenceMask`

8.847.2.2 `uint32_t* unpack_pds_GetXTRAAutomaticDownload_t::pbEnabled`

8.847.2.3 `uint16_t* unpack_pds_GetXTRAAutomaticDownload_t::pInterval`

8.848 unpack_pds_GetXTRANetwork_t Struct Reference

Data Fields

- `uint32_t * pPreference`
- `swi_uint256_t ParamPresenceMask`

8.848.1 Detailed Description

Structure contain the parameter for XTRA WWAN network preference

Parameters

<i>pPreference</i>	<ul style="list-style-type: none"> XTRA WWAN network preference <ul style="list-style-type: none"> 0x00 - None (any available network) 0x01 - Home-only, only when on home systems 0x02 - Roam-only, only when on non-home systems Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.848.2 Field Documentation

8.848.2.1 `swi_uint256_t unpack_pds_GetXTRANetwork_t::ParamPresenceMask`

8.848.2.2 `uint32_t* unpack_pds_GetXTRANetwork_t::pPreference`

8.849 unpack_pds_GetXTRAVality_t Struct Reference

Data Fields

- `uint16_t * pGPSWeek`

- uint16_t * [pGPSWeekOffset](#)
- uint16_t * [pDuration](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.849.1 Detailed Description

Structure contain the parameter for XTRA database validity period

Parameters

<i>pGPSWeek</i>	<ul style="list-style-type: none"> • Starting GPS week of validity period • Bit to check in ParamPresenceMask - 19
<i>pGPSWeek-Offset</i>	<ul style="list-style-type: none"> • Starting GPS week offset (minutes) of validity period • Bit to check in ParamPresenceMask - 19
<i>pDuration</i>	<ul style="list-style-type: none"> • Length of validity period (hours) • NULL pointer - Invalid data. • Bit to check in ParamPresenceMask - 19
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.849.2 Field Documentation

8.849.2.1 [swi_uint256_t](#) [unpack_pds_GetXTRAVValidity_t::ParamPresenceMask](#)

8.849.2.2 [uint16_t*](#) [unpack_pds_GetXTRAVValidity_t::pDuration](#)

8.849.2.3 [uint16_t*](#) [unpack_pds_GetXTRAVValidity_t::pGPSWeek](#)

8.849.2.4 [uint16_t*](#) [unpack_pds_GetXTRAVValidity_t::pGPSWeekOffset](#)

8.850 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](#)

8.850.1 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"> • Position Data NMEA. • String containing the position data in NMEA sentence format (maximum 200 bytes). • Bit to check in ParamPresenceMask - 16
<i>dLongitude</i>	<ul style="list-style-type: none"> • 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. • Units - Decimal degrees • Range - -180 to +180 • Note - Value is in double float format • Bit to check in ParamPresenceMask - 19
<i>dLatitude</i>	<ul style="list-style-type: none"> • Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator. • Units: Decimal degrees • Range: -90 to +90 • Note - Value is in double float format • Bit to check in ParamPresenceMask - 19
<i>session_status</i>	<ul style="list-style-type: none"> • Position Session Status • Status of current session. • Values <ul style="list-style-type: none"> – 0x00 - Success – 0x01 - In progress – 0x02 - General failure – 0x03 - Timeout – 0x04 - User ended the session – 0x05 - Bad parameter – 0x06 - Phone is offline – 0x07 - Engine is locked – 0x08 - E911 session in progress • Bit to check in ParamPresenceMask - 18 • 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.

<i>posSrc</i>	<ul style="list-style-type: none"> • Position Source • 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). • Values <ul style="list-style-type: none"> – 0x00000001 - GPS – 0x00000002 - Cell ID – 0x00000004 - GLONASS – 0x00000008 - Network – 0x00000010 - External position injection • Bit to check in ParamPresenceMask - 28
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.850.2 Field Documentation

- 8.850.2.1 double unpack_pds_SetEventReport_Ind_t::dLatitude
- 8.850.2.2 double unpack_pds_SetEventReport_Ind_t::dLongitude
- 8.850.2.3 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLatitude
- 8.850.2.4 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLongitude
- 8.850.2.5 uint8_t unpack_pds_SetEventReport_Ind_t::has_PositionDataNMEA
- 8.850.2.6 uint8_t unpack_pds_SetEventReport_Ind_t::has_posSrc
- 8.850.2.7 uint8_t unpack_pds_SetEventReport_Ind_t::has_SessionStatus
- 8.850.2.8 swi_uint256_t unpack_pds_SetEventReport_Ind_t::ParamPresenceMask
- 8.850.2.9 char unpack_pds_SetEventReport_Ind_t::PositionDataNMEA[200]
- 8.850.2.10 uint32_t unpack_pds_SetEventReport_Ind_t::posSrc
- 8.850.2.11 uint8_t unpack_pds_SetEventReport_Ind_t::SessionStatus

8.851 unpack_pds_SetPdsState_Ind_t Struct Reference

Data Fields

- uint16_t [TlvPresent](#)
- uint32_t [EnabledStatus](#)
- uint32_t [TrackingStatus](#)
- swi_uint256_t [ParamPresenceMask](#)

8.851.1 Detailed Description

Structure contain the parameter for the Set PDS State Indication.

Parameters

<i>EnabledStatus</i>	<ul style="list-style-type: none"> • GPS service state. • Values <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable • Bit to check in ParamPresenceMask - 1
<i>TrackingStatus</i>	<ul style="list-style-type: none"> • Tracking session state. • Values <ul style="list-style-type: none"> – 0x00 - Unknown – 0x01 - Inactive – 0x02 - Active • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.851.2 Field Documentation

8.851.2.1 uint32_t unpack_pds_SetPdsState_Ind_t::EnabledStatus

8.851.2.2 swi_uint256_t unpack_pds_SetPdsState_Ind_t::ParamPresenceMask

8.851.2.3 uint16_t unpack_pds_SetPdsState_Ind_t::TlvPresent

8.851.2.4 uint32_t unpack_pds_SetPdsState_Ind_t::TrackingStatus

8.852 unpack_pds_SLQSGetAGPSConfig_t Struct Reference

Data Fields

- uint32_t * [pServerAddress](#)
- uint32_t * [pServerPort](#)
- uint8_t * [pServerURL](#)
- uint8_t * [pServerURLLength](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.852.1 Detailed Description

Structure contain the parameter for PDS AGPS (MS-based) configuration unpack.

Parameters

<i>pServerAddress</i>	<ul style="list-style-type: none"> • IPv4 address of AGPS server. "0" if not set • Bit to check in ParamPresenceMask - 16
<i>pServerPort</i>	<ul style="list-style-type: none"> • Port number of AGPS server. "0" if not set • Bit to check in ParamPresenceMask - 16

<i>pServerURL</i>	<ul style="list-style-type: none"> • URL of the AGPS server. "0" if not set • Bit to check in ParamPresenceMask - 17
<i>pServerURL- Length</i>	<ul style="list-style-type: none"> • URL length of AGPS server. "0" if not set • Bit to check in ParamPresenceMask - 17
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.852.2 Field Documentation

8.852.2.1 `swi_uint256_t unpack_pds_SLQSGetAGPSConfig_t::ParamPresenceMask`

8.852.2.2 `uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerAddress`

8.852.2.3 `uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerPort`

8.852.2.4 `uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURL`

8.852.2.5 `uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURLLength`

8.853 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](#)

8.853.1 Detailed Description

Structure contain the parameter for GPS state Info.

Parameters

<i>EngineState</i>	<ul style="list-style-type: none"> • Values: <ul style="list-style-type: none"> – 0 - OFF – 1 - ON • This field is always valid • Bit to check in ParamPresenceMask - 16
<i>ValidMask</i>	<ul style="list-style-type: none"> • Mask of valid state information data. • Values: <ul style="list-style-type: none"> – 0x00000001 - Position(latitude/longitude/horizontal uncertainty) – 0x00000002 - Altitude and vertical uncertainty – 0x00000004 - Time ms – 0x00000008 - Time week number – 0x00000010 - Time uncertainty – 0x00000020 - Iono validity – 0x00000040 - GPS ephemeris – 0x00000080 - GPS almanac – 0x00000100 - GPS health – 0x00000200 - GPS visible SVs – 0x00000400 - GLONASS ephemeris – 0x00000800 - GLONASS almanac – 0x00001000 - GLONASS health – 0x00002000 - GLONASS visible SVs – 0x00004000 - SBAS ephemeris – 0x00008000 - SBAS almanac – 0x00010000 - SBAS health – 0x00020000 - SBAS visible SVs – 0x00040000 - XTRA information • Bit to check in ParamPresenceMask - 16
<i>Latitude</i>	<ul style="list-style-type: none"> • Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator. • Units: Decimal degrees • Range: -90 to +90 degrees. • Value is in double float format (refer to IEEE Std 754-1985) • Bit to check in ParamPresenceMask - 16

<i>Longitude</i>	<ul style="list-style-type: none"> Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of Greenwich meridian. Units: Decimal degrees Range: -180 to +180 degrees Value is in double float format (refer to IEEE Std 754-1985) Bit to check in ParamPresenceMask - 16
<i>Horizontal-Uncertainty</i>	<ul style="list-style-type: none"> Circular horizontal uncertainty (in meters). The uncertainty is provided at 63 percent confidence. Value is in single float format (refer to IEEE Std 754-1985) Bit to check in ParamPresenceMask - 16
<i>Altitude</i>	<ul style="list-style-type: none"> Height above the WGS-84 reference ellipsoid. Value conveys height (in meters) plus 500 m Range -500 to 15883 Value in single float format (refer to IEEE Std 754-1985) Bit to check in ParamPresenceMask - 16
<i>Vertical-Uncertainty</i>	<ul style="list-style-type: none"> Vertical uncertainty (in meters). The uncertainty is provided at 68 percent confidence. Value in single float format (refer to IEEE Std 754-1985) Bit to check in ParamPresenceMask - 16
<i>TimeStmp_tow_ms</i>	<ul style="list-style-type: none"> Time stamp in GPS time of week(in milliseconds) Bit to check in ParamPresenceMask - 16
<i>TimeStmp_gps-week</i>	<ul style="list-style-type: none"> GPS week number Bit to check in ParamPresenceMask - 16
<i>Time_uncert_ms</i>	<ul style="list-style-type: none"> Time uncertainty (in milliseconds). The uncertainty is provided at 99 percent confidence. Bit to check in ParamPresenceMask - 16
<i>Iono_valid</i>	<ul style="list-style-type: none"> Iono validity. Values: <ul style="list-style-type: none"> 0 - Invalid 1 - Valid Bit to check in ParamPresenceMask - 16
<i>gps_ephemeris-sv_msk</i>	<ul style="list-style-type: none"> GPS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. Bit to check in ParamPresenceMask - 16
<i>gps_almanac-sv_msk</i>	<ul style="list-style-type: none"> GPS SV mask for almanac; if the bit is set, almanac for that SV is available. Bit to check in ParamPresenceMask - 16

<i>gps_health_sv_msk</i>	<ul style="list-style-type: none"> GPS SV mask for health; if the bit is set, health for that SV is available. Bit to check in ParamPresenceMask - 16
<i>gps_visible_sv_msk</i>	<ul style="list-style-type: none"> GPS SV mask for visible Svs; if the bit is set, the SV is available. Bit to check in ParamPresenceMask - 16
<i>glo_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> GLONASS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. Bit to check in ParamPresenceMask - 16
<i>glo_almanac_sv_msk</i>	<ul style="list-style-type: none"> GLONASS SV mask for almanac; if the bit is set, almanac for that SV is available. Bit to check in ParamPresenceMask - 16
<i>glo_health_sv_msk</i>	<ul style="list-style-type: none"> GLONASS SV mask for health; if the bit is set, health for that SV is available. Bit to check in ParamPresenceMask - 16
<i>glo_visible_sv_msk</i>	<ul style="list-style-type: none"> GLONASS SV mask for visible SVs; if the bit is set, the SV is available. Bit to check in ParamPresenceMask - 16
<i>sbas_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> SBAS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available. Bit to check in ParamPresenceMask - 16
<i>sbas_almanac_sv_msk</i>	<ul style="list-style-type: none"> SBAS SV mask for almanac; if the bit is set, almanac for that SV is available. Bit to check in ParamPresenceMask - 16
<i>sbas_health_sv_msk</i>	<ul style="list-style-type: none"> SBAS SV mask for health; if the bit is set, health for that SV is available. Bit to check in ParamPresenceMask - 16
<i>sbas_visible_sv_msk</i>	<ul style="list-style-type: none"> SBAS SV mask for visible SVs; if the bit is set, the SV is available. Bit to check in ParamPresenceMask - 16
<i>xtra_start_gps_week</i>	<ul style="list-style-type: none"> Current XTRA information is valid starting from this GPS week number Bit to check in ParamPresenceMask - 16
<i>xtra_start_gps_minutes</i>	<ul style="list-style-type: none"> Current XTRA information is valid starting from the GPS minutes with the GPS week Bit to check in ParamPresenceMask - 16
<i>xtra_valid_duration_hours</i>	<ul style="list-style-type: none"> XTRA information is valid for this many hours starting from the specified GPS week/minutes Bit to check in ParamPresenceMask - 16

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.853.2 Field Documentation

- 8.853.2.1 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::Altitude
- 8.853.2.2 uint8_t unpack_pds_SLQSGetGPSSStateInfo_t::EngineState
- 8.853.2.3 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::glo_almanac_sv_msk
- 8.853.2.4 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::glo_ephemeris_sv_msk
- 8.853.2.5 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::glo_health_sv_msk
- 8.853.2.6 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::glo_visible_sv_msk
- 8.853.2.7 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::gps_almanac_sv_msk
- 8.853.2.8 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::gps_ephemeris_sv_msk
- 8.853.2.9 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::gps_health_sv_msk
- 8.853.2.10 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::gps_visible_sv_msk
- 8.853.2.11 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::HorizontalUncertainty
- 8.853.2.12 uint8_t unpack_pds_SLQSGetGPSSStateInfo_t::lono_valid
- 8.853.2.13 uint64_t unpack_pds_SLQSGetGPSSStateInfo_t::Latitude
- 8.853.2.14 uint64_t unpack_pds_SLQSGetGPSSStateInfo_t::Longitude
- 8.853.2.15 swi_uint256_t unpack_pds_SLQSGetGPSSStateInfo_t::ParamPresenceMask
- 8.853.2.16 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::sbas_almanac_sv_msk
- 8.853.2.17 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::sbas_ephemeris_sv_msk
- 8.853.2.18 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::sbas_health_sv_msk
- 8.853.2.19 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::sbas_visible_sv_msk
- 8.853.2.20 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::Time_uncert_ms
- 8.853.2.21 uint16_t unpack_pds_SLQSGetGPSSStateInfo_t::TimeStmp_gps_week
- 8.853.2.22 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::TimeStmp_tow_ms
- 8.853.2.23 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::ValidMask
- 8.853.2.24 uint32_t unpack_pds_SLQSGetGPSSStateInfo_t::VerticalUncertainty

8.853.2.25 `uint16_t unpack_pds_SLQSGetGPSStateInfo_t::extra_start_gps_minutes`

8.853.2.26 `uint16_t unpack_pds_SLQSGetGPSStateInfo_t::extra_start_gps_week`

8.853.2.27 `uint16_t unpack_pds_SLQSGetGPSStateInfo_t::extra_valid_duration_hours`

8.854 `unpack_qmi_t` Struct Reference

Data Fields

- enum [msgtype](#) `type`
- `uint16_t` [msgid](#)
- `uint16_t` [xid](#)

8.854.1 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

8.854.2 Field Documentation

8.854.2.1 `uint16_t unpack_qmi_t::msgid`

8.854.2.2 `enum msgtype unpack_qmi_t::type`

8.854.2.3 `uint16_t unpack_qmi_t::xid`

8.855 `unpack_qos_BindDataPort_t` Struct Reference

Data Fields

- [swi_uint256_t](#) `ParamPresenceMask`

8.855.1 Detailed Description

This structure contains unpack Qos BindDataPort.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.855.2 Field Documentation

8.855.2.1 `swi_uint256_t unpack_qos_BindDataPort_t::ParamPresenceMask`

8.856 unpack_qos_dataRate_t Struct Reference

Data Fields

- uint32_t [dataRateMax](#)
- uint32_t [guaranteedRate](#)

8.856.1 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)

8.856.2 Field Documentation

8.856.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

8.856.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

8.857 unpack_qos_IPv4Addr_t Struct Reference

Data Fields

- uint32_t [addr](#)
- uint32_t [subnetMask](#)

8.857.1 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"> • (addr and subnetMask) == (IP pkt addr & 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

8.857.2 Field Documentation

8.857.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

8.857.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

8.858 unpack_qos_IPv6Addr_t Struct Reference

Data Fields

- uint8_t [addr](#) [16]

- uint8_t [prefixLen](#)

8.858.1 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.

8.858.2 Field Documentation

8.858.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

8.858.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

8.859 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

- uint8_t [val](#)
- uint8_t [mask](#)

8.859.1 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"> • IPv6_filter_tc_val = 00101000 • 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

8.859.2 Field Documentation

8.859.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.859.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.860 unpack_qos_pktErrRate_t Struct Reference

Data Fields

- uint16_t [multiplier](#)

- uint16_t [exponent](#)

8.860.1 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)

8.860.2 Field Documentation

8.860.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.860.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.861 unpack_qos_Port_t Struct Reference

Data Fields

- uint16_t [port](#)
- uint16_t [range](#)

8.861.1 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

8.861.2 Field Documentation

8.861.2.1 uint16_t unpack_qos_Port_t::port

8.861.2.2 uint16_t unpack_qos_Port_t::range

8.862 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](#)

8.862.1 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"> • QoS flow state information, please check unpack_qos_QosFlowInfoState_t for more information
<i>is_TxQFlow-Granted_-Available</i>	<ul style="list-style-type: none"> • TRUE if optional TxQFlowGranted is available
<i>TxQFlow-Granted</i>	<ul style="list-style-type: none"> • The Tx Qos flow granted, please check unpack_qos_swiQosFlow_t for more information
<i>is_RxQFlow-Granted_-Available</i>	<ul style="list-style-type: none"> • TRUE if optional RxQFlowGranted is available
<i>RxQFlow-Granted</i>	<ul style="list-style-type: none"> • The Rx Qos flow granted, please check unpack_qos_swiQosFlow_t for more information
<i>NumTxFilters</i>	<ul style="list-style-type: none"> • Number of Tx filters available
<i>TxQFilter</i>	<ul style="list-style-type: none"> • The Tx Qos filter, please check unpack_qos_swiQosFilter_t for more information • See LITEQMI_MAX_QOS_FILTERS for more information
<i>NumRxFilters</i>	<ul style="list-style-type: none"> • Number of Tx filters available
<i>RxQFilter</i>	<ul style="list-style-type: none"> • The Rx Qos filter, please check unpack_qos_swiQosFilter_t for more information • See LITEQMI_MAX_QOS_FILTERS for more information
<i>BearerID</i>	<ul style="list-style-type: none"> • The bearer ID • Bearer ID or Radio Link Protocol (RLP) ID of the activated flow. • Valid Values - 0 to 16 • 0xFF - Invalid value.

8.862.2 Field Documentation

8.862.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`

8.862.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`

8.862.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`

8.862.2.4 uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters

8.862.2.5 uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters

8.862.2.6 unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState

8.862.2.7 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]

8.862.2.8 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted

8.862.2.9 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]

8.862.2.10 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted

8.863 unpack_qos_QosFlowInfoState_t Struct Reference

Data Fields

- uint32_t [id](#)
- uint8_t [isNewFlow](#)
- uint8_t [state](#)

8.863.1 Detailed Description

This structure contains QoS flow state

Parameters

<i>id</i>	QoS identifier
<i>isNewFlow</i>	<ul style="list-style-type: none"> • 1 - Newly added flow • 0 - Existing flow
<i>state</i>	This indicates that the flow that was added/modified/deleted: <ul style="list-style-type: none"> • 0x01 - Flow activated • 0x02 - Flow modified • 0x03 - Flow deleted • 0x04 - Flow suspended • 0x05 - Flow enabled • 0x06 - Flow disabled

8.863.2 Field Documentation

8.863.2.1 uint32_t unpack_qos_QosFlowInfoState_t::id

8.863.2.2 uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow

8.863.2.3 uint8_t unpack_qos_QosFlowInfoState_t::state

8.864 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

Data Fields

- uint8_t [NWQoSStatus](#)
- swi_uint256_t [ParamPresenceMask](#)

8.864.1 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"> • 0 - No QoS support in network • 1 - Network supports QoS • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.864.2 Field Documentation

8.864.2.1 uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus

8.864.2.2 swi_uint256_t unpack_qos_SLQSQosGetNetworkStatus_t::ParamPresenceMask

8.865 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](#)

8.865.1 Detailed Description

Structure that contains extra APN parameters

Parameters

<i>apnId</i>	<ul style="list-style-type: none"> • APN id • ID identifying the APN that the client would like to query the AMBR params • Bit to check in ParamPresenceMask - 3
--------------	--

<i>ambr_ul</i>	<ul style="list-style-type: none"> • APN AMBR uplink • APN AMBR uplink values from 1 kbps to 8640 kbps • Bit to check in ParamPresenceMask - 3
<i>ambr_dl</i>	<ul style="list-style-type: none"> • APN AMBR downlink • APN AMBR downlink values from 1 kbps to 8640 kbps • Bit to check in ParamPresenceMask - 3
<i>ambr_ul_ext</i>	<ul style="list-style-type: none"> • Extended APN AMBR uplink • APN AMBR uplink values from 8700 kbps to 256 Mbps • Bit to check in ParamPresenceMask - 3
<i>ambr_dl_ext</i>	<ul style="list-style-type: none"> • Extended APN AMBR downlink • APN AMBR downlink values from 8700 kbps to 256 Mbps • Bit to check in ParamPresenceMask - 3
<i>ambr_ul_ext2</i>	<ul style="list-style-type: none"> • Second extended APN AMBR uplink • APN AMBR uplink values from 256 Mbps to 65280 Mbps • Bit to check in ParamPresenceMask - 3
<i>ambr_dl_ext2</i>	<ul style="list-style-type: none"> • Second extended APN AMBR downlink • APN AMBR downlink values from 256 Mbps to 65280 Mbps • Bit to check in ParamPresenceMask - 3
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.865.2 Field Documentation

8.865.2.1 uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl

8.865.2.2 uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl_ext

8.865.2.3 uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl_ext2

8.865.2.4 uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul

8.865.2.5 uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul_ext

8.865.2.6 uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul_ext2

8.865.2.7 uint32_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::apnId

8.865.2.8 swi_uint256_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ParamPresenceMask

8.866 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](#)

8.866.1 Detailed Description

Structure that contains APN data statistics

Parameters

<i>apnId</i>	<ul style="list-style-type: none"> • APN id • ID identifying the connected APN that the client would like to query the data statistic for • Bit to check in ParamPresenceMask - 3
<i>total_tx_pkt</i>	<ul style="list-style-type: none"> • sum of all packets sent • Bit to check in ParamPresenceMask - 3
<i>total_tx_pkt_drp</i>	<ul style="list-style-type: none"> • sum of all(TX) packets dropped • Bit to check in ParamPresenceMask - 3
<i>total_rx_pkt</i>	<ul style="list-style-type: none"> • sum of all packets received • Bit to check in ParamPresenceMask - 3
<i>total_tx_bytes</i>	<ul style="list-style-type: none"> • sum of all bytes sent • Bit to check in ParamPresenceMask - 3
<i>total_tx_bytes_drp</i>	<ul style="list-style-type: none"> • sum of all(TX) bytes dropped • Bit to check in ParamPresenceMask - 3
<i>total_rx_bytes</i>	<ul style="list-style-type: none"> • number of received bytes for the QoS flow ID • Bit to check in ParamPresenceMask - 3
<i>numQosFlow</i>	<ul style="list-style-type: none"> • pointer to number of QoS flow Stat • Bit to check in ParamPresenceMask - 4

<i>qosFlow[LITEQMI_MAX_QOS_FLOW_PER_APN_STATS]</i>	<ul style="list-style-type: none"> • Data statistic per QoS flow • See unpack_QosFlowStat_t for more information • See LITEQMI_MAX_QOS_FLOW_PER_APN_STATS for more information • Bit to check in ParamPresenceMask - 4
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.866.2 Field Documentation

8.866.2.1 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId`

8.866.2.2 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow`

8.866.2.3 `swi_uint256_t unpack_qos_SLQSQosSwiReadDataStats_t::ParamPresenceMask`

8.866.2.4 `unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]`

8.866.2.5 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes`

8.866.2.6 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt`

8.866.2.7 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes`

8.866.2.8 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp`

8.866.2.9 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt`

8.866.2.10 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp`

8.867 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

Data Fields

- `uint8_t NumFlows`
- `unpack_qos_QosFlowInfo_t QosFlowInfo` [8]
- `swi_uint256_t ParamPresenceMask`

8.867.1 Detailed Description

Structure with QoS event details

Parameters

<i>NumFlows</i>	<ul style="list-style-type: none"> • Number of QoS flows available • Bit to check in ParamPresenceMask - 16
-----------------	---

<i>QosFlowInfo</i>	<ul style="list-style-type: none"> • The Qos flow details, please check unpack_qos_QosFlowInfo_t for more information • See LITEQMI_MAX_QOS_FLOWS for more information • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.867.2 Field Documentation

8.867.2.1 `uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows`

8.867.2.2 `swi_uint256_t unpack_qos_SLQSSetQosEventCallback_ind_t::ParamPresenceMask`

8.867.2.3 `unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]`

8.868 unpack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.868.1 Detailed Description

This structure contains unpack QOS SLQSSetQosEventCallback. param ParamPresenceMask

- bitmask representation to indicate valid parameters.

8.868.2 Field Documentation

8.868.2.1 `swi_uint256_t unpack_qos_SLQSSetQosEventCallback_t::ParamPresenceMask`

8.869 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

Data Fields

- `uint8_t status`
- [swi_uint256_t ParamPresenceMask](#)

8.869.1 Detailed Description

Structure with network's QoS status

Parameters

<i>status</i>	<p>Network QoS support status</p> <ul style="list-style-type: none"> • 0x00 - Current network does not support QoS • 0x01 - Current network supports QoS • Bit to check in ParamPresenceMask - 1
---------------	--

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

Note

- Technology Supported: CDMA

8.869.2 Field Documentation

8.869.2.1 `swi_uint256_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::ParamPresenceMask`

8.869.2.2 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

8.870 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference**Data Fields**

- `uint16_t event`
- `swi_uint256_t ParamPresenceMask`

8.870.1 Detailed Description

Structure with QoS primary flow events

Parameters

<i>event</i>	Event which causes this indication: <ul style="list-style-type: none"> 0x0001 - Primary flow QoS modify operation success 0x0002 - Primary flow QoS modify operation failure Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.870.2 Field Documentation

8.870.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

8.870.2.2 `swi_uint256_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::ParamPresenceMask`

8.871 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`

8.871.1 Detailed Description

Structure with QoS status indication details

Parameters

<i>id</i>	<ul style="list-style-type: none"> • Index identifying the QoS flow whose status is being reported • Bit to check in ParamPresenceMask - 1
<i>status</i>	Current QoS flow status: <ul style="list-style-type: none"> • 0x01 - QMI_QOS_STATUS_ACTIVATED • 0x02 - QMI_QOS_STATUS_SUSPENDED • 0x03 - QMI_QOS_STATUS_GONE • Bit to check in ParamPresenceMask - 1
<i>event</i>	<ul style="list-style-type: none"> • 0x01 - QMI_QOS_ACTIVATED_EV • 0x02 - QMI_QOS_SUSPENDED_EV • 0x03 - QMI_QOS_GONE_EV • 0x04 - QMI_QOS_MODIFY_ACCEPTED_EV • 0x05 - QMI_QOS_MODIFY_REJECTED_EV • 0x06 - QMI_QOS_INFO_CODE_UPDATED_EV • Bit to check in ParamPresenceMask - 1
<i>reason</i>	<ul style="list-style-type: none"> • 0x01 - QMI_QOS_INVALID_PARAMS • 0x02 - QMI_QOS_INTERNAL_CALL_ENDED • 0x03 - QMI_QOS_INTERNAL_ERROR • 0x04 - QMI_QOS_INSUFFICIENT_LOCAL_Resources • 0x05 - QMI_QOS_TIMED_OUT_OPERATION • 0x06 - QMI_QOS_INTERNAL_UNKNOWN_CAUSE_CODE • 0x07 - QMI_QOS_INTERNAL_MODIFY_IN_PROGRESS • 0x08 - QMI_QOS_NOT_SUPPORTED • 0x09 - QMI_QOS_NOT_AVAILABLE • 0x0A - QMI_QOS_NOT_GUARANTEED • 0x0B - QMI_QOS_INSUFFICIENT_NETWORK_RESOURCES • 0x0C - QMI_QOS_AWARE_SYSTEM • 0x0D - QMI_QOS_UNAWARE_SYSTEM • 0x0E - QOS_REJECTED_OPERATION • 0x0F - QMI_QOS_WILL_GRANT_WHEN_QOS_RESUMED • 0x10 - QMI_QOS_NETWORK_CALL_ENDED • 0x11 - QMI_QOS_NETWORK_SERVICE_NOT_AVAILABLE • 0x12 - QMI_QOS_NETWORK_L2_LINK_RELEASED • 0x13 - QMI_QOS_NETWORK_L2_LINK_REESTAB_REJ • 0x14 - QMI_QOS_NETWORK_L2_LINK_REESTAB_IND • 0x15 - QMI_QOS_NETWORK_UNKNOWN_CAUSE_CODE • 0x16 - QMI_NETWORK_BUSY • Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.871.2 Field Documentation

8.871.2.1 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event`

8.871.2.2 `uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id`

8.871.2.3 `swi_uint256_t unpack_qos_SLQSSetQosStatusCallback_ind_t::ParamPresenceMask`

8.871.2.4 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason`

8.871.2.5 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status`

8.872 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`

8.872.1 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"> • 0x04 - IPv4 • 0x06 - Ipv6
<i>IPv4SrcAddr</i>	IPv4 filter soruce address See unpack_qos_IPv4Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>IPv4DstAddr</i>	IPv4 filter destination address See unpack_qos_IPv4Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<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"> • 0x01 = ICMP • 0x06 = TCP • 0x11 = UDP • 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP & UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified.
<i>IPv4Tos</i>	IPv4 filter type of service See unpack_qos_Tos_t for more information
<i>IPv6SrcAddr</i>	IPv6 filter soruce address See unpack_qos_IPv6Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>IPv6DstAddr</i>	IPv6 filter destination address See unpack_qos_IPv6Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>IPv6TrafCls</i>	IPv6 filter traffic class See unpack_qos_IPv6TrafCls_t 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"> • Implemented only for unsolicited indication
<i>TCPSrcPort</i>	TCP filter source port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>TCPDstPort</i>	TCP filter destination port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>UDPSrcPort</i>	UDP filter source port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>UDPDstPort</i>	UDP filter destination port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication

<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"> Implemented only for unsolicited indication
<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 unpack_qos_Port_t for more information <ul style="list-style-type: none"> Implemented only for unsolicited indication
<i>UDPDstPort</i>	Transport protocol filter destination port See unpack_qos_Port_t for more information <ul style="list-style-type: none"> Implemented only for unsolicited indication

8.872.2 Field Documentation

8.872.2.1 `uint32_t unpack_qos_swiQosFilter_t::EspSpi`

8.872.2.2 `uint16_t unpack_qos_swiQosFilter_t::Id`

8.872.2.3 `uint8_t unpack_qos_swiQosFilter_t::index`

8.872.2.4 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr`

8.872.2.5 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr`

8.872.2.6 `unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos`

8.872.2.7 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr`

8.872.2.8 `uint32_t unpack_qos_swiQosFilter_t::IPv6Label`

8.872.2.9 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr`

8.872.2.10 `unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls`

8.872.2.11 `uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available`

8.872.2.12 `uint8_t unpack_qos_swiQosFilter_t::is_Id_Available`

8.872.2.13 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available`

8.872.2.14 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available`

8.872.2.15 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available`

8.872.2.16 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available`

8.872.2.17 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available`

8.872.2.18 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available`

8.872.2.19 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available`

8.872.2.20 `uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available`

8.872.2.21 `uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available`

8.872.2.22 `uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available`

8.872.2.23 `uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available`

8.872.2.24 `uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available`

8.872.2.25 `uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available`

8.872.2.26 `uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available`

8.872.2.27 `uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available`

8.872.2.28 `uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto`

8.872.2.29 `uint16_t unpack_qos_swiQosFilter_t::Precedence`

8.872.2.30 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort`

8.872.2.31 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort`

8.872.2.32 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort`

8.872.2.33 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort`

8.872.2.34 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort`

8.872.2.35 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort`

8.872.2.36 `uint8_t unpack_qos_swiQosFilter_t::version`

8.873 `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](#)

8.873.1 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"> • Mandatory parameter • IP flow index • Integer that uniquely identifies each flow instance • Unique index must be assigned by the control point to every flow_spec instance
<i>ProfileId3GPP2</i>	<ul style="list-style-type: none"> • IP flow 3GPP2 profile ID • 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
<i>val_3GPP2Pri</i>	<ul style="list-style-type: none"> • IP flow 3GPP2 flow priority • Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices
<i>TrafficClass</i>	<ul style="list-style-type: none"> • IP flow traffic class • Integer that designates the requested traffic class: <ul style="list-style-type: none"> • 0 - Conversational • 1 - Streaming • 2 - Interactive • 3 - Background
<i>DataRate</i>	<ul style="list-style-type: none"> • IP flow data rate min max • See unpack_qos_dataRate_t for more information

<i>TokenBucket</i>	<ul style="list-style-type: none"> • IP flow data rate token bucket • See unpack_qos_tokenBucket_t for more information
<i>Latency</i>	<ul style="list-style-type: none"> • IP flow latency • Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link
<i>Jitter</i>	<ul style="list-style-type: none"> • IP flow jitter • Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link
<i>PktErrRate</i>	<ul style="list-style-type: none"> • IP flow packet error rate • See unpack_qos_pktErrRate_t for more information
<i>MinPolicedPktSz</i>	<ul style="list-style-type: none"> • IP flow minimum policed packet size • 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
<i>MaxAllowedPktSz</i>	<ul style="list-style-type: none"> • IP flow maximum allowed packet size • 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
<i>val_3GPPRes-ResidualBER</i>	<ul style="list-style-type: none"> • IP flow 3GPP residual bit error rate • residual_bit_error_rate • 0 = 5×10^{-2} residual BER • 1 = 1×10^{-2} residual BER • 2 = 5×10^{-3} residual BER • 3 = 4×10^{-3} residual BER • 4 = 1×10^{-3} residual BER • 5 = 1×10^{-4} residual BER • 6 = 1×10^{-5} residual BER • 7 = 1×10^{-6} residual BER • 8 = 6×10^{-8} residual BER • Integer that indicates the undetected BER for each IP flow in the delivered packets; Applies only to 3GPP networks
<i>val_3GPPTra-HdlPri</i>	<ul style="list-style-type: none"> • 3GPP traffic handling priority • 0 - Relative traffic handling priority 1 • 1 - Relative traffic handling priority 2 • 2 - Relative traffic handling priority 3 • Defines the relative priority of the flow; applies only to 3GPP networks

<i>val_3GPPImCn</i>	<ul style="list-style-type: none"> • IP flow 3GPP IM CN flag • IM CN subsystem signaling flag: • 0x00 - FALSE • 0x01 - TRUE • This parameter applies only to 3GPP networks
<i>val_3GPPSigInd</i>	<ul style="list-style-type: none"> • IP flow 3GPP signaling indication • 0x00 - FALSE • 0x01 - TRUE • This parameter applies only to 3GPP networks
<i>LteQci</i>	<ul style="list-style-type: none"> • LTE QoS Class Identifier • QoS Class Identifier(QCI) is a required parameter to request QoS in LTE • QCI values: <ul style="list-style-type: none"> – QCI value 0 requests the network to assign the appropriate QCI value – QCI values 1-4 are associated with guaranteed bitrates – QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored

8.873.2 Field Documentation

8.873.2.1 `unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate`

8.873.2.2 `uint8_t unpack_qos_swiQosFlow_t::index`

8.873.2.3 `uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.873.2.4 `uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.873.2.5 `uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available`

8.873.2.6 `uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.873.2.7 `uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.873.2.8 `uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.873.2.9 `uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.873.2.10 `uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.873.2.11 `uint8_t unpack_qos_swiQosFlow_t::is-TokenBucket_Available`

8.873.2.12 `uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.873.2.13 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.873.2.14 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

8.873.2.15 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available`

8.873.2.16 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available`

8.873.2.17 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available`

8.873.2.18 `uint32_t unpack_qos_swiQosFlow_t::Jitter`

8.873.2.19 `uint32_t unpack_qos_swiQosFlow_t::Latency`

8.873.2.20 `uint8_t unpack_qos_swiQosFlow_t::LteQci`

8.873.2.21 `uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz`

8.873.2.22 `uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz`

8.873.2.23 `unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate`

8.873.2.24 `uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2`

8.873.2.25 `unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket`

8.873.2.26 `uint8_t unpack_qos_swiQosFlow_t::TrafficClass`

8.873.2.27 `uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri`

8.873.2.28 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPIImCn`

8.873.2.29 `uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER`

8.873.2.30 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd`

8.873.2.31 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri`

8.874 `unpack_qos_tokenBucket_t` Struct Reference

Data Fields

- `uint32_t` [peakRate](#)
- `uint32_t` [tokenRate](#)
- `uint32_t` [bucketSz](#)

8.874.1 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

8.874.2 Field Documentation

8.874.2.1 uint32_t unpack_qos_tokenBucket_t::bucketSz

8.874.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

8.874.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

8.875 unpack_qos_Tos_t Struct Reference

Data Fields

- uint8_t [val](#)
- uint8_t [mask](#)

8.875.1 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"> • IPv4_filter_tos_val = 00101000 • 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.

8.875.2 Field Documentation

8.875.2.1 uint8_t unpack_qos_Tos_t::mask

8.875.2.2 uint8_t unpack_qos_Tos_t::val

8.876 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](#)

8.876.1 Detailed Description

This structure contains the Data statistic per QoS flow

Parameters

<i>bearerId</i>	<ul style="list-style-type: none"> • Bearer ID
<i>tx_pkt</i>	<ul style="list-style-type: none"> • number of sent packets for the QoS flow ID
<i>tx_pkt_drp</i>	<ul style="list-style-type: none"> • number of dropped(TX) packets for the QoS flow ID
<i>tx_bytes</i>	<ul style="list-style-type: none"> • number of sent bytes for the QoS flow ID
<i>tx_bytes_drp</i>	<ul style="list-style-type: none"> • number of dropped(TX) bytes for the QoS flow ID

8.876.2 Field Documentation

8.876.2.1 `uint32_t unpack_QosFlowStat_t::bearerId`8.876.2.2 `uint64_t unpack_QosFlowStat_t::tx_bytes`8.876.2.3 `uint64_t unpack_QosFlowStat_t::tx_bytes_drp`8.876.2.4 `uint32_t unpack_QosFlowStat_t::tx_pkt`8.876.2.5 `uint32_t unpack_QosFlowStat_t::tx_pkt_drp`8.877 `unpack_result_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

8.877.1 Detailed Description

unpack result struct

Parameters

<code>out</code>	<i>Tlvresult</i>	
<code>out</code>	<i>ParamPresenceMask</i>	Parameter presence bit mask

8.877.2 Field Documentation

8.877.2.1 `swi_uint256_t unpack_result_t::ParamPresenceMask`8.877.2.2 `uint16_t unpack_result_t::Tlvresult`

8.878 unpack_rms_GetSMSWake_t Struct Reference

Data Fields

- uint32_t [enabled](#)
- uint32_t [wake_mask](#)
- [swi_uint256_t ParamPresenceMask](#)

8.878.1 Detailed Description

Unpack get SMS wake parameters

Parameters

<i>enabled</i>	<ul style="list-style-type: none"> • SMS wake functionality enabled <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 16
<i>wake_mask</i>	<ul style="list-style-type: none"> • SMS wake mask to search for incoming messages (only relevant when enabled) • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.878.2 Field Documentation

8.878.2.1 uint32_t unpack_rms_GetSMSWake_t::enabled

8.878.2.2 swi_uint256_t unpack_rms_GetSMSWake_t::ParamPresenceMask

8.878.2.3 uint32_t unpack_rms_GetSMSWake_t::wake_mask

8.879 unpack_rms_SetSMSWake_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.879.1 Detailed Description

This structure contains unpack Set SMS Wake.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
---------------------------	--

8.879.2 Field Documentation

8.879.2.1 `swi_uint256_t unpack_rms_SetSMSWake_t::ParamPresenceMask`

8.880 `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](#)

8.880.1 Detailed Description

WDS packet RM Transfer Statistics data structure for individual session

Parameters

<i>TxOkConutTlv</i>	<ul style="list-style-type: none"> • Tx Ok Packet Tlv Value. • Bit to check in ParamPresenceMask - 16
<i>RxOkConutTlv</i>	<ul style="list-style-type: none"> • Rx Ok Packet Tlv Value. • Bit to check in ParamPresenceMask - 17
<i>TxOkByteCount-Tlv</i>	<ul style="list-style-type: none"> • Tx Ok Byte Count Packet Tlv Value. • Bit to check in ParamPresenceMask - 25
<i>RxOkByteCount-Tlv</i>	<ul style="list-style-type: none"> • Rx Ok Byte Count Packet Tlv Value. • Bit to check in ParamPresenceMask - 26
<i>TxDropConutTlv</i>	<ul style="list-style-type: none"> • Tx Drop Count Packet Tlv Value. • Bit to check in ParamPresenceMask - 37
<i>RxDropConutTlv</i>	<ul style="list-style-type: none"> • Rx Drop Count Packet Tlv Value. • Bit to check in ParamPresenceMask - 38

8.880.2 Field Documentation

8.880.2.1 `swi_uint256_t unpack_RMTransferStatistics_ind_t::ParamPresenceMask`

8.880.2.2 `wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv`

8.880.2.3 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv

8.880.2.4 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv

8.880.2.5 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv

8.880.2.6 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv

8.880.2.7 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv

8.881 unpack_sar_SLQSGetRfSarState_t Struct Reference

Data Fields

- uint32_t * [pSarRFState](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.881.1 Detailed Description

This structure contains unpack get RF SAR state parameter.

Parameters

<i>pSarRFState</i>	<ul style="list-style-type: none"> • SAR RF State <ul style="list-style-type: none"> – QMI_SAR_RF_STATE_DEFAULT – QMI_SAR_RF_STATE_1 – QMI_SAR_RF_STATE_2 – QMI_SAR_RF_STATE_3 – QMI_SAR_RF_STATE_4 – QMI_SAR_RF_STATE_5 – QMI_SAR_RF_STATE_6 – QMI_SAR_RF_STATE_7 – QMI_SAR_RF_STATE_8 • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.881.2 Field Documentation

8.881.2.1 [swi_uint256_t](#) [unpack_sar_SLQSGetRfSarState_t::ParamPresenceMask](#)

8.881.2.2 [uint32_t*](#) [unpack_sar_SLQSGetRfSarState_t::pSarRFState](#)

8.882 unpack_sms_GetSMSCAddress_t Struct Reference

Data Fields

- uint8_t [addressSize](#)
- uint8_t * [pSMSCAddress](#)

- [uint8_t typeSize](#)
- [uint8_t * pSMSCType](#)
- [swi_uint256_t ParamPresenceMask](#)

8.882.1 Detailed Description

Structure contains Gets the SMS center address parameters.

Parameters

<i>addressSize</i>	<ul style="list-style-type: none"> • The maximum number of characters (including NULL terminator) that the SMS center address array can contain. • Bit to check in ParamPresenceMask - 1
<i>pSMSCAddress</i>	<ul style="list-style-type: none"> • The SMS center address represented as a NULL terminated string. • Bit to check in ParamPresenceMask - 1
<i>typeSize</i>	<ul style="list-style-type: none"> • The maximum number of characters (including NULL terminator) that the SMS center address type array can contain. • Bit to check in ParamPresenceMask - 1
<i>pSMSCType</i>	<ul style="list-style-type: none"> • The SMS center address type represented as a NULL terminated string. • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.882.2 Field Documentation

8.882.2.1 [uint8_t unpack_sms_GetSMSCAddress_t::addressSize](#)

8.882.2.2 [swi_uint256_t unpack_sms_GetSMSCAddress_t::ParamPresenceMask](#)

8.882.2.3 [uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCAddress](#)

8.882.2.4 [uint8_t* unpack_sms_GetSMSCAddress_t::pSMSCType](#)

8.882.2.5 [uint8_t unpack_sms_GetSMSCAddress_t::typeSize](#)

8.883 unpack_sms_SaveSMS_t Struct Reference

Data Fields

- [uint32_t * pMessageIndex](#)
- [swi_uint256_t ParamPresenceMask](#)

8.883.1 Detailed Description

This structure contains unpack save SMS parameter.

Parameters

<i>pMessageIndex</i>	<ul style="list-style-type: none"> The message index assigned by the device Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.883.2 Field Documentation

8.883.2.1 `swi_uint256_t unpack_sms_SaveSMS_t::ParamPresenceMask`

8.883.2.2 `uint32_t* unpack_sms_SaveSMS_t::pMessageIndex`

8.884 unpack_sms_SendSMS_t Struct Reference

Data Fields

- `uint16_t` [messageID](#)
- `uint32_t` [messageFailureCode](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.884.1 Detailed Description

This structure contains unpack send SMS parameters.

Parameters

<i>messageID</i>	<ul style="list-style-type: none"> WMS message ID Bit to check in ParamPresenceMask - 1
<i>messageFailureCode</i>	<ul style="list-style-type: none"> pointer to message failure code. Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.884.2 Field Documentation

8.884.2.1 `uint32_t unpack_sms_SendSMS_t::messageFailureCode`

8.884.2.2 `uint16_t unpack_sms_SendSMS_t::messageID`

8.884.2.3 `swi_uint256_t unpack_sms_SendSMS_t::ParamPresenceMask`

8.885 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

8.885.1 Detailed Description

This structure contains unpack new SMS callback indication.

Parameters

<i>NewMMTlv</i>	<ul style="list-style-type: none"> • MT message • See newMTMessageTlv for more information • Bit to check in ParamPresenceMask - 16
<i>TRMessageTlv</i>	<ul style="list-style-type: none"> • Transfer Route MT Message • See transferRouteMessageTlv for more information • Bit to check in ParamPresenceMask - 17
<i>MMTlv</i>	<ul style="list-style-type: none"> • Message mode • See messageModeTlv for more information • Bit to check in ParamPresenceMask - 18
<i>ETWSTlv</i>	<ul style="list-style-type: none"> • ETWS Message • See sMSEtwsMessageTlv for more information • Bit to check in ParamPresenceMask - 19
<i>ETWSPLMNTlv</i>	<ul style="list-style-type: none"> • ETWS PLMN Information • See eTWSPLMNInfoTlv for more information • Bit to check in ParamPresenceMask - 20
<i>SMSCTlv</i>	<ul style="list-style-type: none"> • SMSC Address • See sMSCAddressTlv for more information • Bit to check in ParamPresenceMask - 21
<i>IMSTlv</i>	<ul style="list-style-type: none"> • SMS on IMS • See sMSOnIMSTlv for more information • Bit to check in ParamPresenceMask - 22

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">bitmask representation to indicate valid parameters.
---------------------------	--

8.885.2 Field Documentation

8.885.2.1 `eTWSPLMNInfoTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv`

8.885.2.2 `sMSEtwsMessageTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv`

8.885.2.3 `sMSONIMSTlv unpack_sms_SetNewSMSCallback_ind_t::IMSTlv`

8.885.2.4 `messageModeTlv unpack_sms_SetNewSMSCallback_ind_t::MMTlv`

8.885.2.5 `newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv`

8.885.2.6 `swi_uint256_t unpack_sms_SetNewSMSCallback_ind_t::ParamPresenceMask`

8.885.2.7 `sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSCtlv`

8.885.2.8 `transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv`

8.886 unpack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.886.1 Detailed Description

This structure contains unpack set new SMS callback.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">bitmask representation to indicate valid parameters.
---------------------------	--

8.886.2 Field Documentation

8.886.2.1 `swi_uint256_t unpack_sms_SetNewSMSCallback_t::ParamPresenceMask`

8.887 unpack_sms_SetSMSCAddress_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.887.1 Detailed Description

This structure contains unpack Set SMS Address parameter.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.887.2 Field Documentation

8.887.2.1 `swi_uint256_t` `unpack_sms_SetSMSCAddress_t::ParamPresenceMask`

8.888 `unpack_sms_SLQSDeleteSMS_t` Struct Reference

Data Fields

- [swi_uint256_t](#) `ParamPresenceMask`

8.888.1 Detailed Description

This structure contains unpack delete SMS parameters.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.888.2 Field Documentation

8.888.2.1 `swi_uint256_t` `unpack_sms_SLQSDeleteSMS_t::ParamPresenceMask`

8.889 `unpack_sms_SLQSGetIndicationRegister_t` Struct Reference

Data Fields

- [sms_getIndicationReg](#) * `pGetIndicationRegInfo`
- [swi_uint256_t](#) `ParamPresenceMask`

8.889.1 Detailed Description

Structure containing Parameters that provides registration state of different WMS indications.

Parameters

<i>pGetIndicationRegInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of <code>getIndicationReg</code> <ul style="list-style-type: none"> – See sms_getIndicationReg for more information – <code>pRegTransLayerInfoEvt</code> <ul style="list-style-type: none"> * Bit to check in <code>ParamPresenceMask</code> - 16 – <code>pRegTransNWRegInfoEvt</code> <ul style="list-style-type: none"> * Bit to check in <code>ParamPresenceMask</code> - 17 – <code>pRegCallStatInfoEvt</code> <ul style="list-style-type: none"> * Bit to check in <code>ParamPresenceMask</code> - 18
------------------------------	--

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.889.2 Field Documentation

8.889.2.1 `swi_uint256_t unpack_sms_SLQSGetIndicationRegister_t::ParamPresenceMask`

8.889.2.2 `sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo`

8.890 unpack_sms_SLQSGetMessageWaiting_t Struct Reference

Data Fields

- [sms_getMsgWaitingInfo](#) * [pGetMsgWaitingInfoResp](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.890.1 Detailed Description

Structure contain Parameter that provide information about the message waiting information.

Parameters

<i>pGetMsgWaitingInfoResp</i>	<ul style="list-style-type: none"> Pointer to structure of <code>getMsgWaitingInfoResp</code> <ul style="list-style-type: none"> See sms_getMsgWaitingInfo for more information Bit to check in <code>ParamPresenceMask</code> - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.890.2 Field Documentation

8.890.2.1 `swi_uint256_t unpack_sms_SLQSGetMessageWaiting_t::ParamPresenceMask`

8.890.2.2 `sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp`

8.891 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](#)

8.891.1 Detailed Description

This sturcture contains unpack get SMS parameters.

Parameters

<i>messageTag</i>	<ul style="list-style-type: none"> • Message tag <ul style="list-style-type: none"> – 0 - Read – 1 - Not read – 2 - Mobile originated and sent – 3 - Mobile originated but not yet sent
-------------------	---

- Bit to check in ParamPresenceMask - 1

Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> • Message format <ul style="list-style-type: none"> – 0 - CDMA (IS-637B) – 1 - 5 (Reserved) – 6 - GSM/WCDMA PP
----------------------	--

- Bit to check in ParamPresenceMask - 1

Parameters

<i>messageSize</i>	<ul style="list-style-type: none"> • Upon input the maximum number of bytes that can be written to the message array. • Upon successful output the actual number of bytes written to the message array. • Bit to check in ParamPresenceMask - 1
<i>message</i>	<ul style="list-style-type: none"> • The message contents array
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.891.2 Field Documentation

8.891.2.1 `uint8_t unpack_sms_SLQSGetSMS_t::message[2048]`

8.891.2.2 `uint32_t unpack_sms_SLQSGetSMS_t::messageFormat`

8.891.2.3 `uint32_t unpack_sms_SLQSGetSMS_t::messageSize`

8.891.2.4 `uint32_t unpack_sms_SLQSGetSMS_t::messageTag`

8.891.2.5 `swi_uint256_t unpack_sms_SLQSGetSMS_t::ParamPresenceMask`

8.892 `unpack_sms_SLQSGetSmsBroadcastConfig_t` Struct Reference

Data Fields

- [sms_qaQmi3GPPBroadcastCfgInfo](#) * [pBroadcastConfig](#)
- [sms_qaQmi3GPP2BroadcastCfgInfo](#) * [pCDMABroadcastConfig](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.892.1 Detailed Description

Structure contain parameters that Provides Information about the SMS BroadcastConfiguration

Parameters

<i>pBroadcast-Config</i>	<ul style="list-style-type: none"> • The data for 3GPP Broadcast Information(Optional). • Bit to check in ParamPresenceMask - 16
<i>pCDMA-BroadcastConfig</i>	<ul style="list-style-type: none"> • The data for 3GPP2 Broadcast Information(Optional). • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.892.2 Field Documentation

8.892.2.1 [swi_uint256_t](#) [unpack_sms_SLQSGetSmsBroadcastConfig_t::ParamPresenceMask](#)

8.892.2.2 [sms_qaQmi3GPPBroadcastCfgInfo](#)* [unpack_sms_SLQSGetSmsBroadcastConfig_t::pBroadcastConfig](#)

8.892.2.3 [sms_qaQmi3GPP2BroadcastCfgInfo](#)* [unpack_sms_SLQSGetSmsBroadcastConfig_t::pCDMABroadcastConfig](#)

8.893 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- [uint32_t](#) [messageListSize](#)
- [qmiSmsMessageList](#) [messageList](#) [255]
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.893.1 Detailed Description

This structure contains unpack get SMS list parameters.

Parameters

<i>messageListSize</i>	<ul style="list-style-type: none"> • Upon input the maximum number of elements that the message list array can contain. • Upon successful output the actual number of elements in the message list array. • Bit to check in ParamPresenceMask - 1
------------------------	---

<i>messageList</i>	<ul style="list-style-type: none"> • Message List • See qmiSmsMessageList for more information • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.893.2 Field Documentation

8.893.2.1 `qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]`

8.893.2.2 `uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize`

8.893.2.3 `swi_uint256_t unpack_sms_SLQSGetSMSList_t::ParamPresenceMask`

8.894 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference

Data Fields

- [sms_getTransLayerInfo](#) * [pGetTransLayerInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.894.1 Detailed Description

Structure contain parameters that gives information about the transport layer.

Parameters

<i>pGetTransLayerInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of getTransLayerInfo <ul style="list-style-type: none"> – See sms_getTransLayerInfo for more information – <code>pGetTransLayerInfo->pRegInd</code> <ul style="list-style-type: none"> * Bit to check in ParamPresenceMask - 16 – <code>pGetTransLayerInfo->pTransLayerInfo</code> <ul style="list-style-type: none"> * Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.894.2 Field Documentation

8.894.2.1 `swi_uint256_t unpack_sms_SLQSGetTransLayerInfo_t::ParamPresenceMask`

8.894.2.2 `sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo`

8.895 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference

Data Fields

- [sms_getTransNWRegInfo](#) * [pGetTransNWRegInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.895.1 Detailed Description

Structure containing parameters that provides transport layer network registration info.

Parameters

<i>pGetTransNW-RegInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of <code>getTransNWRegInfo</code> <ul style="list-style-type: none"> – See sms_getTransNWRegInfo for more information – Bit to check in <code>ParamPresenceMask</code> - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.895.2 Field Documentation

8.895.2.1 `swi_uint256_t` `unpack_sms_SLQSGetTransNWRegInfo_t::ParamPresenceMask`

8.895.2.2 `sms_getTransNWRegInfo*` `unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo`

8.896 unpack_sms_SLQSModifySMSStatus_t Struct Reference

Data Fields

- [swi_uint256_t](#) [ParamPresenceMask](#)

8.896.1 Detailed Description

This structure contains unpack modify SMS status parameters.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
---------------------------	--

8.896.2 Field Documentation

8.896.2.1 `swi_uint256_t` `unpack_sms_SLQSModifySMSStatus_t::ParamPresenceMask`

8.897 unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference

Data Fields

- `uint8_t` [NWRegStat](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.897.1 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"> • provides the transport network registration information • Values: <ul style="list-style-type: none"> – 0x00 - No Service – 0x01 - In Progress – 0x02 - Failed – 0x03 - Limited Service – 0x04 - Full Service • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

None

8.897.2 Field Documentation

8.897.2.1 `uint8_t unpack_sms_SLQSNWRegInfoCallback_ind_t::NWRegStat`

8.897.2.2 `swi_uint256_t unpack_sms_SLQSNWRegInfoCallback_ind_t::ParamPresenceMask`

8.898 `unpack_sms_SLQSSendAsyncSMS_t` Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.898.1 Detailed Description

This structure contains unpack SLQSSendAsyncSMS parameter.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.898.2 Field Documentation

8.898.2.1 `swi_uint256_t unpack_sms_SLQSSendAsyncSMS_t::ParamPresenceMask`

8.899 `unpack_sms_SLQSSetIndicationRegister_t` Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.899.1 Detailed Description

This structure contains unpack SLQSSetIndicationRegister parameter.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.
---------------------------	--

8.899.2 Field Documentation

8.899.2.1 [swi_uint256_t unpack_sms_SLQSSetIndicationRegister_t::ParamPresenceMask](#)

8.900 unpack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.900.1 Detailed Description

This structure contains unpack SLQSSetSmsBroadcastActivation parameter.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.
---------------------------	--

8.900.2 Field Documentation

8.900.2.1 [swi_uint256_t unpack_sms_SLQSSetSmsBroadcastActivation_t::ParamPresenceMask](#)

8.901 unpack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.901.1 Detailed Description

This structure contains unpack SLQSSetSmsBroadcastConfig parameters.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.
---------------------------	--

8.901.2 Field Documentation

8.901.2.1 [swi_uint256_t unpack_sms_SLQSSetSmsBroadcastConfig_t::ParamPresenceMask](#)

8.902 unpack_sms_SLQSSetSmsStorage_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.902.1 Detailed Description

This structure contains unpack SLQSSetSmsStorage parameter.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.902.2 Field Documentation

8.902.2.1 [swi_uint256_t unpack_sms_SLQSSetSmsStorage_t::ParamPresenceMask](#)

8.903 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

Data Fields

- [sms_maxStorageSizeResp](#) * [pMaxStorageSizeResp](#)
- [swi_uint256_t ParamPresenceMask](#)

8.903.1 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"> • Response parameters for SmsSLQSGetMaxStorageSize <ul style="list-style-type: none"> – See sms_maxStorageSizeResp for more information • pMaxStorageSizeResp->maxStorageSize <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 1 • pMaxStorageSizeResp->freeSlots <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.903.2 Field Documentation

8.903.2.1 `swi_uint256_t` unpack_sms_SLQSSmsGetMaxStorageSize_t::ParamPresenceMask

8.903.2.2 `sms_maxStorageSizeResp*` unpack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeResp

8.904 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference

Data Fields

- [sms_msgProtocolResp](#) * [pMessageProtocol](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.904.1 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">• Pointer to <code>smsMsgprotocolResp</code><ul style="list-style-type: none">– See sms_msgProtocolResp for more information• Bit to check in <code>ParamPresenceMask</code> - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.904.2 Field Documentation

8.904.2.1 `swi_uint256_t` unpack_sms_SLQSSmsGetMessageProtocol_t::ParamPresenceMask

8.904.2.2 `sms_msgProtocolResp*` unpack_sms_SLQSSmsGetMessageProtocol_t::pMessageProtocol

8.905 unpack_sms_SLQSSmsSetRoutes_t Struct Reference

Data Fields

- [swi_uint256_t](#) [ParamPresenceMask](#)

8.905.1 Detailed Description

This structure contains unpack SLQSSmsSetRoutes parameter.

Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.
---------------------------	--

8.905.2 Field Documentation

8.905.2.1 `swi_uint256_t` unpack_sms_SLQSSmsSetRoutes_t::ParamPresenceMask

8.906 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference

Data Fields

- uint32_t * [pSmsStorage](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.906.1 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"> • Values: <ul style="list-style-type: none"> – 0x01 - device's permanent memory – 0x02 - UICC • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.906.2 Field Documentation

8.906.2.1 [swi_uint256_t](#) [unpack_sms_SLQSSwiGetSMSStorage_t::ParamPresenceMask](#)

8.906.2.2 [uint32_t*](#) [unpack_sms_SLQSSwiGetSMSStorage_t::pSmsStorage](#)

8.907 unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference

Data Fields

- uint8_t [regInd](#)
- [sms_transLayerInfo](#) * [pTransLayerInfo](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.907.1 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"> • Indicates whether the transport layer is registered or not • Values: <ul style="list-style-type: none"> – 0x00 - Transport layer is not registered – 0x01 - Transport layer is registered • Bit to check in ParamPresenceMask - 1
---------------	---

<i>pTransLayerInfo</i>	<ul style="list-style-type: none"> Optional parameter See sms_transLayerInfo for more information Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

Note

None

8.907.2 Field Documentation

8.907.2.1 swi_uint256_t unpack_sms_SLQSTransLayerInfoCallback_ind_t::ParamPresenceMask

8.907.2.2 sms_transLayerInfo* unpack_sms_SLQSTransLayerInfoCallback_ind_t::pTransLayerInfo

8.907.2.3 uint8_t unpack_sms_SLQSTransLayerInfoCallback_ind_t::regInd

8.908 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](#)

8.908.1 Detailed Description

This structure contains SMS parameters for indication of RAW ASYNC SEND

Parameters

<i>sendStatus</i>	<ul style="list-style-type: none"> Send Status Values: <ul style="list-style-type: none"> QMI_ERR_NONE - No error in the request QMI_ERR_CAUSE_CODE - SMS cause code QMI_ERR_MESSAGE_DELIVERY_FAILURE - Message could not be delivered QMI_ERR_NO_MEMORY - Device could not allocate memory to formulate a response Bit to check in ParamPresenceMask - 1
-------------------	--

<i>messageID</i>	<ul style="list-style-type: none"> • Unique ID assigned by WMS for non-retry messages. • Bit to check in ParamPresenceMask - 16
<i>causeCode</i>	<ul style="list-style-type: none"> • WMS cause code • Bit to check in ParamPresenceMask - 17
<i>errorClass</i>	<ul style="list-style-type: none"> • Error Class • Values: <ul style="list-style-type: none"> – 0x00 - ERROR_CLASS_TEMPORARY – 0x01 - ERROR_CLASS_PERMANENT • Bit to check in ParamPresenceMask - 18
<i>RPCause</i>	<ul style="list-style-type: none"> • GW RP cause • Bit to check in ParamPresenceMask - 19
<i>TPCause</i>	<ul style="list-style-type: none"> • GW TP Cause • Bit to check in ParamPresenceMask - 19
<i>msgDelFailure-Type</i>	<ul style="list-style-type: none"> • Message delivery failure type • Values: <ul style="list-style-type: none"> – 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY – 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT • Bit to check in ParamPresenceMask - 20
<i>msgDelFailure-Cause</i>	<ul style="list-style-type: none"> • Message delivery failure cause • Values: <ul style="list-style-type: none"> – 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL • Bit to check in ParamPresenceMask - 21
<i>alphaIDLen</i>	<ul style="list-style-type: none"> • Number of sets of the pAlphaID • Bit to check in ParamPresenceMask - 22
<i>pAlphaID</i>	<ul style="list-style-type: none"> • Alpha ID • Bit to check in ParamPresenceMask - 22
<i>userData</i>	<ul style="list-style-type: none"> • Identifies the request associated with this indication. • Bit to check in ParamPresenceMask - 23
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.908.2 Field Documentation

- 8.908.2.1 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::alphaDLen
- 8.908.2.2 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::causeCode
- 8.908.2.3 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::errorClass
- 8.908.2.4 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::messageID
- 8.908.2.5 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::msgDelFailureCause
- 8.908.2.6 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::msgDelFailureType
- 8.908.2.7 uint8_t* unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::pAlphaID
- 8.908.2.8 swi_uint256_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::ParamPresenceMask
- 8.908.2.9 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::RPCause
- 8.908.2.10 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::sendStatus
- 8.908.2.11 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::TPCause
- 8.908.2.12 uint32_t unpack_sms_SLQSWmsAsyncRawSendCallback_ind_t::userData

8.909 unpack_sms_SLQSWmsMemoryFullCallback_ind_t Struct Reference

Data Fields

- uint8_t [storageType](#)
- uint8_t [messageMode](#)
- swi_uint256_t [ParamPresenceMask](#)

8.909.1 Detailed Description

This structure contains unpack SMS memory full callback indication.

Parameters

<i>storageType</i>	<ul style="list-style-type: none"> • SMS message storage type <ul style="list-style-type: none"> – 0 - UIM - Invalid in case of CDMA device that does not require SIM – 1 - NV • Bit to check in ParamPresenceMask - 1
<i>messageMode</i>	<ul style="list-style-type: none"> • 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS) • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.909.2 Field Documentation

8.909.2.1 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode`

8.909.2.2 `swi_uint256_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::ParamPresenceMask`

8.909.2.3 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType`

8.910 `unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t` Struct Reference

Data Fields

- `uint8_t numInstances`
- `sms_messageWaitingInfoContent msgWaitInfo` [0xFF]
- `swi_uint256_t ParamPresenceMask`

8.910.1 Detailed Description

This structure holds information related to message waiting information indication

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • Number of sets of the elements in structure <code>sms_messageWaitingInfoContent</code> • Bit to check in ParamPresenceMask - 1
<i>msgWaitInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of <code>sms_messageWaitingInfoContent</code>. <ul style="list-style-type: none"> – See <code>sms_messageWaitingInfoContent</code> for more information. • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.910.2 Field Documentation

8.910.2.1 `sms_messageWaitingInfoContent unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::msgWaitInfo`[0xFF]

8.910.2.2 `uint8_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::numInstances`

8.910.2.3 `swi_uint256_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::ParamPresenceMask`

8.911 `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](#)

8.911.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile unpack parameters.

Parameters

<i>Profile</i>	<ul style="list-style-type: none"> • Audio Profile <ul style="list-style-type: none"> – 0-5 • Bit to check in ParamPresenceMask - 1
<i>EarMute</i>	<ul style="list-style-type: none"> • Ear Mute <ul style="list-style-type: none"> – 0 - Mute – 1 - Unmute • Bit to check in ParamPresenceMask - 2
<i>MicMute</i>	<ul style="list-style-type: none"> • MIC Mute <ul style="list-style-type: none"> – 0 - Mute – 1 - Unmute • Bit to check in ParamPresenceMask - 3
<i>Generator</i>	<ul style="list-style-type: none"> • Generator <ul style="list-style-type: none"> – 0 - Voice • Bit to check in ParamPresenceMask - 4
<i>Volume</i>	<ul style="list-style-type: none"> • RX volume level <ul style="list-style-type: none"> – 0-5 • Bit to check in ParamPresenceMask - 5
<i>CwtMute</i>	<ul style="list-style-type: none"> • Call waiting tone Mute <ul style="list-style-type: none"> – 0 - Mute – 1 - Unmute • Bit to check in ParamPresenceMask - 6
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.911.2 Field Documentation

8.911.2.1 `uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::CwtMute`

8.911.2.2 `uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::EarMute`

8.911.2.3 `uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::Generator`

8.911.2.4 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::MicMute`

8.911.2.5 `swi_uint256_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::ParamPresenceMask`

8.911.2.6 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::Profile`

8.911.2.7 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::Volume`

8.912 `unpack_swiaudio_SLQSGetM2MAudioVolume_t` Struct Reference

Data Fields

- `uint8_t` [Level](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.912.1 Detailed Description

This structure contains the `SLQSGetM2MAudioVolume` unpack parameters.

Parameters

<i>Level</i>	<ul style="list-style-type: none"> • The RX Volume Level <ul style="list-style-type: none"> – 0-5 • Bit to check in <code>ParamPresenceMask</code> - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.912.2 Field Documentation

8.912.2.1 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioVolume_t::Level`

8.912.2.2 `swi_uint256_t` `unpack_swiaudio_SLQSGetM2MAudioVolume_t::ParamPresenceMask`

8.913 `unpack_swiaudio_SLQSGetM2MAVMute_t` Struct Reference

Data Fields

- `uint8_t` [EarMute](#)
- `uint8_t` [MicMute](#)
- `uint8_t` [CwtMute](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.913.1 Detailed Description

This structure contains the `SLQSGetM2MAVMute` unpack parameters.

Parameters

<i>pEarMute</i>	<ul style="list-style-type: none"> • Ear Mute <ul style="list-style-type: none"> – 0-Mute – 1-UnMute • Bit to check in ParamPresenceMask - 1
<i>pMicMute</i>	<ul style="list-style-type: none"> • Mic Mute <ul style="list-style-type: none"> – 0-Mute – 1-unmute • Bit to check in ParamPresenceMask - 2
<i>CwtMute</i>	<ul style="list-style-type: none"> • Waiting tone Mute <ul style="list-style-type: none"> – 0-5 • Bit to check in ParamPresenceMask - 3
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.913.2 Field Documentation

8.913.2.1 uint8_t unpack_swiaudio_SLQSGetM2MAVMute_t::CwtMute

8.913.2.2 uint8_t unpack_swiaudio_SLQSGetM2MAVMute_t::EarMute

8.913.2.3 uint8_t unpack_swiaudio_SLQSGetM2MAVMute_t::MicMute

8.913.2.4 swi_uint256_t unpack_swiaudio_SLQSGetM2MAVMute_t::ParamPresenceMask

8.914 unpack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference

Data Fields

- uint16_t [Value](#)
- [swi_uint256_t ParamPresenceMask](#)

8.914.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain unpack parameters.

Parameters

<i>Value</i>	<ul style="list-style-type: none"> • RX speakerphone gain <ul style="list-style-type: none"> – 0x0 - 0x7fff • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.914.2 Field Documentation

8.914.2.1 `swi_uint256_t unpack_swiaudio_SLQSGetM2MSpkrGain_t::ParamPresenceMask`

8.914.2.2 `uint16_t unpack_swiaudio_SLQSGetM2MSpkrGain_t::Value`

8.915 `unpack_swiaavs_SLQSAVMSEventReportInd_t` Struct Reference

Data Fields

- [UnpackSwiAvmsEventReportBinaryUpdateSessionInfo](#) * `pBinaryUpdateSessionInfo`
- [UnpackSwiAvmsEventReportConfig](#) * `pConfig`
- [UnpackSwiAvmsEventReportNotification](#) * `pNotification`
- [UnpackSwiAvmsEventReportPackageID](#) * `pPackageID`
- [UnpackSwiAvmsEventReportConnectionRequest](#) * `pConnectionRequest`
- [UnpackSwiAvmsEventReportWAMSPParamChange](#) * `pWAMSPParamChanged`
- [UnpackSwiAvmsEventReportRegStatus](#) * `pRegStatus`
- [UnpackSwiAvmsEventReportDataSessionStatus](#) * `pDataSessionStatus`
- [UnpackSwiAvmsEventReportSessionType](#) * `pSessionType`
- [UnpackSwiAvmsEventReportHTTPStatus](#) * `pHTTPStatus`
- `uint32_t resultCode`
- `swi_uint256_t ParamPresenceMask`

8.915.1 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"> • See UnpackSwiAvmsEventReportBinaryUpdateSessionInfo for more information • Bit to check in ParamPresenceMask - 16
<i>pConfig</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportConfig for more information • Bit to check in ParamPresenceMask - 17
<i>pNotification</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportNotification for more information • Bit to check in ParamPresenceMask - 18
<i>pPackageID</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportPackageID for more information • Bit to check in ParamPresenceMask - 21
<i>pConnectionRequest</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportConnectionRequest for more information • Bit to check in ParamPresenceMask - 19
<i>pWAMSPParamChanged</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportWAMSPParamChange for more information • Bit to check in ParamPresenceMask - 20

<i>pRegStatus[OUT]</i>	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportRegStatus for more information • Bit to check in ParamPresenceMask - 22
<i>pDataSession-Status[OUT]</i>	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportDataSessionStatus for more information • Bit to check in ParamPresenceMask - 23
<i>pSessionType[OUT]</i>	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportSessionType for more information • Bit to check in ParamPresenceMask - 24
<i>pHTTPStatus[OUT]</i>	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportHTTPStatus for more information • Bit to check in ParamPresenceMask - 25
<i>resultcode</i>	- Tlv Result Code.

8.915.2 Field Documentation

8.915.2.1 `swi_uint256_t unpack_swiavms_SLQSAVMSEventReportInd_t::ParamPresenceMask`

8.915.2.2 `UnpackSwiAvmsEventReportBinaryUpdateSessionInfo* unpack_swiavms_SLQSAVMSEventReportInd_t::pBinaryUpdateSessionInfo`

8.915.2.3 `UnpackSwiAvmsEventReportConfig* unpack_swiavms_SLQSAVMSEventReportInd_t::pConfig`

8.915.2.4 `UnpackSwiAvmsEventReportConnectionRequest* unpack_swiavms_SLQSAVMSEventReportInd_t::pConnectionRequest`

8.915.2.5 `UnpackSwiAvmsEventReportDataSessionStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pDataSessionStatus`

8.915.2.6 `UnpackSwiAvmsEventReportHTTPStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pHTTPStatus`

8.915.2.7 `UnpackSwiAvmsEventReportNotification* unpack_swiavms_SLQSAVMSEventReportInd_t::pNotification`

8.915.2.8 `UnpackSwiAvmsEventReportPackageID* unpack_swiavms_SLQSAVMSEventReportInd_t::pPackageID`

8.915.2.9 `UnpackSwiAvmsEventReportRegStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pRegStatus`

8.915.2.10 `UnpackSwiAvmsEventReportSessionType* unpack_swiavms_SLQSAVMSEventReportInd_t::pSessionType`

8.915.2.11 `UnpackSwiAvmsEventReportWAMSParamChange* unpack_swiavms_SLQSAVMSEventReportInd_t::pWAMSParaChanged`

8.915.2.12 `uint32_t unpack_swiavms_SLQSAVMSEventReportInd_t::resultcode`

8.916 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](#)

8.916.1 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"> • 0 - Disabled. • 1 - Enabled.
----------------------	---

- Bit to check in ParamPresenceMask - **1**

Parameters

<i>AutoConnect</i>	- Auto Connect. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.
--------------------	---

- Bit to check in ParamPresenceMask - **6**

Parameters

<i>AutoReboot</i>	- Auto Reboot. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.
-------------------	--

- Bit to check in ParamPresenceMask - **21**

Parameters

<i>Fw-Autodownload</i>	- Firmware auto download. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled. – Bit to check in ParamPresenceMask - 3
<i>FwAutoUpdate</i>	- Firmware auto update. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.

- Bit to check in ParamPresenceMask - 4

Parameters

<i>FwAutoSDM</i>	- OMA Automatic UI Alert Response. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.
------------------	--

- Bit to check in ParamPresenceMask - 5

Parameters

<i>pPollingTimer</i>	- Polling timer to connect to AVMS server. <ul style="list-style-type: none"> • 0-525600 (min) • 0:disabled
----------------------	---

- Bit to check in ParamPresenceMask - 16

Parameters

<i>pConnection-RetryTimers</i>	- Connection Retry timers. <ul style="list-style-type: none"> • See PackSwiAVMSSettingsConnectionRetryTimers for more information. • Bit to check in ParamPresenceMask - 17
<i>pAPNInfo</i>	- APN Information. <ul style="list-style-type: none"> • See PackSwiAVMSSettingsAPNInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>pNotification-Store</i>	- Notification Storing When Disabled or Offline. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.

- Bit to check in ParamPresenceMask - 19

Parameters

<i>pPeroidsInfo</i>	- Min and Max Period of an Observation. <ul style="list-style-type: none"> • See PackSwiAVMSSettingsPeriodsInfo for more information. • Bit to check in ParamPresenceMask - 20
<i>resultcode</i>	- Tlv Result Code.

8.916.2 Field Documentation

8.916.2.1 uint8_t unpack_swiaavms_SLQSAVMSGetSettings_t::AutoConnect

8.916.2.2 uint8_t unpack_swiaavms_SLQSAVMSGetSettings_t::AutoReboot

8.916.2.3 uint8_t unpack_swiaavms_SLQSAVMSGetSettings_t::FwAutoSDM

- 8.916.2.4 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwPromptdownload`
- 8.916.2.5 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwPromptUpdate`
- 8.916.2.6 `uint32_t unpack_swiavms_SLQSAVMSGetSettings_t::OMADMEEnabled`
- 8.916.2.7 `PackSwiAVMSSettingsAPNInfo* unpack_swiavms_SLQSAVMSGetSettings_t::pAPNInfo`
- 8.916.2.8 `swi_uint256_t unpack_swiavms_SLQSAVMSGetSettings_t::ParamPresenceMask`
- 8.916.2.9 `PackSwiAVMSSettingsConnectionRetryTimers* unpack_swiavms_SLQSAVMSGetSettings_t::pConnection-RetryTimers`
- 8.916.2.10 `uint8_t* unpack_swiavms_SLQSAVMSGetSettings_t::pNotificationStore`
- 8.916.2.11 `PackSwiAVMSSettingsPeriodsInfo* unpack_swiavms_SLQSAVMSGetSettings_t::pPeroidsInfo`
- 8.916.2.12 `uint32_t* unpack_swiavms_SLQSAVMSGetSettings_t::pPollingTimer`
- 8.916.2.13 `uint32_t unpack_swiavms_SLQSAVMSGetSettings_t::resultcode`

8.917 `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`

8.917.1 Detailed Description

Structure that contains the response for AVMS Get Settings command It maps with SLQSAVMSSetSettings_v2 (For AVC2 service)

Parameters

<i>OMADMEEnabled</i>	- OMA DM Enabled. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.
----------------------	---

- Bit to check in ParamPresenceMask - 1

Parameters

<i>AutoConnect</i>	<ul style="list-style-type: none"> - Auto Connect. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.
--------------------	--

- Bit to check in ParamPresenceMask - **6**

Parameters

<i>Fw-Promptdownload</i>	<ul style="list-style-type: none"> - Firmware Prompt download. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled. <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 3
<i>FwPrompt-Update</i>	<ul style="list-style-type: none"> - Firmware Prompt update. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.

- Bit to check in ParamPresenceMask - **4**

Parameters

<i>FwAutoSDM</i>	<ul style="list-style-type: none"> - OMA Automatic UI Alert Response. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled.
------------------	---

- Bit to check in ParamPresenceMask - **5**

Parameters

<i>pPollingTimer[OPTIONAL]</i>	<ul style="list-style-type: none"> - Polling timer to connect to AVMS server. <ul style="list-style-type: none"> • 0-525600 (min) • 0:disabled <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 16
<i>pConnection-RetryTimers[OPTIONAL]</i>	<ul style="list-style-type: none"> - Connection Retry timers. <ul style="list-style-type: none"> • See PackSwiAVMSSettingsConnectionRetryTimers for more information. • Bit to check in ParamPresenceMask - 17
<i>pAPNInfo[OPTIONAL]</i>	<ul style="list-style-type: none"> - APN Information. <ul style="list-style-type: none"> • See PackSwiAVMSSettingsAPNInfo for more information. • Bit to check in ParamPresenceMask - 18
<i>pNotification-Store[OPTIONAL]</i>	<ul style="list-style-type: none"> - Notification Storing When Disabled or Offline. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled. <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 19

<i>pPeriodsInfo</i> [OPTIONAL]	- Min and Max Period of an Observation. <ul style="list-style-type: none"> • See PackSwiAVMSSettingsPeriodsInfo for more information. • Bit to check in ParamPresenceMask - 20
<i>pAutoReboot</i> [OPTIONAL]	- Auto Reboot. <ul style="list-style-type: none"> • 0 - Disabled. • 1 - Enabled. <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 21
<i>resultcode</i>	- Tlv Result Code.

8.917.2 Field Documentation

- 8.917.2.1 `uint8_t unpack_swiavms_SLQSAVMSSettings_v2_t::AutoConnect`
- 8.917.2.2 `uint8_t unpack_swiavms_SLQSAVMSSettings_v2_t::FwAutoSDM`
- 8.917.2.3 `uint8_t unpack_swiavms_SLQSAVMSSettings_v2_t::FwPromptdownload`
- 8.917.2.4 `uint8_t unpack_swiavms_SLQSAVMSSettings_v2_t::FwPromptUpdate`
- 8.917.2.5 `uint32_t unpack_swiavms_SLQSAVMSSettings_v2_t::OMADMEEnabled`
- 8.917.2.6 `PackSwiAVMSSettingsAPNInfo* unpack_swiavms_SLQSAVMSSettings_v2_t::pAPNInfo`
- 8.917.2.7 `swi_uint256_t unpack_swiavms_SLQSAVMSSettings_v2_t::ParamPresenceMask`
- 8.917.2.8 `uint8_t* unpack_swiavms_SLQSAVMSSettings_v2_t::pAutoReboot`
- 8.917.2.9 `PackSwiAVMSSettingsConnectionRetryTimers* unpack_swiavms_SLQSAVMSSettings_v2_t::pConnectionRetryTimers`
- 8.917.2.10 `uint8_t* unpack_swiavms_SLQSAVMSSettings_v2_t::pNotificationStore`
- 8.917.2.11 `PackSwiAVMSSettingsPeriodsInfo* unpack_swiavms_SLQSAVMSSettings_v2_t::pPeriodsInfo`
- 8.917.2.12 `uint32_t* unpack_swiavms_SLQSAVMSSettings_v2_t::pPollingTimer`
- 8.917.2.13 `uint32_t unpack_swiavms_SLQSAVMSSettings_v2_t::resultcode`

8.918 unpack_swiavms_SLQSAVMSSendSelection_t Struct Reference

Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

8.918.1 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
-------------------	---

8.918.2 Field Documentation

8.918.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSSendSelection_t::ParamPresenceMask

8.918.2.2 uint32_t unpack_swiavms_SLQSAVMSSendSelection_t::resultcode

8.919 unpack_swiavms_SLQSAVMSSessionGetInfo_t Struct Reference

Data Fields

- [UnpackSwiAvmsEventReportBinaryUpdateSessionInfo](#) * pBinaryUpdateSessionInfo
- [UnpackSwiAvmsEventReportConfig](#) * pConfig
- [UnpackSwiAvmsEventReportNotification](#) * pNotification
- [UnpackSwiAvmsEventReportPackageID](#) * pPackageID
- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

8.919.1 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"> • See UnpackSwiAvmsEventReportBinaryUpdateSessionInfo for more information • Bit to check in ParamPresenceMask - 16
<i>pConfig</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportConfig for more information • Bit to check in ParamPresenceMask - 17
<i>pNotification</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportNotification for more information • Bit to check in ParamPresenceMask - 18
<i>pPackageID</i> [OUT]	<ul style="list-style-type: none"> • See UnpackSwiAvmsEventReportPackageID for more information • Bit to check in ParamPresenceMask - 19
<i>resultcode</i>	- Tlv Result Code.

8.919.2 Field Documentation

8.919.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSSessionGetInfo_t::ParamPresenceMask

8.919.2.2 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo* unpack_swiavms_SLQSAVMSSessionGetInfo_t::pBinaryUpdateSessionInfo

8.919.2.3 **UnpackSwiAvmsEventReportConfig*** `unpack_swiavms_SLQSAVMSSessionGetInfo_t::pConfig`

8.919.2.4 **UnpackSwiAvmsEventReportNotification*** `unpack_swiavms_SLQSAVMSSessionGetInfo_t::pNotification`

8.919.2.5 **UnpackSwiAvmsEventReportPackageID*** `unpack_swiavms_SLQSAVMSSessionGetInfo_t::pPackageID`

8.919.2.6 `uint32_t unpack_swiavms_SLQSAVMSSessionGetInfo_t::resultcode`

8.920 `unpack_swiavms_SLQSAvmsSetEventReport_t` Struct Reference

Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

8.920.1 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"> • Bit to check in ParamPresenceMask - 2
-------------------	--

8.920.2 Field Documentation

8.920.2.1 `swi_uint256_t unpack_swiavms_SLQSAvmsSetEventReport_t::ParamPresenceMask`

8.920.2.2 `uint32_t unpack_swiavms_SLQSAvmsSetEventReport_t::resultcode`

8.921 `unpack_swiavms_SLQSAVMSSetSettings_t` Struct Reference

Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

8.921.1 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"> • Bit to check in ParamPresenceMask - 2
-------------------	--

8.921.2 Field Documentation

8.921.2.1 `swi_uint256_t unpack_swiavms_SLQSAVMSSetSettings_t::ParamPresenceMask`

8.921.2.2 uint32_t unpack_swiavms_SLQSAVMSSetSettings_t::resultcode

8.922 unpack_swiavms_SLQSAVMSSetSettings_v2_t Struct Reference

Data Fields

- uint32_t [resultcode](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.922.1 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">• Bit to check in ParamPresenceMask - 2
-------------------	---

8.922.2 Field Documentation

8.922.2.1 [swi_uint256_t](#) unpack_swiavms_SLQSAVMSSetSettings_v2_t::ParamPresenceMask

8.922.2.2 uint32_t unpack_swiavms_SLQSAVMSSetSettings_v2_t::resultcode

8.923 unpack_swiavms_SLQSAVMSStartSession_t Struct Reference

Data Fields

- uint32_t [sessionResponse](#)
- uint32_t [resultcode](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.923.1 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">• 0x00000001 - Available.• 0x00000002 - Not Available.• 0x00000003 - Check Timed Out.• Bit to check in ParamPresenceMask - 1
<i>resultcode</i>	- Tlv Result Code.

8.923.2 Field Documentation

8.923.2.1 [swi_uint256_t](#) unpack_swiavms_SLQSAVMSStartSession_t::ParamPresenceMask

8.923.2.2 `uint32_t unpack_swiavms_SLQSAVMSStartSession_t::resultcode`

8.923.2.3 `uint32_t unpack_swiavms_SLQSAVMSStartSession_t::sessionResponse`

8.924 `unpack_swiavms_SLQSAVMSStopSession_t` Struct Reference

Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

8.924.1 Detailed Description

Structure that contains the session type response for AVMS Stop session command

Parameters

<i>resultcode</i>	<ul style="list-style-type: none"> - Tlv Result Code. • Bit to check in ParamPresenceMask - 2
-------------------	--

8.924.2 Field Documentation

8.924.2.1 `swi_uint256_t unpack_swiavms_SLQSAVMSStopSession_t::ParamPresenceMask`

8.924.2.2 `uint32_t unpack_swiavms_SLQSAVMSStopSession_t::resultcode`

8.925 `unpack_swidms_SLQSSwiDmsGetHWWatchdog_t` Struct Reference

Data Fields

- `swidms_SwiDmsGetHWWatchdog * pHWWatchdog`
- `swi_uint256_t ParamPresenceMask`

8.925.1 Detailed Description

This structure contains the get hw watchdog response

Parameters

<i>pHWWatchdog</i>	<ul style="list-style-type: none"> • See <code>swidms_SwiDmsGetHWWatchdog</code> for more information • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.925.2 Field Documentation

8.925.2.1 `swi_uint256_t unpack_swidms_SLQSSwiDmsGetHWWatchdog_t::ParamPresenceMask`

8.925.2.2 swidms_SwiDmsGetHWWatchdog* unpack_swidms_SLQSSwiDmsGetHWWatchdog_t::pHWWatchdog

8.926 unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference

Data Fields

- [swidms_mtuSize3gppTlv](#) * [pMTUSize3gpp](#)
- [swidms_hrpdMTUSizeTlv](#) * [pHrpdMTUSize](#)
- [swidms_ehrpdMTUSizeTlv](#) * [pEhrpdMTUSize](#)
- [swidms_usbMTUSizeTlv](#) * [pUsbMTUSize](#)
- [swi_uint256_t](#) ParamPresenceMask

8.926.1 Detailed Description

This structure contains the Get MTU Response parameter.

Parameters

<i>pMTUSize3gpp</i>	[Optional] <ul style="list-style-type: none"> • See swidms_mtuSize3gppTlv for more information • Bit to check in ParamPresenceMask - 16
<i>pHrpdMTUSize</i>	[Optional] <ul style="list-style-type: none"> • See swidms_hrpdMTUSizeTlv for more information • Bit to check in ParamPresenceMask - 17
<i>pEhrpdMTUSize</i>	[Optional] <ul style="list-style-type: none"> • See swidms_ehrpdMTUSizeTlv for more information • Bit to check in ParamPresenceMask - 18
<i>pUsbMTUSize</i>	[Optional] <ul style="list-style-type: none"> • See swidms_usbMTUSizeTlv for more information • Bit to check in ParamPresenceMask - 19
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.926.2 Field Documentation

8.926.2.1 swi_uint256_t unpack_swidms_SLQSSwiDmsGetMTU_t::ParamPresenceMask

8.926.2.2 swidms_ehrpdMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pEhrpdMTUSize

8.926.2.3 swidms_hrpdMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pHrpdMTUSize

8.926.2.4 swidms_mtuSize3gppTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pMTUSize3gpp

8.926.2.5 swidms_usbMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pUsbMTUSize

8.927 unpack_swidms_SLQSSwiDmsGetSecureInfo_t Struct Reference

Data Fields

- uint8_t [secureBootEnabled](#)
- uint8_t [memoryDumpAllowed](#)
- uint8_t [jtagAccessAllowed](#)
- uint16_t [TlvResult](#)
- swi_uint256_t [ParamPresenceMask](#)

8.927.1 Detailed Description

This structure contains secure boot config and other capabilities

Parameters

<i>secureBoot-Enabled</i>	<ul style="list-style-type: none"> • secure boot enabled or disabled 0- disabled 1- enabled • Bit to check in ParamPresenceMask - 1
<i>memoryDump-Allowed</i>	<ul style="list-style-type: none"> • memory dump allowed or disallowed 0- disallowed 1- allowed • Bit to check in ParamPresenceMask - 3
<i>jtagAccess-Allowed</i>	<ul style="list-style-type: none"> • Jtag access allowed or disallowed 0- disallowed 1- allowed • Bit to check in ParamPresenceMask - 4
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv result • Bit to check in ParamPresenceMask - 2
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.927.2 Field Documentation

8.927.2.1 uint8_t [unpack_swidms_SLQSSwiDmsGetSecureInfo_t::jtagAccessAllowed](#)

8.927.2.2 uint8_t [unpack_swidms_SLQSSwiDmsGetSecureInfo_t::memoryDumpAllowed](#)

8.927.2.3 swi_uint256_t [unpack_swidms_SLQSSwiDmsGetSecureInfo_t::ParamPresenceMask](#)

8.927.2.4 uint8_t [unpack_swidms_SLQSSwiDmsGetSecureInfo_t::secureBootEnabled](#)

8.927.2.5 uint16_t [unpack_swidms_SLQSSwiDmsGetSecureInfo_t::TlvResult](#)

8.928 unpack_swidms_SLQSSwiDmsGetUsbComp_t Struct Reference

Data Fields

- [swidms_interfaceCfgTlv](#) * [pInterfaceCfg](#)
- [swidms_supportedIntBitmaskTlv](#) * [pSupportedBitmasks](#)
- swi_uint256_t [ParamPresenceMask](#)

8.928.1 Detailed Description

This structure contains the get usb composition response

Parameters

<i>pInterfaceCfg</i>	<ul style="list-style-type: none"> See swidms_ifaceCfgTlv for more information Bit to check in ParamPresenceMask - 1
<i>pSupported-Bitmaps</i>	[Optional] <ul style="list-style-type: none"> See swidms_supportedIntBitmaskTlv for more information Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.928.2 Field Documentation

8.928.2.1 `swi_uint256_t unpack_swidms_SLQSSwiDmsGetUsbComp_t::ParamPresenceMask`

8.928.2.2 `swidms_ifaceCfgTlv* unpack_swidms_SLQSSwiDmsGetUsbComp_t::pInterfaceCfg`

8.928.2.3 `swidms_supportedIntBitmaskTlv* unpack_swidms_SLQSSwiDmsGetUsbComp_t::pSupportedBitmaps`

8.929 unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference

Data Fields

- `uint8_t usbNetNum`
- `swi_uint256_t ParamPresenceMask`

8.929.1 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"> value of usb net numbers on the device Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.929.2 Field Documentation

8.929.2.1 `swi_uint256_t unpack_swidms_SLQSSwiDmsGetUsbNetNum_t::ParamPresenceMask`

8.929.2.2 `uint8_t unpack_swidms_SLQSSwiDmsGetUsbNetNum_t::usbNetNum`

8.930 unpack_swidms_SLQSSwiDmsSetHWWatchdog_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- swi_uint256_t [ParamPresenceMask](#)

8.930.1 Detailed Description

This structure is used to store [unpack_swidms_SLQSSwiDmsSetHWWatchdog_t](#) parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv result • Bit to check in ParamPresenceMask - 2
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.930.2 Field Documentation

8.930.2.1 [swi_uint256_t](#) [unpack_swidms_SLQSSwiDmsSetHWWatchdog_t::ParamPresenceMask](#)

8.930.2.2 [uint16_t](#) [unpack_swidms_SLQSSwiDmsSetHWWatchdog_t::Tlvresult](#)

8.931 unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- swi_uint256_t [ParamPresenceMask](#)

8.931.1 Detailed Description

This structure is used to store [unpack_swidms_SLQSSwiDmsSetMTU_t](#) parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack Tlv result • Bit to check in ParamPresenceMask - 2
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.931.2 Field Documentation

8.931.2.1 [swi_uint256_t](#) [unpack_swidms_SLQSSwiDmsSetMTU_t::ParamPresenceMask](#)

8.931.2.2 [uint16_t unpack_swidms_SLQSSwiDmsSetMTU_t::Tlvresult](#)

8.932 unpack_swidms_SLQSSwiDmsSetUsbComp_t Struct Reference

Data Fields

- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.932.1 Detailed Description

This structure is used to store [unpack_swidms_SLQSSwiDmsSetUsbComp_t](#) parameters.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack Tlv result• Bit to check in ParamPresenceMask - 2
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.932.2 Field Documentation

8.932.2.1 [swi_uint256_t unpack_swidms_SLQSSwiDmsSetUsbComp_t::ParamPresenceMask](#)

8.932.2.2 [uint16_t unpack_swidms_SLQSSwiDmsSetUsbComp_t::Tlvresult](#)

8.933 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](#)

8.933.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

Parameters

<i>function</i>	<ul style="list-style-type: none"> • Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> – 0 - disabled – 1 - At bootup – 2 - When NMEA port is opened • Bit to check in ParamPresenceMask - 16
<i>function_ - reported</i>	<ul style="list-style-type: none"> • 0 - not reported by modem • 1 - reported by modem
<i>fix_type</i>	<ul style="list-style-type: none"> • Type of GNSS fix: <ul style="list-style-type: none"> – 1 - Default Engine mode – 2 - MS-Based – 3 - MS-Assisted – 4 - Standalone • Bit to check in ParamPresenceMask - 17
<i>fix_type_ - reported</i>	<ul style="list-style-type: none"> • 0 - not reported by modem • 1 - reported by modem
<i>max_time</i>	<ul style="list-style-type: none"> • Maximum time allowed for the receiver to get a fix in seconds • Valid range: 1-255 • Bit to check in ParamPresenceMask - 18
<i>max_time_ - reported</i>	<ul style="list-style-type: none"> • 0 - not reported by modem • 1 - reported by modem
<i>max_dist</i>	<ul style="list-style-type: none"> • Maximum uncertainty of a fix measured by distance in meters • Valid range: 1 - 4294967280 • Bit to check in ParamPresenceMask - 19
<i>max_dist_ - reported</i>	<ul style="list-style-type: none"> • 0 - not reported by modem • 1 - reported by modem
<i>fix_rate</i>	<ul style="list-style-type: none"> • Time between fixes in seconds • Valid range: 1–65535 • Bit to check in ParamPresenceMask - 20
<i>fix_rate_ - reported</i>	<ul style="list-style-type: none"> • 0 - not reported by modem • 1 - reported by modem

8.933.2 Field Documentation

- 8.933.2.1 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate`
- 8.933.2.2 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported`
- 8.933.2.3 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type`
- 8.933.2.4 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported`
- 8.933.2.5 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function`
- 8.933.2.6 `int unpack_swiloc_SwiLocGetAutoStart_t::function_reported`
- 8.933.2.7 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist`
- 8.933.2.8 `int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported`
- 8.933.2.9 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time`
- 8.933.2.10 `int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported`
- 8.933.2.11 `swi_uint256_t unpack_swiloc_SwiLocGetAutoStart_t::ParamPresenceMask`

8.934 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`

8.934.1 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"> • 0x00 - SWIOMA-DM FOTA • 0x01 - SWIOMA-DM Config • 0x02 - SWIOMA-DM Notification • 0xff - indication missing event information.
<i>SessionInfo-Fota[OUT]</i>	<ul style="list-style-type: none"> • See unpack_omaDmFotaTlv_t for more information • Bit to check in ParamPresenceMask - 16
<i>SessionInfo-Config[OUT]</i>	<ul style="list-style-type: none"> • See unpack_omaDmConfigTlv_t for more information • Bit to check in ParamPresenceMask - 17

<i>SessionInfo-Notification[OUT]</i>	<ul style="list-style-type: none"> • See unpack_omaDmNotificationsTlv_t for more information • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.934.2 Field Documentation

8.934.2.1 `uint32_t unpack_swioma_SLQSOMADMAAlertCallback_ind_t::eventType`

8.934.2.2 `swi_uint256_t unpack_swioma_SLQSOMADMAAlertCallback_ind_t::ParamPresenceMask`

8.934.2.3 `unpack_omaDmConfigTlv_t unpack_swioma_SLQSOMADMAAlertCallback_ind_t::SessionInfoConfig`

8.934.2.4 `unpack_omaDmFotaTlv_t unpack_swioma_SLQSOMADMAAlertCallback_ind_t::SessionInfoFota`

8.934.2.5 `unpack_omaDmNotificationsTlv_t unpack_swioma_SLQSOMADMAAlertCallback_ind_t::SessionInfoNotification`

8.935 unpack_swioma_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`

8.935.1 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"> • 1 Byte parameter indicating status <ul style="list-style-type: none"> – 0x01 - No Firmware available – 0x02 - Query Firmware Download – 0x03 - Firmware Downloading – 0x04 - Firmware Downloaded – 0x05 - Query Firmware Update – 0x06 - Firmware Updating – 0x07 - Firmware Updated • Bit to check in ParamPresenceMask - 16
<i>Update-CompleteStatus</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Update Complete Status <ul style="list-style-type: none"> – See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status • Bit to check in ParamPresenceMask - 16
<i>Severity</i>	<ul style="list-style-type: none"> • 1 byte parameter indicating severity <ul style="list-style-type: none"> – 0x01 - Mandatory – 0x02 - Optional • Bit to check in ParamPresenceMask - 16
<i>SourceLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Vendor Name String in Bytes. • Bit to check in ParamPresenceMask - 16
<i>Source</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Vendor Name in ASCII • See LITEQMI_MAX_SWIOMA_STR_LEN for more information • Bit to check in ParamPresenceMask - 16
<i>PkgNameLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Name String in Bytes. • Bit to check in ParamPresenceMask - 16
<i>PkgName</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Name in ASCII • See LITEQMI_MAX_SWIOMA_STR_LEN for more information • Bit to check in ParamPresenceMask - 16
<i>PkgDescLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Description String in Bytes. • Bit to check in ParamPresenceMask - 16
<i>PkgDescription</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Description in ASCII • See LITEQMI_MAX_SWIOMA_STR_LEN for more information • Bit to check in ParamPresenceMask - 16

<i>DateLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Description String in Bytes. • Bit to check in ParamPresenceMask - 16
<i>Date</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Description in ASCII • See LITEQMI_MAX_SWIOMA_STR_LEN for more information • Bit to check in ParamPresenceMask - 16
<i>TimeLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Time String in Bytes. • Bit to check in ParamPresenceMask - 16
<i>Time</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Time String in ASCII • See LITEQMI_MAX_SWIOMA_STR_LEN for more information • Bit to check in ParamPresenceMask - 16
<i>SessionType</i>	<ul style="list-style-type: none"> • 1 byte parameter reflects the last session started for Sprint <ul style="list-style-type: none"> – 0x00 - No session since boot – 0x01 - Sprint CI-DC Session – 0x02 - Sprint CI-PRL Session – 0x03 - Sprint CI-FUMO Session – 0x04 - Sprint HFA-DC Session – 0x05 - Sprint HFA-PRL Session – 0x06 - Sprint HFA-FUMO Session – 0x07 - Sprint NI Session • Bit to check in ParamPresenceMask - 16
<i>SessionState</i>	<ul style="list-style-type: none"> • 1 byte parameter indicating session state <ul style="list-style-type: none"> – 0x01 - idle – 0x02 - active – 0x03 - pending • Bit to check in ParamPresenceMask - 16
<i>RetryCount</i>	<ul style="list-style-type: none"> • 1 byte parameter indicating retries left count <ul style="list-style-type: none"> – valid values 0 to 6 • Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.935.2 Field Documentation

8.935.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

- 8.935.2.2 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::DateLength`
- 8.935.2.3 `swi_uint256_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::ParamPresenceMask`
- 8.935.2.4 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgDescLength`
- 8.935.2.5 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgDescription[255]`
- 8.935.2.6 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgName[255]`
- 8.935.2.7 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgNameLength`
- 8.935.2.8 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::RetryCount`
- 8.935.2.9 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::SessionState`
- 8.935.2.10 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::SessionType`
- 8.935.2.11 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::Severity`
- 8.935.2.12 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::Source[255]`
- 8.935.2.13 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::SourceLength`
- 8.935.2.14 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::Status`
- 8.935.2.15 `uint8_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::Time[255]`
- 8.935.2.16 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::TimeLength`
- 8.935.2.17 `uint16_t` `unpack_swioama_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus`

8.936 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](#)

8.936.1 Detailed Description

Structure containing info for OMADM session Bit to check in ParamPresenceMask - 16

Parameters

<i>status</i>	<ul style="list-style-type: none"> • 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.
<i>sessionState</i>	<ul style="list-style-type: none"> • 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
<i>hfaStatus</i>	<ul style="list-style-type: none"> • 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
<i>hfaMaxRetry</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating max retry for HFA 0 – HFA completed
<i>hfaRetryInterval</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating retry interval in seconds for HFA 0 – HFA completed
<i>hfaRetryIndex</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating number of retry currently ongoing 0 – no retrying 0xFFFF- Invalid value
<i>fumoState</i>	<ul style="list-style-type: none"> • 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

<i>fumoResultCode</i>	<ul style="list-style-type: none"> • 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.
<i>pkgVendor-NameLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of package vendor String in WORDs
<i>pkgVendorName</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Name in UCS2 • size in bytes is 2*pkgVendorNameLength
<i>pkgSize</i>	<ul style="list-style-type: none"> • 4 byte parameter indicating pkg size
<i>pkgVersion-NameLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package version Name String in WORDs
<i>pkgVersion-Name</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package version Name in UCS2 • size in bytes is 2*pkgVersionNameLength
<i>pkgNameLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Name String in WORDs
<i>pkgName</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Name in UCS2 • size in bytes is 2*pkgNameLength
<i>pkgDescLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package description String in WORDs
<i>pkgDesc</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package description in UCS2 • size in bytes is 2*pkgDescLength
<i>pkgDateLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package date String in WORDs
<i>pkgDate</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package date in UCS2 • size in bytes is 2*pkgDateLength
<i>pkgInstallTime-Length</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package install time String in WORDs

<i>pkgInstallTime</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package install time in UCS2 • size in bytes is 2*pkgInstallTimeLength
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.936.2 Field Documentation

8.936.2.1 uint32_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::fumoResultCode

8.936.2.2 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::fumoState

8.936.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::hfaMaxRetry

8.936.2.4 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::hfaRetryIndex

8.936.2.5 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::hfaRetryInterval

8.936.2.6 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::hfaStatus

8.936.2.7 swi_uint256_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::ParamPresenceMask

8.936.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgDate[512]

8.936.2.9 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgDateLength

8.936.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgDesc[512]

8.936.2.11 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgDescLength

8.936.2.12 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgInstallTime[512]

8.936.2.13 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgInstallTimeLength

8.936.2.14 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgName[512]

8.936.2.15 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgNameLength

8.936.2.16 uint32_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgSize

8.936.2.17 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgVendorName[512]

8.936.2.18 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgVendorNameLength

8.936.2.19 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgVersionName[512]

8.936.2.20 uint16_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::pkgVersionNameLength

8.936.2.21 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::sessionState

8.936.2.22 uint8_t unpack_swioma_SLQSOMADMGetSessionInfoExt_t::status

8.937 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](#)

8.937.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

Parameters

<i>OMADMEEnabled</i>	[out] <ul style="list-style-type: none"> • Optional 4 byte parameter indicating OMADM service enabled <ul style="list-style-type: none"> – 0x00000001 - Client-initiated device configuration – 0x00000002 - Network-initiated device configuration – 0x00000010 - Client-initiated FUMO – 0x00000020 - Network-initiated FUMO • Bit to check in ParamPresenceMask - 16
<i>FOTAdownload</i>	[out] <ul style="list-style-type: none"> • Optional 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> – 0x00 - Host permission required before downloading – 0x01 - Automatically start downloading, no host permission required – 0x02 - Automatically start downloading, while not roaming – 0x03 - Automatically reject download – 0x04 - Automatically reject download with “Enterprise Reject Policy • Bit to check in ParamPresenceMask - 17
<i>FOTAUpdate</i>	[out] <ul style="list-style-type: none"> • Optional 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> – 0x00 - User permission required before updating firmware – 0x01 - No user permission required before updating firmware – 0x02 - User permission required, auto update on power up • Bit to check in ParamPresenceMask - 18
<i>Autosdm</i>	[out] <ul style="list-style-type: none"> • Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject • Bit to check in ParamPresenceMask - 19

<i>FwAutoCheck</i>	[out] <ul style="list-style-type: none"> Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> 0x00 - Disabled 0x01 - Enabled Bit to check in ParamPresenceMask - 20
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.937.2 Field Documentation

8.937.2.1 uint8_t unpack_swioama_SLQSOMADMGetSettings_t::Autosdm

8.937.2.2 uint8_t unpack_swioama_SLQSOMADMGetSettings_t::FOTAdownload

8.937.2.3 uint8_t unpack_swioama_SLQSOMADMGetSettings_t::FOTAUpdate

8.937.2.4 uint8_t unpack_swioama_SLQSOMADMGetSettings_t::FwAutoCheck

8.937.2.5 uint32_t unpack_swioama_SLQSOMADMGetSettings_t::OMADMEEnabled

8.937.2.6 swi_uint256_t unpack_swioama_SLQSOMADMGetSettings_t::ParamPresenceMask

8.938 unpack_swioama_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [FwAvailability](#)
- swi_uint256_t [ParamPresenceMask](#)

8.938.1 Detailed Description

Structure that contains the responses for OMA start session command

Parameters

<i>FwAvailability</i>	[out] <ul style="list-style-type: none"> OMA-DM CHECK FW Available Values <ul style="list-style-type: none"> 0x00000001 - FW Available. For CIDC and CIPRL, this value will be returned by the modem. CIDC and CIPRL are asynchronous OMADM sessions. 0x00000002 - FW Not Available 0x00000003 - FW Check Timed Out Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.938.2 Field Documentation

8.938.2.1 uint32_t unpack_swioama_SLQSOMADMStartSession_t::FwAvailability

8.938.2.2 swi_uint256_t unpack_swioama_SLQSOMADMStartSession_t::ParamPresenceMask

8.939 unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.939.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result• Bit to check in ParamPresenceMask - 2
<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.

8.939.2 Field Documentation

8.939.2.1 swi_uint256_t unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t::ParamPresenceMask

8.939.2.2 uint16_t unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t::Tlvresult

8.940 unpack_tmd_SLQSTmdGetMitigationDevList_t Struct Reference

Data Fields

- uint8_t [MitigationDevListLen](#)
- [tmd_mitigationDevList](#) [MitigationDevList](#) [255]
- uint16_t [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.940.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

Parameters

<i>MitigationDevListLen</i>	<ul style="list-style-type: none">• Mitigation Device List Length (Optional)• Number of sets of the following elements<ul style="list-style-type: none">– MitigationDevList• Bit to check in ParamPresenceMask - 16
-----------------------------	---

<i>pMitigationDevList</i>	<ul style="list-style-type: none"> • Mitigation Device List (Optional) • See tmd_mitigationDevList for more information. • Bit to check in ParamPresenceMask - 16
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.940.2 Field Documentation

8.940.2.1 `tmd_mitigationDevList` `unpack_tmd_SLQSTmdGetMitigationDevList_t::MitigationDevList[255]`

8.940.2.2 `uint8_t` `unpack_tmd_SLQSTmdGetMitigationDevList_t::MitigationDevListLen`

8.940.2.3 `swi_uint256_t` `unpack_tmd_SLQSTmdGetMitigationDevList_t::ParamPresenceMask`

8.940.2.4 `uint16_t` `unpack_tmd_SLQSTmdGetMitigationDevList_t::Tlvresult`

8.941 `unpack_tmd_SLQSTmdGetMitigationLvl_t` Struct Reference

Data Fields

- `uint8_t` [CurrentmitigationLvl](#)
- `uint8_t` [ReqMitigationLvl](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.941.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

<i>CurrentmitigationLvl</i>	<ul style="list-style-type: none"> • Current thermal mitigation level (Optional) <ul style="list-style-type: none"> – Bit to check in ParamPresenceMask - 16
<i>ReqMitigationLvl</i>	<ul style="list-style-type: none"> • Requested Thermal Mitigation Level (Optional) • 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"> – Bit to check in ParamPresenceMask - 17
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.941.2 Field Documentation

8.941.2.1 uint8_t unpack_tmd_SLQSTmdGetMitigationLvl_t::CurrentmitigationLvl

8.941.2.2 swi_uint256_t unpack_tmd_SLQSTmdGetMitigationLvl_t::ParamPresenceMask

8.941.2.3 uint8_t unpack_tmd_SLQSTmdGetMitigationLvl_t::ReqMitigationLvl

8.941.2.4 uint16_t unpack_tmd_SLQSTmdGetMitigationLvl_t::Tlvresult

8.942 unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t Struct Reference

Data Fields

- uint8_t [deviceIDLen](#)
- char [deviceID](#) [255]
- uint8_t [lvl](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.942.1 Detailed Description

Mitigation Level Report

Parameters

<i>deviceIDLen</i>	<ul style="list-style-type: none"> • Mitigation Device ID Length • Bit to check in ParamPresenceMask - 1
<i>deviceID</i>	<ul style="list-style-type: none"> • Mitigation Device ID • Bit to check in ParamPresenceMask - 1
<i>lvl</i>	<ul style="list-style-type: none"> • Current thermal mitigation level • Bit to check in ParamPresenceMask - 2
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.942.2 Field Documentation

8.942.2.1 char unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::deviceID[255]

8.942.2.2 uint8_t unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::deviceIDLen

8.942.2.3 uint8_t unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::lvl

8.942.2.4 swi_uint256_t unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::ParamPresenceMask

8.943 unpack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference

Data Fields

- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.943.1 Detailed Description

This structure contains mitigation devices Level request parameters

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result • Bit to check in ParamPresenceMask - 2
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.943.2 Field Documentation

8.943.2.1 [swi_uint256_t unpack_tmd_SLQSTmdRegNotMitigationLvl_t::ParamPresenceMask](#)

8.943.2.2 [uint16_t unpack_tmd_SLQSTmdRegNotMitigationLvl_t::Tlvresult](#)

8.944 unpack_uim_ChangePin_t Struct Reference

Data Fields

- [uim_remainingRetries](#) * [pRemainingRetries](#)
- [uim_encryptedPIN1](#) * [pEncryptedPIN1](#)
- [uint32_t](#) * [pIndicationToken](#)
- [uint16_t Tlvresult](#)
- [swi_uint256_t ParamPresenceMask](#)

8.944.1 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"> • See uim_remainingRetries for more information. • Bit to check in ParamPresenceMask - 16
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information. • Bit to check in ParamPresenceMask - 17
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • Bit to check in ParamPresenceMask - 18

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.944.2 Field Documentation

8.944.2.1 `swi_uint256_t unpack_uim_ChangePin_t::ParamPresenceMask`

8.944.2.2 `uim_encryptedPIN1* unpack_uim_ChangePin_t::pEncryptedPIN1`

8.944.2.3 `uint32_t* unpack_uim_ChangePin_t::pIndicationToken`

8.944.2.4 `uim_remainingRetries* unpack_uim_ChangePin_t::pRemainingRetries`

8.944.2.5 `uint16_t unpack_uim_ChangePin_t::Tlvresult`

8.945 unpack_uim_GetCardStatus_t Struct Reference

Data Fields

- [uim_cardStatus](#) * [pCardStatus](#)
- [uim_hotSwapStatus](#) * [pHotSwapStatus](#)
- `uint16_t` [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.945.1 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"> See uim_cardStatus for more information. Bit to check in ParamPresenceMask - 16
<i>pHotSwap-Status(optional)</i>	<ul style="list-style-type: none"> See uim_hotSwapStatus for more information. Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.945.2 Field Documentation

8.945.2.1 `swi_uint256_t unpack_uim_GetCardStatus_t::ParamPresenceMask`

8.945.2.2 `uim_cardStatus* unpack_uim_GetCardStatus_t::pCardStatus`

8.945.2.3 `uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus`

8.945.2.4 uint16_t unpack_uim_GetCardStatus_t::Tlvresult

8.946 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](#)

8.946.1 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"> • See uim_cardStatus for more information. • Bit to check in ParamPresenceMask - 16
<i>pHotSwap-Status(optional)</i>	<ul style="list-style-type: none"> • See uim_hotSwapStatus for more information. • Bit to check in ParamPresenceMask - 17
<i>pValidCard-Status(optional)</i>	<ul style="list-style-type: none"> • See uim_validCardStatus for more information. • Bit to check in ParamPresenceMask - 18
<i>pSimBusy-Status(optional)</i>	<ul style="list-style-type: none"> • See uim_simBusyStatus for more information. • Bit to check in ParamPresenceMask - 21
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.946.2 Field Documentation

8.946.2.1 swi_uint256_t unpack_uim_GetCardStatusV2_t::ParamPresenceMask

8.946.2.2 uim_cardStatus* unpack_uim_GetCardStatusV2_t::pCardStatus

8.946.2.3 uim_hotSwapStatus* unpack_uim_GetCardStatusV2_t::pHotSwapStatus

8.946.2.4 uim_simBusyStatus* unpack_uim_GetCardStatusV2_t::pSimBusyStatus

8.946.2.5 uim_validCardStatus* unpack_uim_GetCardStatusV2_t::pValidCardStatus

8.946.2.6 uint16_t unpack_uim_GetCardStatusV2_t::Tlvresult

8.947 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](#)

8.947.1 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"> • See uim_cardResult for more information. • Bit to check in ParamPresenceMask - 16
<i>pReadResult</i>	<ul style="list-style-type: none"> • See uim_readResult for more information. • Bit to check in ParamPresenceMask - 17
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result must be provided in a subsequent indication. • Bit to check in ParamPresenceMask - 18
<i>pEncrypted-Data(optional)</i>	<ul style="list-style-type: none"> • Encrypted Data. • Indicates whether the data from the card passed in read_result is encrypted. • Bit to check in ParamPresenceMask - 19
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.947.2 Field Documentation

8.947.2.1 [swi_uint256_t](#) [unpack_uim_ReadTransparent_t::ParamPresenceMask](#)

8.947.2.2 [uim_cardResult*](#) [unpack_uim_ReadTransparent_t::pCardResult](#)

8.947.2.3 [uint8_t*](#) [unpack_uim_ReadTransparent_t::pEncryptedData](#)

8.947.2.4 [uint32_t*](#) [unpack_uim_ReadTransparent_t::pIndicationToken](#)

8.947.2.5 [uim_readResult*](#) [unpack_uim_ReadTransparent_t::pReadResult](#)

8.947.2.6 [uint16_t](#) [unpack_uim_ReadTransparent_t::Tlvresult](#)

8.948 unpack_uim_SetPinProtection_t Struct Reference

Data Fields

- [uim_remainingRetries](#) * [pRemainingRetries](#)
- [uim_encryptedPIN1](#) * [pEncryptedPIN1](#)
- [uint32_t](#) * [pIndicationToken](#)
- [uint16_t](#) [Tlvresult](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.948.1 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"> • See uim_remainingRetries for more information. • Bit to check in ParamPresenceMask - 16
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information. • Bit to check in ParamPresenceMask - 17
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.948.2 Field Documentation

8.948.2.1 [swi_uint256_t](#) [unpack_uim_SetPinProtection_t::ParamPresenceMask](#)

8.948.2.2 [uim_encryptedPIN1*](#) [unpack_uim_SetPinProtection_t::pEncryptedPIN1](#)

8.948.2.3 [uint32_t*](#) [unpack_uim_SetPinProtection_t::pIndicationToken](#)

8.948.2.4 [uim_remainingRetries*](#) [unpack_uim_SetPinProtection_t::pRemainingRetries](#)

8.948.2.5 [uint16_t](#) [unpack_uim_SetPinProtection_t::Tlvresult](#)

8.949 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference

Data Fields

- [slots_t](#) [slotsstatusChange](#)
- [uint8_t](#) [bNumberOfPhySlots](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.949.1 Detailed Description

Structure consist of card status params

Parameters

<i>slotsstatus-Change</i>	<ul style="list-style-type: none"> See slot_t for more information Bit to check in ParamPresenceMask - 16
<i>bNumberOfPhy-Slots</i>	<ul style="list-style-type: none"> Number of Physical Slot(s) Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.949.2 Field Documentation

8.949.2.1 `uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots`

8.949.2.2 `swi_uint256_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::ParamPresenceMask`

8.949.2.3 `slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange`

8.950 unpack_uim_SLQSUIMAuthenticate_t Struct Reference

Data Fields

- `uim_cardResult * pCardResult`
- `uim_authenticateResult * pAuthenticateResult`
- `uint32_t * pIndicationToken`
- `swi_uint256_t ParamPresenceMask`

8.950.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMAuthenticate.

Parameters

<i>pCard-Result(optional)</i>	<ul style="list-style-type: none"> See uim_cardResult for more information. Bit to check in ParamPresenceMask - 16
<i>pAuthenticate-Result(optional)</i>	<ul style="list-style-type: none"> See uim_authenticateResult for more information. Bit to check in ParamPresenceMask - 17
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> Response in Indication. When this TLV is present, it indicates that the result must be provided in a subsequent indication. Bit to check in ParamPresenceMask - 18

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.950.2 Field Documentation

8.950.2.1 `swi_uint256_t unpack_uim_SLQSUIMAuthenticate_t::ParamPresenceMask`

8.950.2.2 `uim_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult`

8.950.2.3 `uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult`

8.950.2.4 `uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken`

8.951 `unpack_uim_SLQSUIMDepersonalization_t` Struct Reference

Data Fields

- [uim_remainingRetries](#) * [pRemainingRetries](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.951.1 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"> See uim_remainingRetries for more information. Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.951.2 Field Documentation

8.951.2.1 `swi_uint256_t unpack_uim_SLQSUIMDepersonalization_t::ParamPresenceMask`

8.951.2.2 `uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries`

8.952 `unpack_uim_SLQSUIEventRegister_t` Struct Reference

Data Fields

- `uint32_t` [eventMask](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.952.1 Detailed Description

This structure contains unpack event register parameter.

Parameters

<i>eventMask</i>	<ul style="list-style-type: none"> - bit 0 - card status • bit 1 - SAP connection • bit 4 - physical slot status
------------------	---

- Bit to check in ParamPresenceMask - 16

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.
--------------------------	--

8.952.2 Field Documentation

8.952.2.1 uint32_t unpack_uim_SLQSUIEventRegister_t::eventMask

8.952.2.2 swi_uint256_t unpack_uim_SLQSUIEventRegister_t::ParamPresenceMask

8.953 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference

Data Fields

- uint8_t * [pAutoSelection](#)
- uim_personalizationStatus * [pPersonalizationStatus](#)
- uint8_t * [pHaltSubscription](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.953.1 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"> • Indicates whether the modem is configured to automatically select the provisioning sessions at powerup. • Valid values <ul style="list-style-type: none"> – 0 - Automatic provisioning is off – 1 - Automatic provisioning is on • Bit to check in ParamPresenceMask - 16
<i>p-Personalization-Status(optional)</i>	<ul style="list-style-type: none"> • See uim_personalizationStatus for more information. • Bit to check in ParamPresenceMask - 17

<i>pHalt-Subscription(optional)</i>	<ul style="list-style-type: none"> Indicates if the modem is configured to publish the subscription after successful initialization. Valid values <ul style="list-style-type: none"> 0 - Modem proceeds with publishing the subscription 1 - Modem does not publish the subscription Bit to check in ParamPresenceMask - 18
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.953.2 Field Documentation

8.953.2.1 `swi_uint256_t unpack_uim_SLQSUIMGetConfiguration_t::ParamPresenceMask`

8.953.2.2 `uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection`

8.953.2.3 `uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription`

8.953.2.4 `uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus`

8.954 `unpack_uim_SLQSUIMGetFileAttributes_t` Struct Reference

Data Fields

- [uim_cardResult](#) * [pCardResult](#)
- [uim_fileAttributes](#) * [pFileAttributes](#)
- [uint32_t](#) * [pIndicationToken](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.954.1 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"> See uim_cardResult for more information. Bit to check in ParamPresenceMask - 16
<i>pFile-Attributes(optional)</i>	<ul style="list-style-type: none"> See uim_fileAttributes for more information. Bit to check in ParamPresenceMask - 17
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> Response in Indication. When this TLV is present, it indicates that the result must be provided in a subsequent indication. Bit to check in ParamPresenceMask - 18
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.954.2 Field Documentation

8.954.2.1 `swi_uint256_t` `unpack_uim_SLQSUIMGetFileAttributes_t::ParamPresenceMask`

8.954.2.2 `uim_cardResult*` `unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult`

8.954.2.3 `uim_fileAttributes*` `unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes`

8.954.2.4 `uint32_t*` `unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken`

8.955 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

Data Fields

- `uint8_t *` `pNumberOfPhySlot`
- `slots_t *` `pUimSlotsStatus`
- `swi_uint256_t` `ParamPresenceMask`

8.955.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

Parameters

<i>pNumberOfPhy-Slot</i>	<ul style="list-style-type: none"> • Number of sets of the Slot Status. • Bit to check in ParamPresenceMask - 16
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> • Slots Status See slots_t for more information.. • Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.955.2 Field Documentation

8.955.2.1 `swi_uint256_t` `unpack_uim_SLQSUIMGetSlotsStatus_t::ParamPresenceMask`

8.955.2.2 `uint8_t*` `unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot`

8.955.2.3 `slots_t*` `unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus`

8.956 unpack_uim_SLQSUIMGetSlotsStatusV2_t Struct Reference

Data Fields

- `uim_GetSlotsStatusTlv *` `pGetSlotsStatusTlv`
- `uim_GetSlotsInfoTlv *` `pGetSlotsInfoTlv`
- `swi_uint256_t` `ParamPresenceMask`

8.956.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

Parameters

<i>pNumberOfPhy-Slot</i>	<ul style="list-style-type: none"> Number of sets of the Slot Status. Bit to check in ParamPresenceMask - 16
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> Slots Status See slots_t for more information.. Bit to check in ParamPresenceMask - 16
<i>pNumberOfPhy-SlotInfo</i>	<ul style="list-style-type: none"> Number of Physical slot information. Bit to check in ParamPresenceMask - 17
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> Slots Status See uim_physlotsInfo for more information.. Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.956.2 Field Documentation

8.956.2.1 `swi_uint256_t unpack_uim_SLQSUIMGetSlotsStatusV2_t::ParamPresenceMask`

8.956.2.2 `uim_GetSlotsInfoTlv* unpack_uim_SLQSUIMGetSlotsStatusV2_t::pGetSlotsInfoTlv`

8.956.2.3 `uim_GetSlotsStatusTlv* unpack_uim_SLQSUIMGetSlotsStatusV2_t::pGetSlotsStatusTlv`

8.957 unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uim_refreshevent refreshEvent`
- `swi_uint256_t ParamPresenceMask`

8.957.1 Detailed Description

This structure hold parameters about UIM refresh event indication.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> value is 1 if refresh event TLV is present in indication, otherwise 0
<i>refreshEvent[OPTIONAL]</i>	<ul style="list-style-type: none"> see uim_refreshevent Bit to check in ParamPresenceMask - 16

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.957.2 Field Documentation

8.957.2.1 swi_uint256_t unpack_uim_SLQSUIMRefreshCallback_Ind_t::ParamPresenceMask

8.957.2.2 uim_refreshevent unpack_uim_SLQSUIMRefreshCallback_Ind_t::refreshEvent

8.957.2.3 uint8_t unpack_uim_SLQSUIMRefreshCallback_Ind_t::TlvPresent

8.958 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

Data Fields

- [uim_refreshevent](#) * [pRefreshEvent](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.958.1 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"> See uim_refreshevent for more information. Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.958.2 Field Documentation

8.958.2.1 swi_uint256_t unpack_uim_SLQSUIMRefreshGetLastEvent_t::ParamPresenceMask

8.958.2.2 uim_refreshevent* unpack_uim_SLQSUIMRefreshGetLastEvent_t::pRefreshEvent

8.959 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

Data Fields

- [uim_cardStatus](#) * [pCardStatus](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.959.1 Detailed Description

This structure contains information about Status change callback.

Parameters

<i>pCardStatus</i>	Card Status <ul style="list-style-type: none"> • See uim_cardStatus for more information. • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.959.2 Field Documentation

8.959.2.1 `swi_uint256_t unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::ParamPresenceMask`8.959.2.2 `uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus`8.960 `unpack_uim_UnblockPin_t` Struct Reference

Data Fields

- [uim_remainingRetries](#) * [pRemainingRetries](#)
- [uim_encryptedPIN1](#) * [pEncryptedPIN1](#)
- `uint32_t` * [pIndicationToken](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.960.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"> • See uim_remainingRetries for more information. • Bit to check in ParamPresenceMask - 16
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information. • Bit to check in ParamPresenceMask - 17
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.960.2 Field Documentation

8.960.2.1 `swi_uint256_t unpack_uim_UnblockPin_t::ParamPresenceMask`

8.960.2.2 uim_encryptedPIN1* unpack_uim_UnblockPin_t::pEncryptedPIN1

8.960.2.3 uint32_t* unpack_uim_UnblockPin_t::pIndicationToken

8.960.2.4 uim_remainingRetries* unpack_uim_UnblockPin_t::pRemainingRetries

8.960.2.5 uint16_t unpack_uim_UnblockPin_t::Tlvresult

8.961 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](#)

8.961.1 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"> • See uim_remainingRetries for more information. • Bit to check in ParamPresenceMask - 16
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information. • Bit to check in ParamPresenceMask - 17
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • Bit to check in ParamPresenceMask - 18
<i>pCardResult(optional)</i>	<ul style="list-style-type: none"> • See uim_cardResult for more information. • Bit to check in ParamPresenceMask - 19
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.961.2 Field Documentation

8.961.2.1 swi_uint256_t unpack_uim_UnblockPinV2_t::ParamPresenceMask

8.961.2.2 uim_cardResult* unpack_uim_UnblockPinV2_t::pCardResult

8.961.2.3 `uim_encryptedPIN1*` `unpack_uim_UnblockPinV2_t::pEncryptedPIN1`

8.961.2.4 `uint32_t*` `unpack_uim_UnblockPinV2_t::pIndicationToken`

8.961.2.5 `uim_remainingRetries*` `unpack_uim_UnblockPinV2_t::pRemainingRetries`

8.961.2.6 `uint16_t` `unpack_uim_UnblockPinV2_t::Tlvresult`

8.962 `unpack_uim_VerifyPin_t` Struct Reference

Data Fields

- `uim_remainingRetries` * `pRemainingRetries`
- `uim_encryptedPIN1` * `pEncryptedPIN1`
- `uint32_t` * `pIndicationToken`
- `uint16_t` `Tlvresult`
- `swi_uint256_t` `ParamPresenceMask`

8.962.1 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"> • See uim_remainingRetries for more information. • Bit to check in ParamPresenceMask - 16
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information. • Bit to check in ParamPresenceMask - 17
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> • Response in Indication. • When this TLV is present, it indicates that the result is provided in a subsequent indication. • Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.962.2 Field Documentation

8.962.2.1 `swi_uint256_t` `unpack_uim_VerifyPin_t::ParamPresenceMask`

8.962.2.2 `uim_encryptedPIN1*` `unpack_uim_VerifyPin_t::pEncryptedPIN1`

8.962.2.3 `uint32_t*` `unpack_uim_VerifyPin_t::pIndicationToken`

8.962.2.4 `uim_remainingRetries*` `unpack_uim_VerifyPin_t::pRemainingRetries`

8.962.2.5 `uint16_t` `unpack_uim_VerifyPin_t::Tlvresult`

8.963 unpack_voice_allCallStatusCallback_ind_t Struct Reference

Data Fields

- [voice_arrCallInfo](#) [arrCallInfo](#) information
- [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](#)

8.963.1 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"> • See voice_arrCallInfo for more information. • Bit to check in ParamPresenceMask - 1
<i>pArrRemote-Party-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrRemotePartyNum for more information. • Bit to check in ParamPresenceMask - 16
<i>pArrRemote-Party-Name(optional)</i>	<ul style="list-style-type: none"> • See voice_arrRemotePartyName for more information. • Bit to check in ParamPresenceMask - 17
<i>pArrAlerting-Type(optional)</i>	<ul style="list-style-type: none"> • See voice_arrAlertingType for more information. • Bit to check in ParamPresenceMask - 18
<i>pArrSvc-Option(optional)</i>	<ul style="list-style-type: none"> • See voice_arrSvcOption for more information. • Bit to check in ParamPresenceMask - 19
<i>pArrCallEnd-Reason(optional)</i>	<ul style="list-style-type: none"> • See voice_arrCallEndReason for more information. • Bit to check in ParamPresenceMask - 20
<i>pArrAlpha-ID(optional)</i>	<ul style="list-style-type: none"> • See voice_arrAlphaID for more information. • Bit to check in ParamPresenceMask - 21

<i>pArrConnect-Party-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrConnectPartyNum for more information. • Bit to check in ParamPresenceMask - 22
<i>pArrDiag-Info(optional)</i>	<ul style="list-style-type: none"> • See voice_arrDiagInfo for more information. • Bit to check in ParamPresenceMask - 23
<i>pArrCalledParty-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrCalledPartyNum for more information. • Bit to check in ParamPresenceMask - 24
<i>pArrRedirParty-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrRedirPartyNum for more information. • Bit to check in ParamPresenceMask - 25
<i>pArrAlerting-Pattern(optional)</i>	<ul style="list-style-type: none"> • See voice_arrAlertingPattern for more information. • Bit to check in ParamPresenceMask - 26
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.963.2 Field Documentation

8.963.2.1 `voice_arrCallInfo` `unpack_voice_allCallStatusCallback_ind_t::arrCallInfomation`

8.963.2.2 `swi_uint256_t` `unpack_voice_allCallStatusCallback_ind_t::ParamPresenceMask`

8.963.2.3 `voice_arrAlertingPattern*` `unpack_voice_allCallStatusCallback_ind_t::pArrAlertingPattern`

8.963.2.4 `voice_arrAlertingType*` `unpack_voice_allCallStatusCallback_ind_t::pArrAlertingType`

8.963.2.5 `voice_arrAlphaID*` `unpack_voice_allCallStatusCallback_ind_t::pArrAlphaID`

8.963.2.6 `voice_arrCalledPartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrCalledPartyNum`

8.963.2.7 `voice_arrCallEndReason*` `unpack_voice_allCallStatusCallback_ind_t::pArrCallEndReason`

8.963.2.8 `voice_arrConnectPartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrConnectPartyNum`

8.963.2.9 `voice_arrDiagInfo*` `unpack_voice_allCallStatusCallback_ind_t::pArrDiagInfo`

8.963.2.10 `voice_arrRedirPartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrRedirPartyNum`

8.963.2.11 `voice_arrRemotePartyName*` `unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyName`

8.963.2.12 `voice_arrRemotePartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyNum`

8.963.2.13 `voice_arrSvcOption*` `unpack_voice_allCallStatusCallback_ind_t::pArrSvcOption`

8.964 unpack_voice_DTMFEventCallback_ind_t Struct Reference

Data Fields

- [voice_DTMFInfo](#) DTMFInformation
- `uint8_t * pOnLength`
- `uint8_t * pOffLength`
- [swi_uint256_t](#) ParamPresenceMask

8.964.1 Detailed Description

This structure contains the parameters passed for DTMF event indication by the device.

Parameters

<i>DTMF-Information(mandatory)</i>	See voice_DTMFInfo for more information. <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 1
<i>pOn-Length(optional)</i>	<ul style="list-style-type: none"> • DTMF Pulse Width <ul style="list-style-type: none"> – 0x00 - DTMF_ONLENGTH_95MS - 95 ms – 0x01 - DTMF_ONLENGTH_150MS - 150 ms – 0x02 - DTMF_ONLENGTH_200MS - 200 ms – 0x03 - DTMF_ONLENGTH_250MS - 250 ms – 0x04 - DTMF_ONLENGTH_300MS - 300 ms – 0x05 - DTMF_ONLENGTH_350MS - 350 ms – 0x06 - DTMF_ONLENGTH_SMS - SMS Tx special pulse width • Bit to check in ParamPresenceMask - 16
<i>pOff-Length(optional)</i>	<ul style="list-style-type: none"> • DTMF Interdigit Interval <ul style="list-style-type: none"> – 0x00 - DTMF_OFFLENGTH_60MS - 60 ms – 0x01 - DTMF_OFFLENGTH_100MS - 100 ms – 0x02 - DTMF_OFFLENGTH_150MS - 150 ms – 0x03 - DTMF_OFFLENGTH_200MS - 200 ms • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

None

8.964.2 Field Documentation

8.964.2.1 `voice_DTMFInfo` `unpack_voice_DTMFEventCallback_ind_t::DTMFInformation`

8.964.2.2 `swi_uint256_t` `unpack_voice_DTMFEventCallback_ind_t::ParamPresenceMask`

8.964.2.3 `uint8_t*` `unpack_voice_DTMFEventCallback_ind_t::pOffLength`

8.964.2.4 uint8_t* unpack_voice_DTMFEventCallback_ind_t::pOnLength

8.965 unpack_voice_OTASPStatusCallback_ind_t Struct Reference

Data Fields

- uint8_t [callID](#)
- uint8_t [OTASPStatus](#)
- swi_uint256_t [ParamPresenceMask](#)

8.965.1 Detailed Description

This structure consist of OTASP or OTAPA event params

Parameters

<i>callID</i>	<ul style="list-style-type: none"> • Call identifier for the call. • Bit to check in ParamPresenceMask - 1
<i>OTASPStatus</i>	<ul style="list-style-type: none"> • OTASP status for the OTASP call. Values: <ul style="list-style-type: none"> – 0x00 - OTASP_STATUS_SPL_UNLOCKED.SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED. SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED.A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED. SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP(OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP(OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP(OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP(OTAPA) • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.965.2 Field Documentation

8.965.2.1 uint8_t unpack_voice_OTASPStatusCallback_ind_t::callID

8.965.2.2 uint8_t unpack_voice_OTASPStatusCallback_ind_t::OTASPStatus

8.965.2.3 swi_uint256_t unpack_voice_OTASPStatusCallback_ind_t::ParamPresenceMask

8.966 unpack_voice_SLQSPoriginateUSSD_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](#)

8.966.1 Detailed Description

This structure contains the parameters of USS response.

Parameters

<i>pfailureCause</i>	<ul style="list-style-type: none"> • Failure Cause. • 0 - QMI_FAILURE_CAUSE_OFFLINE - Phone is offline • 20 - QMI_FAILURE_CAUSE_CDMA_LOCK - Phone is CDMA locked until a power cycle; CDMA only • 21 - QMI_FAILURE_CAUSE_NO_SRV - Phone has no service • 22 - QMI_FAILURE_CAUSE_FADE - Call has ended abnormally • 23 - QMI_FAILURE_CAUSE_INTERCEPT - Received intercept from the base station; originating only; CDMA only • 24 - QMI_FAILURE_CAUSE_REORDER - Received reorder from the base station; originating only; CDMA only • 25 - QMI_FAILURE_CAUSE_REL_NORMAL - Received release from the base station; no reason was given • 26 - QMI_FAILURE_CAUSE_REL_SO_REJ - Received release from the base station; SO reject; CDMA only • 27 - QMI_FAILURE_CAUSE_INCOM_CALL - Received incoming call from the base station • 28 - QMI_FAILURE_CAUSE_ALERT_STOP - Received alert stop from the base station; incoming only; CDMA only • 29 - QMI_FAILURE_CAUSE_CLIENT_END - Client ended the call • 30 - QMI_FAILURE_CAUSE_ACTIVATION - Received end activation; OTASP call only; CDMA only • 31 - QMI_FAILURE_CAUSE_MC_ABORT - MC aborted the origination/conversation; CDMA only • 32 - QMI_FAILURE_CAUSE_MAX_ACCESS_PROBE - Maximum access probes were transmitted; CDMA only • 33 - QMI_FAILURE_CAUSE_PSIST_N - Persistence test failure; FEATURE_JCDMA only; CDMA only • 34 - QMI_FAILURE_CAUSE_UIM_NOT_PRESENT - R-UIM is not present • 35 - QMI_FAILURE_CAUSE_ACC_IN_PROG - Access attempt is already in progress • 36 - QMI_FAILURE_CAUSE_ACC_FAIL - Access failure for a reason other than the above • 37 - QMI_FAILURE_CAUSE_RETRY_ORDER - Received retry order; originating only; IS 2000; CDMA only • 38 - QMI_FAILURE_CAUSE_CCS_NOT_SUPPORTED_BYBS - Concurrent service is not supported by the base station • 39 - QMI_FAILURE_CAUSE_NO_RESPONSE_FROM_BS - No response was received from the base station • 40 - QMI_FAILURE_CAUSE_REJECTED_BY_BS - Call was rejected by the base station; CDMA only • 41 - QMI_FAILURE_CAUSE_INCOMPATIBLE - Concurrent services requested were not compatible; CDMA only • 42 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station; CDMA only • 43 - QMI_FAILURE_CAUSE_ALREADY_IN_TC - Corresponds to CM_CALL_ORIGERR_ALREADY_IN_TC • 44 - QMI_FAILURE_CAUSE_EMERGENCY_FLASHED - Call is ended because an emergency call was flashed over this call; CDMA only • 45 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_GPS - Used if CM is ending a GPS call in preference of a user call • 46 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_SMS - Used if CM is ending an SMS call in preference of a user call • 47 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_DATA - Used if CM is ending a data call in preference of an emergency call • 48 - QMI_FAILURE_CAUSE_REDIR_OR_HANDOFF - Call was rejected because of a redirection or handoff
	<ul style="list-style-type: none"> • 49 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only • 50 - QMI_FAILURE_CAUSE_OTASP_SPC_ERR - To support OTASP SPC Error indication • 51 - QMI_FAILURE_CAUSE_IS707B_MAX_ACC - Maximum access probes for an IS-707B

<i>AlphaIdentifier</i>	<ul style="list-style-type: none"> • see voice_alphaIDInfo definition • Bit to check in ParamPresenceMask - 17
<i>pUSSDInfo</i>	<ul style="list-style-type: none"> • USS Data from Network (See structure voice_USSInfo) • Bit to check in ParamPresenceMask - 18
<i>pCcResultType</i>	<ul style="list-style-type: none"> • CC result code • Bit to check in ParamPresenceMask - 19
<i>pCallId</i>	<ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 20 • NULL pointer - Invalid data.
<i>pCCSuppsType</i>	<ul style="list-style-type: none"> • See structure 'voice_ccSUPSType' definition • Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.966.2 Field Documentation

8.966.2.1 [voice_alphaIDInfo*](#) [unpack_voice_SLQSOriinateUSSD_t::pAlphaIDInfo](#)

8.966.2.2 [swi_uint256_t](#) [unpack_voice_SLQSOriinateUSSD_t::ParamPresenceMask](#)

8.966.2.3 [uint8_t*](#) [unpack_voice_SLQSOriinateUSSD_t::pCallId](#)

8.966.2.4 [uint8_t*](#) [unpack_voice_SLQSOriinateUSSD_t::pCcResultType](#)

8.966.2.5 [voice_ccSUPSType*](#) [unpack_voice_SLQSOriinateUSSD_t::pCCSuppsType](#)

8.966.2.6 [uint16_t*](#) [unpack_voice_SLQSOriinateUSSD_t::pfailureCause](#)

8.966.2.7 [struct voice_USSInfo*](#) [unpack_voice_SLQSOriinateUSSD_t::pUSSDInfo](#)

8.967 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference

Data Fields

- [uint8_t *](#) [pCallId](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.967.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> Unique call identifier for the call that must be answered. Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.967.2 Field Documentation

8.967.2.1 `swi_uint256_t unpack_voice_SLQSVoiceAnswerCall_t::ParamPresenceMask`8.967.2.2 `uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId`8.968 `unpack_voice_SLQSVoiceBurstDTMF_t` Struct Reference

Data Fields

- `uint8_t * pCallID`
- `swi_uint256_t ParamPresenceMask`

8.968.1 Detailed Description

This structure contains Voice Burst DTMF Information for unpack.

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> 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. 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. If the call ID value received is 0, no value has been returned by the device Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.968.2 Field Documentation

8.968.2.1 `swi_uint256_t unpack_voice_SLQSVoiceBurstDTMF_t::ParamPresenceMask`8.968.2.2 `uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID`8.969 `unpack_voice_SLQSVoiceDialCall_t` Struct Reference

Data Fields

- `uint8_t * pCallID`
- `voice_alphaIDInfo * pAlphaIDInfo`

- uint8_t * [pCCResultType](#)
- [voice_ccSUPSType](#) * [pCCSUPSType](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.969.1 Detailed Description

This structure contains Voice Call Response Parameters

Parameters

<i>pCallID(optional)</i>	<ul style="list-style-type: none"> • Unique call identifier for the dialed call • Bit to check in ParamPresenceMask - 16
<i>pAlphaID-Info(optional)</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 17
<i>pCCResult-Type(optional)</i>	<ul style="list-style-type: none"> • Call Control Result Type. <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service • Bit to check in ParamPresenceMask - 18
<i>pCCSUPS-Type(optional)</i>	<ul style="list-style-type: none"> • Pointer to structure of ccSUPSType • Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 19
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.969.2 Field Documentation

8.969.2.1 [voice_alphaIDInfo](#)* [unpack_voice_SLQSVoiceDialCall_t::pAlphaIDInfo](#)

8.969.2.2 [swi_uint256_t](#) [unpack_voice_SLQSVoiceDialCall_t::ParamPresenceMask](#)

8.969.2.3 [uint8_t](#)* [unpack_voice_SLQSVoiceDialCall_t::pCallID](#)

8.969.2.4 [uint8_t](#)* [unpack_voice_SLQSVoiceDialCall_t::pCCResultType](#)

8.969.2.5 [voice_ccSUPSType](#)* [unpack_voice_SLQSVoiceDialCall_t::pCCSUPSType](#)

8.970 unpack_voice_SLQSVoiceEndCall_t Struct Reference

Data Fields

- [uint8_t](#) * [pCallId](#)

- [swi_uint256_t ParamPresenceMask](#)

8.970.1 Detailed Description

This structure contains unpack voice end call parameter.

Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> • Unique call identifier for the call that must be ended • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.970.2 Field Documentation

8.970.2.1 [swi_uint256_t unpack_voice_SLQSVoiceEndCall_t::ParamPresenceMask](#)

8.970.2.2 [uint8_t* unpack_voice_SLQSVoiceEndCall_t::pCallId](#)

8.971 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](#)

8.971.1 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">• See voice_arrCallInfo for more information.• Bit to check in ParamPresenceMask - 16
<i>pArrRemote-Party-Num(optional)</i>	<ul style="list-style-type: none">• See voice_arrRemotePartyNum for more information.• Bit to check in ParamPresenceMask - 17
<i>pArrRemote-Party-Name(optional)</i>	<ul style="list-style-type: none">• See voice_arrRemotePartyName for more information.• Bit to check in ParamPresenceMask - 18
<i>pArrAlerting-Type(optional)</i>	<ul style="list-style-type: none">• See voice_arrAlertingType for more information.• Bit to check in ParamPresenceMask - 19
<i>pArrUUS-Info(optional)</i>	<ul style="list-style-type: none">• See voice_arrUUSInfo for more information.• Bit to check in ParamPresenceMask - 20
<i>pArrSvc-Option(optional)</i>	<ul style="list-style-type: none">• See voice_arrSvcOption for more information.• Bit to check in ParamPresenceMask - 21

<i>pOTASP-Status(optional)</i>	<ul style="list-style-type: none"> • OTASP status for the OTASP call. • Applicable only for 3GPP2 devices. <ul style="list-style-type: none"> – 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA) – 0xFF - Not Available • Bit to check in ParamPresenceMask - 22
<i>pVoice-Privacy(optional)</i>	<ul style="list-style-type: none"> • Voice Privacy. • Values. <ul style="list-style-type: none"> – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available • Bit to check in ParamPresenceMask - 23
<i>pArrCallEnd-Reason(optional)</i>	<ul style="list-style-type: none"> • See voice_arrCallEndReason for more information. • Bit to check in ParamPresenceMask - 24
<i>pArrAlphaID(optional)</i>	<ul style="list-style-type: none"> • See voice_arrAlphaID for more information. • Bit to check in ParamPresenceMask - 25
<i>pArrConnect-Party-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrConnectPartyNum for more information. • Bit to check in ParamPresenceMask - 26

<i>pArrDiag-Info(optional)</i>	<ul style="list-style-type: none"> • See voice_arrDiagInfo for more information. • Bit to check in ParamPresenceMask - 27
<i>pArrCalledParty-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrCalledPartyNum for more information. • Bit to check in ParamPresenceMask - 28
<i>pArrRedirParty-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_arrRedirPartyNum for more information. • Bit to check in ParamPresenceMask - 29
<i>pArrAlerting-Pattern(optional)</i>	<ul style="list-style-type: none"> • See voice_arrAlertingPattern for more information. • Bit to check in ParamPresenceMask - 30
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.971.2 Field Documentation

8.971.2.1 `swi_uint256_t unpack_voice_SLQSVoiceGetAllCallInfo_t::ParamPresenceMask`

8.971.2.2 `voice_arrAlertingPattern* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern`

8.971.2.3 `voice_arrAlertingType* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType`

8.971.2.4 `voice_arrAlphaID* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID`

8.971.2.5 `voice_arrCalledPartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum`

8.971.2.6 `voice_arrCallEndReason* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason`

8.971.2.7 `voice_arrCallInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo`

8.971.2.8 `voice_arrConnectPartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum`

8.971.2.9 `voice_arrDiagInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo`

8.971.2.10 `voice_arrRedirPartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum`

8.971.2.11 `voice_arrRemotePartyName* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName`

8.971.2.12 `voice_arrRemotePartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum`

8.971.2.13 `voice_arrSvcOption* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption`

8.971.2.14 `voice_arrUUSInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo`

8.971.2.15 `uint8_t* unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus`

8.971.2.16 `uint8_t* unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy`

8.972 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](#)

8.972.1 Detailed Description

This structure contains Voice Get Call Barring Response Parameters

Parameters

<i>pSvcClass</i>	<ul style="list-style-type: none"> • Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available • Bit to check in ParamPresenceMask - 16
<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20

<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.972.2 Field Documentation

8.972.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCallBarring_t::pAlphaIDInfo`

8.972.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCallBarring_t::ParamPresenceMask`

8.972.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pCallID`

8.972.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType`

8.972.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType`

8.972.2.6 `uint16_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause`

8.972.2.7 `uint8_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass`

8.973 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`

8.973.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Unpack Parameters

Parameters

<i>pGetCallFWInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of <code>getCallFWInfo</code> (optional) <ul style="list-style-type: none"> – See voice_getCallFWInfo for more information • Bit to check in ParamPresenceMask - 16
-----------------------	--

<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF, if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>pGetCallFWExt-Info</i>	<ul style="list-style-type: none"> • Pointer to structure of getCallFWExtInfo (optional) <ul style="list-style-type: none"> – See voice_getCallFWExtInfo for more information • Bit to check in ParamPresenceMask - 22
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.973.2 Field Documentation

8.973.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo`

8.973.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::ParamPresenceMask`

8.973.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID`

8.973.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType`

8.973.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType`

8.973.2.6 uint16_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pFailCause

8.973.2.7 voice_getCallFWExtInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWExtInfo

8.973.2.8 voice_getCallFWInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWInfo

8.974 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](#)

8.974.1 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"> • See voice_callInfo for more information. • Bit to check in ParamPresenceMask - 16
<i>pRemoteParty-Num(optional)</i>	<ul style="list-style-type: none"> • See voice_remotePartyNum for more information. • Bit to check in ParamPresenceMask - 17
<i>pSrvOpt</i>	<ul style="list-style-type: none"> • Service option(optional) • Applicable only for 3GPP2 devices. • See Table8 qaGobiApiTableServiceOptions.h for standard service option number assignments. • Bit to check in ParamPresenceMask - 18
<i>pVoicePrivacy</i>	<ul style="list-style-type: none"> • Voice Privacy.(optional) • Applicable only for 3GPP2 devices. • Values. <ul style="list-style-type: none"> – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19

<i>pOTASPStatus</i>	<ul style="list-style-type: none"> • OTASP status for the OTASP call.(optional) • Applicable only for 3GPP2 devices. <ul style="list-style-type: none"> – 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP – 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP – 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP – 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA) – 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP – 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP – 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP – 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP – 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP – 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA) – 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA) – 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA) – 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA) – 0xFF - Not Available • Bit to check in ParamPresenceMask - 20
<i>pRemoteParty-Name(optional)</i>	<ul style="list-style-type: none"> • Applicable only for 3GPP devices. • See voice_remotePartyName for more information. • Bit to check in ParamPresenceMask - 21
<i>pUUS-Info(optional)</i>	<ul style="list-style-type: none"> • Applicable only for 3GPP devices. • See voice_UUSInfo for more information. • Bit to check in ParamPresenceMask - 22
<i>pAlert-Type(optional)</i>	<ul style="list-style-type: none"> • Alerting type. • Applicable only for 3GPP devices. <ul style="list-style-type: none"> – 0x00 - ALERTING_LOCAL - Local – 0x01 - ALERTING_REMOTE - Remote – 0xFF - Not Available • Bit to check in ParamPresenceMask - 23

<i>pAlphaID-Info(optional)</i>	<ul style="list-style-type: none"> • Applicable only for 3GPP devices. • See voice_alphaIDInfo for more information. • Bit to check in ParamPresenceMask - 24
<i>pConnectNum-Info(optional)</i>	<ul style="list-style-type: none"> • See voice_connectNumInfo for more information. • Bit to check in ParamPresenceMask - 25
<i>pDiag-Info(optional)</i>	<ul style="list-style-type: none"> • See voice_diagInfo for more information. • Bit to check in ParamPresenceMask - 26
<i>pAlertingPattern</i>	<ul style="list-style-type: none"> • Alerting pattern.(optional) <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1 – 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2 – 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3 – 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5 – 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6 – 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7 – 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8 – 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9 – 0xFF - Not Available • Bit to check in ParamPresenceMask - 27
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.974.2 Field Documentation

8.974.2.1 `uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern`

8.974.2.2 `uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType`

8.974.2.3 `voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo`

8.974.2.4 `swi_uint256_t unpack_voice_SLQSVoiceGetCallInfo_t::ParamPresenceMask`

8.974.2.5 `voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo`

8.974.2.6 `voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo`

8.974.2.7 `voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo`

8.974.2.8 `uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus`

8.974.2.9 `voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName`

8.974.2.10 `voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum`

8.974.2.11 uint16_t* unpack_voice_SLQSVoiceGetCallInfo_t::pSrvOpt

8.974.2.12 voice_UUSInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pUUSInfo

8.974.2.13 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pVoicePrivacy

8.975 unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

Data Fields

- uint8_t * [pSvcClass](#)
- uint16_t * [pFailCause](#)
- [voice_alphalDInfo](#) * [pAlphaDInfo](#)
- uint8_t * [pCCResType](#)
- uint8_t * [pCallID](#)
- [voice_ccSUPSType](#) * [pCCSUPSType](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.975.1 Detailed Description

This structure contains Voice Get Call Waiting Unpack Parameters

Parameters

<i>pSvcClass</i>	<ul style="list-style-type: none"> • Service class is a combination (sum) of information class constants (optional) • See qaGobiApiTableSupServiceInfoClasses.h for service classes. • Service Class is set to 0 if call waiting is not active for any of the information classes. • 0xFF,if Not Available • Bit to check in ParamPresenceMask - 16
<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaDInfo (optional) <ul style="list-style-type: none"> – See voice_alphalDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19

<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.975.2 Field Documentation

8.975.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCallWaiting_t::pAlphaIDInfo`

8.975.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCallWaiting_t::ParamPresenceMask`

8.975.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCallWaiting_t::pCallID`

8.975.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCallWaiting_t::pCCResType`

8.975.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCallWaiting_t::pCCSUPSType`

8.975.2.6 `uint16_t*` `unpack_voice_SLQSVoiceGetCallWaiting_t::pFailCause`

8.975.2.7 `uint8_t*` `unpack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass`

8.976 unpack_voice_SLQSVoiceGetCLIP_t Struct Reference

Data Fields

- [voice_CLIPResp](#) * `pCLIPResp`
- `uint16_t` * `pFailCause`
- [voice_alphaIDInfo](#) * `pAlphaIDInfo`
- `uint8_t` * `pCCResType`
- `uint8_t` * `pCallID`
- [voice_ccSUPSType](#) * `pCCSUPSType`
- `swi_uint256_t` `ParamPresenceMask`

8.976.1 Detailed Description

This structure contains Voice Get Calling Line Identification Presentation(CLIP) Response Parameters

Parameters

<i>pCLIPResp</i>	<ul style="list-style-type: none"> • Pointer to structure of CLIPResp (optional) <ul style="list-style-type: none"> – See voice_CLIPResp for more information • Bit to check in ParamPresenceMask - 16
<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF, if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.976.2 Field Documentation

8.976.2.1 `voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIP_t::pAlphaIDInfo`8.976.2.2 `swi_uint256_t unpack_voice_SLQSVoiceGetCLIP_t::ParamPresenceMask`8.976.2.3 `uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCallID`8.976.2.4 `uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCCResType`8.976.2.5 `voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType`

8.976.2.6 `voice_CLIPResp`* `unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp`

8.976.2.7 `uint16_t`* `unpack_voice_SLQSVoiceGetCLIP_t::pFailCause`

8.977 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`

8.977.1 Detailed Description

This structure contains Voice Get Calling Line Identification Restriction (CLIR) Response Parameters

Parameters

<i>pCLIRResp</i>	<ul style="list-style-type: none"> • Pointer to structure of CLIRResp (optional) <ul style="list-style-type: none"> – See voice_CLIRResp for more information • Bit to check in ParamPresenceMask - 16
<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaDInfo (optional) <ul style="list-style-type: none"> – See voice_alphalDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20

<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.977.2 Field Documentation

8.977.2.1 [voice_alphaIDInfo](#)* [unpack_voice_SLQSVoiceGetCLIR_t::pAlphaIDInfo](#)

8.977.2.2 [swi_uint256_t](#) [unpack_voice_SLQSVoiceGetCLIR_t::ParamPresenceMask](#)

8.977.2.3 [uint8_t](#)* [unpack_voice_SLQSVoiceGetCLIR_t::pCallID](#)

8.977.2.4 [uint8_t](#)* [unpack_voice_SLQSVoiceGetCLIR_t::pCCResType](#)

8.977.2.5 [voice_ccSUPSType](#)* [unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType](#)

8.977.2.6 [voice_CLIRResp](#)* [unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp](#)

8.977.2.7 [uint16_t](#)* [unpack_voice_SLQSVoiceGetCLIR_t::pFailCause](#)

8.978 [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](#)

8.978.1 Detailed Description

This structure contains Voice Get Calling Name Presentation(CNAP) Response Parameters

Parameters

<i>pCNAPResp</i>	<ul style="list-style-type: none"> • Pointer to structure of CNAPResp (optional) <ul style="list-style-type: none"> – See voice_CNAPResp for more information • Bit to check in ParamPresenceMask - 16
------------------	---

<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF, if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.978.2 Field Documentation

8.978.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCNAP_t::pAlphaIDInfo`

8.978.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCNAP_t::ParamPresenceMask`

8.978.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCNAP_t::pCallID`

8.978.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCNAP_t::pCCResType`

8.978.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType`

8.978.2.6 `voice_CNAPResp*` `unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp`

8.978.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCNAP_t::pFailCause`

8.979 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](#)

8.979.1 Detailed Description

This structure contains Voice Get Connected Line Identification Presentation(COLP) Response Parameters

Parameters

<i>pCOLPResp</i>	<ul style="list-style-type: none"> • Pointer to structure of COLPResp (optional) <ul style="list-style-type: none"> – See voice_COLPResp for more information • Bit to check in ParamPresenceMask - 16
<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • Bit to check in ParamPresenceMask - 17
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.
--------------------------	--

8.979.2 Field Documentation

8.979.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCOLP_t::pAlphaIDInfo`

8.979.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCOLP_t::ParamPresenceMask`

8.979.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLP_t::pCallID`

8.979.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLP_t::pCCResType`

8.979.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType`

8.979.2.6 `voice_COLPResp*` `unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp`

8.979.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCOLP_t::pFailCause`

8.980 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`

8.980.1 Detailed Description

This structure contains Voice Get Connected Line Identification Restriction(COLR) Response Parameters

Parameters

<i>pCOLRResp</i>	<ul style="list-style-type: none"> Pointer to structure of COLRResp (optional) <ul style="list-style-type: none"> See voice_COLRResp for more information Bit to check in ParamPresenceMask - 16
<i>pFailCause</i>	<ul style="list-style-type: none"> Supplementary services failure cause (optional) see qaGobiApiTableVoiceCallEndReasons.h for more information. 0xFFFF,if Not Available Bit to check in ParamPresenceMask - 17

<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 18
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 20
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 21
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.980.2 Field Documentation

8.980.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo`

8.980.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCOLR_t::ParamPresenceMask`

8.980.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLR_t::pCallID`

8.980.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLR_t::pCCResType`

8.980.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType`

8.980.2.6 `voice_COLRResp*` `unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp`

8.980.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCOLR_t::pFailCause`

8.981 `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](#)

8.981.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

Parameters

<i>pAutoAnswer-Stat(optional)</i>	<ul style="list-style-type: none"> • Auto Answer Status • Value returned is read from NV_AUTO_ANSWER_I. <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled – 0xFF - Not Available • Bit to check in ParamPresenceMask - 16
<i>pAirTimer-Cnt(optional)</i>	<ul style="list-style-type: none"> • Air Timer Count • Value returned is read from NV_AIR_CNT_I. • See voice_airTimer for more information • Bit to check in ParamPresenceMask - 17
<i>pRoamTimer-Cnt(optional)</i>	<ul style="list-style-type: none"> • Roam Timer Count • Value returned is read from NV_ROAM_CNT_I. • See voice_roamTimer for more information • Bit to check in ParamPresenceMask - 18
<i>pCurrTTY-Mode(optional)</i>	<ul style="list-style-type: none"> • Current TTY Mode • Value returned is read from NV_TTY_I. <ul style="list-style-type: none"> – 0x00 - TTY_MODE_FULL - Full – 0x01 - TTY_MODE_VCO - Voice carry over – 0x02 - TTY_MODE_HCO - Hearing carry over – 0x03 - TTY_MODE_OFF - Off – 0xFF - Not Available • Bit to check in ParamPresenceMask - 19
<i>pCurPrefVoiceSO(optional)</i>	<ul style="list-style-type: none"> • Current Preferred Voice SO • Value returned is read from NV_PREF_VOICE_SO_I. • See voice_prefVoiceSO for more information • Bit to check in ParamPresenceMask - 20

<i>pCurAMR-Config(optional)</i>	<ul style="list-style-type: none"> • Current Adaptive Multi-Rate Configuration. • Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_AMR_CODE_C_PREFERENCE_CONFIG_I. • See voice_curAMRConfig for more information • Bit to check in ParamPresenceMask - 21
<i>pCurVoice-Privacy-Pref(optional)</i>	<ul style="list-style-type: none"> • Current Voice Privacy Preference • Value returned is read from NV_VOICE_PRIV_I. <ul style="list-style-type: none"> – 0x00 - Standard privacy – 0x01 - Enhanced privacy – 0xFF - Not Available • Bit to check in ParamPresenceMask - 22
<i>pCurVoice-Domain-Pref(optional)</i>	<ul style="list-style-type: none"> • Current Voice Domain Preference. <ul style="list-style-type: none"> – 0x00 - Circuit-switched (CS) only – 0x01 - Packet-switched (PS) only – 0x02 - CS is preferred; PS is secondary – 0x03 - PS is preferred; CS is secondary – 0xFF - Not Available • Bit to check in ParamPresenceMask - 23
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.981.2 Field Documentation

8.981.2.1 `voice_airTimer*` `unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt`

8.981.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetConfig_t::ParamPresenceMask`

8.981.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat`

8.981.2.4 `voice_curAMRConfig*` `unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig`

8.981.2.5 `voice_prefVoiceSO*` `unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO`

8.981.2.6 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode`

8.981.2.7 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref`

8.981.2.8 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref`

8.981.2.9 `voice_roamTimer*` `unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt`

8.982 `unpack_voice_SLQSVoiceManageCalls_t` Struct Reference

Data Fields

- uint16_t * [pFailCause](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.982.1 Detailed Description

This structure contains Failure cause Information. Populated when API Fails.

Parameters

<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary service failure causes (optional, supply NULL if not required). • See Table8 qaGobiApiTableVoiceCallEndReasons.h for supplementary services failure cause <ul style="list-style-type: none"> – 0xFFFF is the value when the information is not received from device • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.982.2 Field Documentation

8.982.2.1 [swi_uint256_t](#) [unpack_voice_SLQSVoiceManageCalls_t::ParamPresenceMask](#)

8.982.2.2 [uint16_t*](#) [unpack_voice_SLQSVoiceManageCalls_t::pFailCause](#)

8.983 unpack_voice_SLQSVoiceSendFlash_t Struct Reference

Data Fields

- uint8_t * [pCallID](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.983.1 Detailed Description

This structure contains the flash information associated with a call.

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> • Unique call identifier associated with the current call. • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.983.2 Field Documentation

8.983.2.1 [swi_uint256_t](#) [unpack_voice_SLQSVoiceSendFlash_t::ParamPresenceMask](#)

8.983.2.2 uint8_t* unpack_voice_SLQSVoiceSendFlash_t::pCallID

8.984 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](#)

8.984.1 Detailed Description

This structure contains Voice Set Call Barring Password Unpack Parameters

Parameters

<i>pFailCause</i>	<ul style="list-style-type: none"> • Supplementary services failure cause (optional) • see qaGobiApiTableVoiceCallEndReasons.h for more information. • 0xFFFF,if Not Available • Bit to check in ParamPresenceMask - 16
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 17
<i>pCCResType</i>	<ul style="list-style-type: none"> • Call Control Result Type (optional) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - Not Available • Bit to check in ParamPresenceMask - 18
<i>pCallID</i>	<ul style="list-style-type: none"> • Call ID of the voice call that resulted from call control. (optional) • It is present when pCCResType is present and is Voice. • If zero(0) then invalid. • Bit to check in ParamPresenceMask - 19
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Supplementary service data that resulted from call control (optional) • Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 20
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.984.2 Field Documentation

8.984.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphaIDInfo`

8.984.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::ParamPresenceMask`

8.984.2.3 `uint8_t*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID`

8.984.2.4 `uint8_t*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType`

8.984.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType`

8.984.2.6 `uint16_t*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause`

8.985 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](#)

8.985.1 Detailed Description

This structure contains information about the Set Configuration Response Parameters.

Parameters

<i>pAutoAnsStatus</i>	<ul style="list-style-type: none"> • Auto Answer Status. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. • Bit to check in ParamPresenceMask - 16
<i>pAirTimerStatus</i>	<ul style="list-style-type: none"> • Air Timer Status. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. • Bit to check in ParamPresenceMask - 17

<i>pRoamTimer-Status</i>	<ul style="list-style-type: none"> • Roam Timer Status. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. • Bit to check in ParamPresenceMask - 18
<i>pTTYConfig-Status</i>	<ul style="list-style-type: none"> • TTY Config Status. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. • Bit to check in ParamPresenceMask - 19
<i>pPrefVoiceSO-Status</i>	<ul style="list-style-type: none"> • Preferred Voice SO Status. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. • Bit to check in ParamPresenceMask - 20
<i>pVoiceDomain-PrefStatus</i>	<ul style="list-style-type: none"> • Voice-Domain Preference Status. (optional) • Values: <ul style="list-style-type: none"> – 0x00 - Information was written successfully – 0x01 - Information write failed – 0xFF - Not Available. • Bit to check in ParamPresenceMask - 21
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.985.2 Field Documentation

8.985.2.1 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus`

8.985.2.2 `swi_uint256_t unpack_voice_SLQSVoiceSetConfig_t::ParamPresenceMask`

8.985.2.3 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus`

8.985.2.4 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOStatus`

8.985.2.5 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus`

8.985.2.6 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus`

8.985.2.7 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus`

8.986 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](#)

8.986.1 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"> • Supplementary service failure causes (optional, supply NULL if not required). <ul style="list-style-type: none"> – 0xFFFF is the value when the information is not received from device • Bit to check in ParamPresenceMask - 16
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> • Pointer to structure of alphaIDInfo. 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"> – See voice_alphaIDInfo for more information • Bit to check in ParamPresenceMask - 17
<i>pCCResultType</i>	<ul style="list-style-type: none"> • Call control result types (optional, supply NULL if not required) <ul style="list-style-type: none"> – 0x00 - CC_RESULT_TYPE_VOICE - Voice – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service – 0xFF - if the device does not provide this information • Bit to check in ParamPresenceMask - 19
<i>pCallID</i>	<ul style="list-style-type: none"> • Unique call identifier for the dialed call (optional, supply NULL if not required) <ul style="list-style-type: none"> – 0x00 - if the device does not provide this information • Bit to check in ParamPresenceMask - 18
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> • Data is present when pCCResultType is present and is other than Voice. (optional, supply NULL if not required) <ul style="list-style-type: none"> – See voice_ccSUPSType for more information • Bit to check in ParamPresenceMask - 20
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.986.2 Field Documentation

8.986.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceSetSUPSService_t::pAlphaIDInfo`

8.986.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceSetSUPSService_t::ParamPresenceMask`

8.986.2.3 `uint8_t*` `unpack_voice_SLQSVoiceSetSUPSService_t::pCallID`

8.986.2.4 `uint8_t*` `unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType`

8.986.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType`

8.986.2.6 `uint16_t*` `unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause`

8.987 `unpack_voice_SLQSVoiceStartContDTMF_t` Struct Reference

Data Fields

- `uint8_t *` `pCallID`
- `swi_uint256_t` `ParamPresenceMask`

8.987.1 Detailed Description

This structure contains parameters of continuous DTMF

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> • 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. • 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. • If the call ID value received is 0, no value has been returned by the device • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.987.2 Field Documentation

8.987.2.1 `swi_uint256_t` `unpack_voice_SLQSVoiceStartContDTMF_t::ParamPresenceMask`

8.987.2.2 `uint8_t*` `unpack_voice_SLQSVoiceStartContDTMF_t::pCallID`

8.988 `unpack_voice_SLQSVoiceStopContDTMF_t` Struct Reference

Data Fields

- `uint8_t` `callID`
- `swi_uint256_t` `ParamPresenceMask`

8.988.1 Detailed Description

This structure contains parameters of stop continuous DTMF

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> • 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. • 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. • If the call ID value received is 0, no value has been returned by the device
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.988.2 Field Documentation

8.988.2.1 `uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID`

8.988.2.2 `swi_uint256_t unpack_voice_SLQSVoiceStopContDTMF_t::ParamPresenceMask`

8.989 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`

8.989.1 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"> See voice_SUPSInfo for more information. Bit to check in ParamPresenceMask - 1
<i>pSvc- Class(optional)</i>	<ul style="list-style-type: none"> Service class is a combination (sum) of information class constants (optional) See qaGobiApiTableSupServiceInfoClasses.h for service classes. Bit to check in ParamPresenceMask - 16
<i>p- Reason(optional)</i>	<ul style="list-style-type: none"> See qaGobiApiTableCallControlReturnReasons.h for return reasons. Bit to check in ParamPresenceMask - 17
<i>pCallFW- Num(optional)</i>	<ul style="list-style-type: none"> Call forwarding number to be registered with the network. ASCII String, NULL terminated. Bit to check in ParamPresenceMask - 18
<i>pCallFWTimer- Val(optional)</i>	<ul style="list-style-type: none"> Call Forwarding No Reply Timer. <ul style="list-style-type: none"> Range: 5 to 30 in steps of 5. Bit to check in ParamPresenceMask - 19
<i>pUSS- Info(optional)</i>	<ul style="list-style-type: none"> See voice_USSInfo for more information. Bit to check in ParamPresenceMask - 20
<i>pCallID(optional)</i>	<ul style="list-style-type: none"> Call identifier of the voice call that has been modified to a supplementary service as a result of call control. Bit to check in ParamPresenceMask - 21
<i>pAlphaID- Info(optional)</i>	<ul style="list-style-type: none"> See voice_alphaIDInfo for more information. Bit to check in ParamPresenceMask - 22
<i>pCallBar- Passwd(optional)</i>	<ul style="list-style-type: none"> Password is required if call barring is provisioned using a password. <ul style="list-style-type: none"> Password consists of 4 ASCII digits. Range: 0000 to 9999. This also serves as the old password in the register password scenario. Bit to check in ParamPresenceMask - 23
<i>pNewPwd- Data(optional)</i>	<ul style="list-style-type: none"> See voice_newPwdData for more information. Bit to check in ParamPresenceMask - 24

<i>pData-Src(optional)</i>	<ul style="list-style-type: none"> • Sups Data Source. • Used to distinguish between the supplementary service data sent to the network and the response received from the network. • If absent, the supplementary service data in this indication can be assumed as a request sent to the network. • Bit to check in ParamPresenceMask - 25
<i>pFail-Cause(optional)</i>	<ul style="list-style-type: none"> • Supplementary services failure cause. • See qaGobiApiTableVoiceCallEndReasons.h for more information. • Bit to check in ParamPresenceMask - 26
<i>pCallFwd-Info(optional)</i>	<ul style="list-style-type: none"> • See voice_getCallFWInfo for more information. • Bit to check in ParamPresenceMask - 27
<i>pCLI-Rstatus(optional)</i>	<ul style="list-style-type: none"> • See voice_CLIRResp for more information. • Bit to check in ParamPresenceMask - 28
<i>pCLI-Pstatus(optional)</i>	<ul style="list-style-type: none"> • See voice_CLIPResp for more information. • Bit to check in ParamPresenceMask - 29
<i>pCOL-Pstatus(optional)</i>	<ul style="list-style-type: none"> • See voice_COLPResp for more information. • Bit to check in ParamPresenceMask - 30
<i>pCOL-Rstatus(optional)</i>	<ul style="list-style-type: none"> • See voice_COLRResp for more information. • Bit to check in ParamPresenceMask - 31
<i>pCNA-Pstatus(optional)</i>	<ul style="list-style-type: none"> • See voice_CNAPResp for more information. • Bit to check in ParamPresenceMask - 32
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

None

8.989.2 Field Documentation

8.989.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pAlphaIDInfo`8.989.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::ParamPresenceMask`8.989.2.3 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallBarPasswd`

- 8.989.2.4 `voice_getCallFWInfo*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFwdInfo`
- 8.989.2.5 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWNum`
- 8.989.2.6 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWTimerVal`
- 8.989.2.7 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallID`
- 8.989.2.8 `voice_CLIPResp*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIPstatus`
- 8.989.2.9 `voice_CLIRResp*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIRstatus`
- 8.989.2.10 `voice_CNAPResp*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCNAPstatus`
- 8.989.2.11 `voice_COLPResp*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLPstatus`
- 8.989.2.12 `voice_COLRResp*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLRstatus`
- 8.989.2.13 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pDataSrc`
- 8.989.2.14 `uint16_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pFailCause`
- 8.989.2.15 `voice_newPwdData*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pNewPwdData`
- 8.989.2.16 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pReason`
- 8.989.2.17 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pSvcClass`
- 8.989.2.18 `struct voice_USSInfo*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pUSSInfo`
- 8.989.2.19 `voice_SUPSInfo` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::SUPSInformation`

8.990 `unpack_voice_SUPSNotificationCallback_ind_t` Struct Reference

Data Fields

- `uint8_t` `callID`
- `uint8_t` `notifType`
- `uint16_t *` `pCUGIndex`
- `voice_ECTNum *` `pECTNum`
- `swi_uint256_t` `ParamPresenceMask`

8.990.1 Detailed Description

Contains the parameters passed for SUPS notification indication by the device.

Parameters

<i>callID</i>	<ul style="list-style-type: none"> • Unique identifier of the call for which the notification is applicable. (mandatory) • Bit to check in <code>ParamPresenceMask</code> - 1
---------------	---

<i>notifType</i>	<ul style="list-style-type: none"> • Notification type parameter (mandatory) <ul style="list-style-type: none"> – 0x01 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_FORWARDED Originated MO call is being forwarded to another user – 0x02 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_WAITING Originated MO call is waiting at the called user – 0x03 - NOTIFICATION_TYPE_OUTGOING_CUG_CALL Outgoing call is a CUG call – 0x04 - NOTIFICATION_TYPE_OUTGOING_CALLS_BARRED Outgoing calls are barred – 0x05 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_DEFLECTED Outgoing call is deflected – 0x06 - NOTIFICATION_TYPE_INCOMING_CUG_CALL Incoming call is a CUG call – 0x07 - NOTIFICATION_TYPE_INCOMING_CALLS_BARRED Incoming calls are barred – 0x08 - NOTIFICATION_TYPE_INCOMING_FORWARDED_CALL Incoming call received is a forwarded call – 0x09 - NOTIFICATION_TYPE_INCOMING_DEFLECTED_CALL Incoming call is a deflected call – 0x0A - NOTIFICATION_TYPE_INCOMING_CALL_IS_FORWARDED Incoming call is forwarded to another user – 0x0B - NOTIFICATION_TYPE_UNCOND_CALL_FORWARD_ACTIVE Unconditional call forwarding is active – 0x0C - NOTIFICATION_TYPE_COND_CALL_FORWARD_ACTIVE Conditional call forwarding is active – 0x0D - NOTIFICATION_TYPE_CLIR_SUPPRESSION_REJECTED CLIR suppression is rejected – 0x0E - NOTIFICATION_TYPE_CALL_IS_ON_HOLD Call is put on hold at the remote party – 0x0F - NOTIFICATION_TYPE_CALL_IS_RETRIEVED Call is retrieved at the remote party from the hold state – 0x10 - NOTIFICATION_TYPE_CALL_IS_IN_MPTY Call is in a conference – 0x11 - NOTIFICATION_TYPE_INCOMING_CALL_IS_ECT Incoming call is an explicit call transfer • Bit to check in ParamPresenceMask - 1
<i>pCUGIndex</i>	<ul style="list-style-type: none"> • The CUG Index used to indicate that the incoming/outgoing call is a CUG call. (optional, NULL when not present) Range: 0x00 to 0x7FFF. • Bit to check in ParamPresenceMask - 16
<i>pECTNum</i>	<ul style="list-style-type: none"> • 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. • Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

None

8.990.2 Field Documentation

8.990.2.1 `uint8_t unpack_voice_SUPSNotificationCallback_ind_t::callID`8.990.2.2 `uint8_t unpack_voice_SUPSNotificationCallback_ind_t::notifType`8.990.2.3 `swi_uint256_t unpack_voice_SUPSNotificationCallback_ind_t::ParamPresenceMask`8.990.2.4 `uint16_t* unpack_voice_SUPSNotificationCallback_ind_t::pCUGIndex`8.990.2.5 `voice_ECTNum* unpack_voice_SUPSNotificationCallback_ind_t::pECTNum`

8.991 `unpack_voice_USSDNotificationCallback_ind_t` Struct Reference

Data Fields

- `uint8_t notification_Type`
- `voice_USSDNotificationNetworkInfo` [USSDNotificationNetworkInfo](#)
- `swi_uint256_t ParamPresenceMask`

8.991.1 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 - 1
<i>USSD-Notification-Network-Info(optional)</i>	• USSD network info (See voice_USSDNotificationNetworkInfo) – Bit to check in ParamPresenceMask - 16
<i>ParamPresence-Mask</i>	• bitmask representation to indicate valid parameters.

8.991.2 Field Documentation

8.991.2.1 `uint8_t unpack_voice_USSDNotificationCallback_ind_t::notification_Type`8.991.2.2 `swi_uint256_t unpack_voice_USSDNotificationCallback_ind_t::ParamPresenceMask`8.991.2.3 `voice_USSDNotificationNetworkInfo` `unpack_voice_USSDNotificationCallback_ind_t::USSDNotification-NetworkInfo`

8.992 `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](#)

8.992.1 Detailed Description

This structure contains Voice record Information

Parameters

<i>callID</i>	[Mandatory] <ul style="list-style-type: none"> • Call identifier for the call. • Bit to check in ParamPresenceMask - 1
<i>pSignalInfo</i> [- Optional]	<ul style="list-style-type: none"> • Signal Information • See voice_signalInfo for more information • Bit to check in ParamPresenceMask - 16
<i>pCallerIDInfo</i> [- Optional]	<ul style="list-style-type: none"> • Caller ID Information • See voice_callerIDInfo for more information • Bit to check in ParamPresenceMask - 17
<i>pDispInfo</i> [- Optional]	<ul style="list-style-type: none"> • Display Information • Bit to check in ParamPresenceMask - 18
<i>pExtDispInfo</i> [- Optional]	<ul style="list-style-type: none"> • Extended Display Information • Bit to check in ParamPresenceMask - 19
<i>pCallerName- Info</i> [Optional]	<ul style="list-style-type: none"> • Caller Name Information • Bit to check in ParamPresenceMask - 20

<i>pCallWaitInd[-Optional]</i>	<ul style="list-style-type: none"> • Call Waiting Indicator • Bit to check in ParamPresenceMask - 21
<i>pConnectNumInfo[Optional]</i>	<ul style="list-style-type: none"> • Connected Number Information • see voice_connectNumInfo for more information • Bit to check in ParamPresenceMask - 22
<i>pCallingPartyInfo[Optional]</i>	<ul style="list-style-type: none"> • Calling Party Number Information • This structure is having exactly same elements as connectNumInfo • see voice_connectNumInfo for more information • Bit to check in ParamPresenceMask - 23
<i>pCalledPartyInfo[Optional]</i>	<ul style="list-style-type: none"> • Called Party Number Information • see voice_calledPartyInfo for more information • Bit to check in ParamPresenceMask - 24
<i>pRedirNumInfo[-Optional]</i>	<ul style="list-style-type: none"> • Redirecting Number Information • see voice_redirNumInfo for more information • Bit to check in ParamPresenceMask - 25
<i>pCLIRCause[-Optional]</i>	<ul style="list-style-type: none"> • National Supplementary Services - CLIR • see voice_NSSAudioCtrl for more information • Bit to check in ParamPresenceMask - 26
<i>pNSSAudioCtrl[-Optional]</i>	<ul style="list-style-type: none"> • National Supplementary Services - Audio Control • Bit to check in ParamPresenceMask - 27
<i>pNSSRelease[-Optional]</i>	<ul style="list-style-type: none"> • National Supplementary Services - Release • Bit to check in ParamPresenceMask - 28
<i>pLineCtrlInfo[-Optional]</i>	<ul style="list-style-type: none"> • Line Control Information • see voice_lineCtrlInfo for more information • Bit to check in ParamPresenceMask - 29
<i>pExtDispRecInfo[Optional]</i>	<ul style="list-style-type: none"> • Extended Display Record Information • see voice_extDispRecInfo for more information • Bit to check in ParamPresenceMask - 30
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.992.2 Field Documentation

- 8.992.2.1 `uint8_t unpack_voice_VoiceInfoRecCallback_ind_t::callID`
- 8.992.2.2 `swi_uint256_t unpack_voice_VoiceInfoRecCallback_ind_t::ParamPresenceMask`
- 8.992.2.3 `voice_calledPartyInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCalledPartyInfo`
- 8.992.2.4 `voice_callerIDInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerIDInfo`
- 8.992.2.5 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerNameInfo`
- 8.992.2.6 `voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallingPartyInfo`
- 8.992.2.7 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallWaitInd`
- 8.992.2.8 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCLIRCause`
- 8.992.2.9 `voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pConnectNumInfo`
- 8.992.2.10 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pDisplInfo`
- 8.992.2.11 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDisplInfo`
- 8.992.2.12 `voice_extDispRecInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDispRecInfo`
- 8.992.2.13 `voice_lineCtrlInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pLineCtrlInfo`
- 8.992.2.14 `voice_NSSAudioCtrl* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSAudioCtrl`
- 8.992.2.15 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSRelease`
- 8.992.2.16 `voice_redirNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pRedirNumInfo`
- 8.992.2.17 `voice_signalInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pSignalInfo`

8.993 unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference

Data Fields

- `uint8_t callID`
- `uint8_t voicePrivacy`
- `swi_uint256_t ParamPresenceMask`

8.993.1 Detailed Description

Contains the parameters passed for voice privacy change indication

Parameters

<i>callID</i>	<ul style="list-style-type: none">• Unique identifier of the call for which the voice privacy is applicable. (mandatory)• Bit to check in ParamPresenceMask - 1
---------------	---

<i>voicePrivacy</i>	<ul style="list-style-type: none"> Voice Privacy (mandatory) <ul style="list-style-type: none"> 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

Note

None

8.993.2 Field Documentation

8.993.2.1 `uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::callID`

8.993.2.2 `swi_uint256_t unpack_voice_voicePrivacyChangeCallback_ind_t::ParamPresenceMask`

8.993.2.3 `uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::voicePrivacy`

8.994 `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](#)

8.994.1 Detailed Description

This structure contains DHCP IPv4 client lease information

Parameters

<i>ProfileIdTlv</i>	profile identifier information, see wds_DHCPProfileIdTlv for more details <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 1
<i>DHCPv4LeaseStateTlv</i>	lease state, see wds_DHCPLeaseStateTlv for more details <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 2
<i>IPv4AddrTlv</i>	IPv4 address, see wds_IPv4AddrTlv for more details <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 16
<i>DHCPv4LeaseOptTlv</i>	DHCP lease option, see wds_DHCPLeaseOptTlv for more details <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 17
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.994.2 Field Documentation

8.994.2.1 wds_DHCPLeaseOptTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv

8.994.2.2 wds_DHCPLeaseStateTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv

8.994.2.3 wds_IPv4AdTlv unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv

8.994.2.4 swi_uint256_t unpack_wds_DHCPv4ClientLease_ind_t::ParamPresenceMask

8.994.2.5 wds_DHCPProfileIdTlv unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv

8.995 unpack_wds_GetAutoconnect_t Struct Reference

Data Fields

- uint32_t * [psetting](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.995.1 Detailed Description

auto connect data session setting parameter.

Parameters

<i>pSetting</i>	<ul style="list-style-type: none"> • NDIS auto connect setting <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.995.2 Field Documentation

8.995.2.1 swi_uint256_t unpack_wds_GetAutoconnect_t::ParamPresenceMask

8.995.2.2 uint32_t* unpack_wds_GetAutoconnect_t::psetting

8.996 unpack_wds_GetByteTotals_t Struct Reference

Data Fields

- uint64_t * [pTXTotalBytes](#)
- uint64_t * [pRXTotalBytes](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.996.1 Detailed Description

This structure contains unpack get byte totals information.

Parameters

<i>pTXTotalBytes</i>	<ul style="list-style-type: none"> Bytes transmitted without error Bit to check in ParamPresenceMask - 25
<i>pRXTotalBytes</i>	<ul style="list-style-type: none"> Bytes received without error Bit to check in ParamPresenceMask - 26
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.996.2 Field Documentation

8.996.2.1 `swi_uint256_t unpack_wds_GetByteTotals_t::ParamPresenceMask`8.996.2.2 `uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes`8.996.2.3 `uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes`8.997 `unpack_wds_GetConnectionRate_t` Struct Reference

Data Fields

- `uint32_t` [currentChannelTXRate](#)
- `uint32_t` [currentChannelRXRate](#)
- `uint32_t` [maxChannelTXRate](#)
- `uint32_t` [maxChannelRXRate](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.997.1 Detailed Description

This structure contains unpack get connection rate information.

Parameters

<i>currentChannelTXRate</i>	<ul style="list-style-type: none"> Current channel Tx rate (in bps) Bit to check in ParamPresenceMask - 1
<i>currentChannelRXRate</i>	<ul style="list-style-type: none"> Current channel Rx rate (in bps) Bit to check in ParamPresenceMask - 1
<i>maxChannelTXRate</i>	<ul style="list-style-type: none"> Maximum Tx rate (bps) that may be assigned to device by serving system. Bit to check in ParamPresenceMask - 1
<i>maxChannelRXRate</i>	<ul style="list-style-type: none"> Maximum Rx rate (bps) that may be assigned to device by serving system. Bit to check in ParamPresenceMask - 1

<i>ParamPresenceMask</i>	<ul style="list-style-type: none">• bitmask representation to indicate valid parameters.
--------------------------	--

8.997.2 Field Documentation

8.997.2.1 `uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate`

8.997.2.2 `uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate`

8.997.2.3 `uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate`

8.997.2.4 `uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate`

8.997.2.5 `swi_uint256_t unpack_wds_GetConnectionRate_t::ParamPresenceMask`

8.998 unpack_wds_GetDataBearerTechnology_t Struct Reference

Data Fields

- `uint32_t * pDataBearer`
- `swi_uint256_t ParamPresenceMask`

8.998.1 Detailed Description

This structure contains unpack get data bearer technology information.

Parameters

<i>pDataBearer[O-UT]</i>	<ul style="list-style-type: none"> • Data bearer technology <ul style="list-style-type: none"> – 0x01 - CDMA2000 1x – 0x02 - CDMA 1xEV-DO Rev 0 – 0x03 - GSM – 0x04 - UMTS – 0x05 - CDMA2000 HRPD (1xEV-DO Rev A) – 0x06 - EDGE – 0x07 - HSDPA AND WCDMA – 0x08 - WCDMA AND HSUPA – 0x09 - HSDPA AND HSUPA – 0x0A - LTE – 0x0B - CDMA2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 0xFF - Unknown • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.998.2 Field Documentation

8.998.2.1 `swi_uint256_t unpack_wds_GetDataBearerTechnology_t::ParamPresenceMask`

8.998.2.2 `uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer`

8.999 `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 apnsize`
- `int8_t apnname [255]`

- [uint8_t usersize](#)
- [int8_t username](#) [255]
- [swi_uint256_t ParamPresenceMask](#)

8.999.1 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"> • 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"> – 0 - PDP-IP (IPv4) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6 • Bit to check in ParamPresenceMask - 17
<i>ipaddr</i>	<ul style="list-style-type: none"> • Preferred IPv4 address to be assigned to device • Bit to check in ParamPresenceMask - 30
<i>pridns</i>	<ul style="list-style-type: none"> • Primary DNS IPv4 address preference • Bit to check in ParamPresenceMask - 21
<i>secdns</i>	<ul style="list-style-type: none"> • Secondary DNS IPv4 address preference • Bit to check in ParamPresenceMask - 22
<i>ipaddrv6</i>	<ul style="list-style-type: none"> • Preferred IPv6 address to be assigned to device • Bit to check in ParamPresenceMask - 143
<i>pridnsv6</i>	<ul style="list-style-type: none"> • Primary DNS Pv6 address preference • Bit to check in ParamPresenceMask - 143
<i>secdnsv6</i>	<ul style="list-style-type: none"> • Secondary DNS IPv6 address preference • Bit to check in ParamPresenceMask - 143

<i>auth</i>	<ul style="list-style-type: none"> • Bitmap that indicates authentication algorithm preference <ul style="list-style-type: none"> – 0x00000001 - PAP preference <ul style="list-style-type: none"> * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference <ul style="list-style-type: none"> * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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. • Bit to check in ParamPresenceMask - 29
<i>namesize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that profile name array can contain. • Bit to check in ParamPresenceMask - 16
<i>name</i>	<ul style="list-style-type: none"> • Profile name • Bit to check in ParamPresenceMask - 16
<i>apnsize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that APN name array can contain • Bit to check in ParamPresenceMask - 20
<i>apnname</i>	<ul style="list-style-type: none"> • Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network. • If value is NULL or omitted, then subscription default value will be requested. • Bit to check in ParamPresenceMask - 20
<i>usersize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that username array can contain. • Bit to check in ParamPresenceMask - 27
<i>username</i>	<ul style="list-style-type: none"> • Username used during network authentication • Bit to check in ParamPresenceMask - 27
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.999.2 Field Documentation

8.999.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

8.999.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

8.999.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

- 8.999.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr
- 8.999.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6[8]
- 8.999.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]
- 8.999.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize
- 8.999.2.8 swi_uint256_t unpack_wds_GetDefaultProfile_t::ParamPresenceMask
- 8.999.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pdptype
- 8.999.2.10 uint32_t unpack_wds_GetDefaultProfile_t::pridns
- 8.999.2.11 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6[8]
- 8.999.2.12 uint32_t unpack_wds_GetDefaultProfile_t::secdns
- 8.999.2.13 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6[8]
- 8.999.2.14 int8_t unpack_wds_GetDefaultProfile_t::username[255]
- 8.999.2.15 uint8_t unpack_wds_GetDefaultProfile_t::usersize

8.1000 unpack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [index](#)
- [swi_uint256_t ParamPresenceMask](#)

8.1000.1 Detailed Description

This structure contains unpack Get Default Profile Number information.

Parameters

<i>index</i>	profile index <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1000.2 Field Documentation

- 8.1000.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index
- 8.1000.2.2 swi_uint256_t unpack_wds_GetDefaultProfileNum_t::ParamPresenceMask

8.1001 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](#)

8.1001.1 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"> • 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"> – 0 - PDP-IP (IPv4) – 1 - PDP-PPP – 2 - PDP-IPv6 – 3 - PDP-IPv4v6 • Bit to check in ParamPresenceMask - 17
<i>ipaddr</i>	<ul style="list-style-type: none"> • Preferred IPv4 address to be assigned to device • Bit to check in ParamPresenceMask - 30
<i>pridns</i>	<ul style="list-style-type: none"> • Primary DNS IPv4 address preference • Bit to check in ParamPresenceMask - 21
<i>secdns</i>	<ul style="list-style-type: none"> • Secondary DNS IPv4 address preference • Bit to check in ParamPresenceMask - 22
<i>ipaddrv6</i>	<ul style="list-style-type: none"> • Preferred IPv6 address to be assigned to device • Bit to check in ParamPresenceMask - 143

<i>pridnsv6</i>	<ul style="list-style-type: none"> • Primary DNS Pv6 address preference • Bit to check in ParamPresenceMask - 143
<i>secdnsv6</i>	<ul style="list-style-type: none"> • Secondary DNS IPv6 address preference • Bit to check in ParamPresenceMask - 143
<i>auth</i>	<ul style="list-style-type: none"> • Bitmap that indicates authentication algorithm preference <ul style="list-style-type: none"> – 0x00000001 - PAP preference <ul style="list-style-type: none"> * 0 - Never performed * 1 - May be performed – 0x00000002 - CHAP preference <ul style="list-style-type: none"> * 0 - Never performed * 1 - May be performed – All other bits are reserved and must be set to 0 – 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. • Bit to check in ParamPresenceMask - 29
<i>namesize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that profile name array can contain. • Bit to check in ParamPresenceMask - 16
<i>name</i>	<ul style="list-style-type: none"> • Profile name • Bit to check in ParamPresenceMask - 16
<i>apnsize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that APN name array can contain • Bit to check in ParamPresenceMask - 20
<i>apnname</i>	<ul style="list-style-type: none"> • Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network. • If value is NULL or omitted, then subscription default value will be requested. • Bit to check in ParamPresenceMask - 20
<i>usersize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that username array can contain. • Bit to check in ParamPresenceMask - 27
<i>username</i>	<ul style="list-style-type: none"> • Username used during network authentication • Bit to check in ParamPresenceMask - 27
<i>pwdsize</i>	<ul style="list-style-type: none"> • Maximum number of characters (including NULL terminator) that pwd name array can contain. • Bit to check in ParamPresenceMask - 28

<i>pwd</i>	<ul style="list-style-type: none"> • Password used during network authentication • Bit to check in ParamPresenceMask - 28
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1001.2 Field Documentation

- 8.1001.2.1 `int8_t unpack_wds_GetDefaultProfileV2_t::apnname[255]`
- 8.1001.2.2 `uint8_t unpack_wds_GetDefaultProfileV2_t::apnsize`
- 8.1001.2.3 `uint32_t unpack_wds_GetDefaultProfileV2_t::auth`
- 8.1001.2.4 `uint32_t unpack_wds_GetDefaultProfileV2_t::ipaddr`
- 8.1001.2.5 `uint16_t unpack_wds_GetDefaultProfileV2_t::ipaddrv6[8]`
- 8.1001.2.6 `int8_t unpack_wds_GetDefaultProfileV2_t::name[255]`
- 8.1001.2.7 `uint8_t unpack_wds_GetDefaultProfileV2_t::namesize`
- 8.1001.2.8 `swi_uint256_t unpack_wds_GetDefaultProfileV2_t::ParamPresenceMask`
- 8.1001.2.9 `uint32_t unpack_wds_GetDefaultProfileV2_t::pdptype`
- 8.1001.2.10 `uint32_t unpack_wds_GetDefaultProfileV2_t::pridns`
- 8.1001.2.11 `uint16_t unpack_wds_GetDefaultProfileV2_t::pridnsv6[8]`
- 8.1001.2.12 `int8_t unpack_wds_GetDefaultProfileV2_t::pwd[255]`
- 8.1001.2.13 `uint8_t unpack_wds_GetDefaultProfileV2_t::pwdsize`
- 8.1001.2.14 `uint32_t unpack_wds_GetDefaultProfileV2_t::secdns`
- 8.1001.2.15 `uint16_t unpack_wds_GetDefaultProfileV2_t::secdnsv6[8]`
- 8.1001.2.16 `int8_t unpack_wds_GetDefaultProfileV2_t::username[255]`
- 8.1001.2.17 `uint8_t unpack_wds_GetDefaultProfileV2_t::usersize`

8.1002 `unpack_wds_GetDormancyState_t` Struct Reference

Data Fields

- `uint32_t dormancyState`
- `swi_uint256_t ParamPresenceMask`

8.1002.1 Detailed Description

This structure contains unpack get dormancy state information.

Parameters

<i>dormancyState</i>	<ul style="list-style-type: none"> Dormancy state of current packet data session <ul style="list-style-type: none"> 1 - Traffic channel dormant 2 - Traffic channel active Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1002.2 Field Documentation

8.1002.2.1 `uint32_t unpack_wds_GetDormancyState_t::dormancyState`

8.1002.2.2 `swi_uint256_t unpack_wds_GetDormancyState_t::ParamPresenceMask`

8.1003 unpack_wds_GetLastMobileIPError_t Struct Reference

Data Fields

- `uint32_t error`
- `swi_uint256_t ParamPresenceMask`

8.1003.1 Detailed Description

structure to store last mobile IP error.

Parameters

<i>error</i>	<ul style="list-style-type: none"> Status of last MIP call (or attempt) <ul style="list-style-type: none"> Zero - Success NonZero - Error code See qaGobiApiTableCallEndReasons.h for Mobile IP Error codes Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1003.2 Field Documentation

8.1003.2.1 `uint32_t unpack_wds_GetLastMobileIPError_t::error`

8.1003.2.2 `swi_uint256_t unpack_wds_GetLastMobileIPError_t::ParamPresenceMask`

8.1004 unpack_wds_GetMobileIP_t Struct Reference

Data Fields

- uint32_t [mipMode](#)
- swi_uint256_t [ParamPresenceMask](#)

8.1004.1 Detailed Description

This structure contains unpack get mobile IP information.

Parameters

<i>mipMode</i>	<ul style="list-style-type: none"> • Mobile IP setting <ul style="list-style-type: none"> – 0 - Mobile IP off (simple IP only) – 1 - Mobile IP preferred – 2 - Mobile IP only • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1004.2 Field Documentation

8.1004.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.1004.2.2 swi_uint256_t unpack_wds_GetMobileIP_t::ParamPresenceMask

8.1005 unpack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t [enabled](#)
- uint32_t [address](#)
- uint32_t [primaryHA](#)
- uint32_t [secondaryHA](#)
- uint8_t [revTunneling](#)
- uint8_t [naiSize](#)
- int8_t [NAI](#) [255]
- uint32_t [HASPI](#)
- uint32_t [AAASPI](#)
- uint32_t [HASState](#)
- uint32_t [AAASState](#)
- swi_uint256_t [ParamPresenceMask](#)

8.1005.1 Detailed Description

This structure contains unpack get mobile IP profile information.

Parameters

<i>enabled</i>	<ul style="list-style-type: none"> • Profile enabled: <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 16
<i>address</i>	<ul style="list-style-type: none"> • Home IPv4 address: • Bit to check in ParamPresenceMask - 17
<i>primaryHA</i>	<ul style="list-style-type: none"> • Primary home agent IPv4 address • Bit to check in ParamPresenceMask - 18
<i>secondaryHA</i>	<ul style="list-style-type: none"> • Secondary home agent IPv4 address • Bit to check in ParamPresenceMask - 19
<i>revTunneling</i>	<ul style="list-style-type: none"> • Reverse tunneling enabled <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled • Bit to check in ParamPresenceMask - 20
<i>naiSize</i>	<ul style="list-style-type: none"> • The maximum number of characters (including NULL terminator) that the NAI array can contain. • Bit to check in ParamPresenceMask - 21
<i>NAI</i>	<ul style="list-style-type: none"> • Network access identifier string • Bit to check in ParamPresenceMask - 21
<i>HASPI</i>	<ul style="list-style-type: none"> • Home agent security parameter index • Bit to check in ParamPresenceMask - 22
<i>AAASPI</i>	<ul style="list-style-type: none"> • AAA server security parameter index • Bit to check in ParamPresenceMask - 23
<i>HASState</i>	<ul style="list-style-type: none"> • Home agent key state <ul style="list-style-type: none"> – 0 - Unset – 1 - Set, default value – 2 - Set, modified from default – 3 - 0xFFFFFFFF - Unknown • Bit to check in ParamPresenceMask - 26

<i>AAASState</i>	<ul style="list-style-type: none"> • AAA key state <ul style="list-style-type: none"> – 0 - Unset – 1 - Set, default value – 2 - Set, modified from default – 3 - 0xFFFFFFFF - Unknown • Bit to check in ParamPresenceMask - 27
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1005.2 Field Documentation

- 8.1005.2.1 `uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI`
- 8.1005.2.2 `uint32_t unpack_wds_GetMobileIPProfile_t::AAASState`
- 8.1005.2.3 `uint32_t unpack_wds_GetMobileIPProfile_t::address`
- 8.1005.2.4 `uint8_t unpack_wds_GetMobileIPProfile_t::enabled`
- 8.1005.2.5 `uint32_t unpack_wds_GetMobileIPProfile_t::HASPI`
- 8.1005.2.6 `uint32_t unpack_wds_GetMobileIPProfile_t::HASState`
- 8.1005.2.7 `int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]`
- 8.1005.2.8 `uint8_t unpack_wds_GetMobileIPProfile_t::naiSize`
- 8.1005.2.9 `swi_uint256_t unpack_wds_GetMobileIPProfile_t::ParamPresenceMask`
- 8.1005.2.10 `uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA`
- 8.1005.2.11 `uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling`
- 8.1005.2.12 `uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA`

8.1006 `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

8.1006.1 Detailed Description

This structure contains unpack get packet statistics information.

Parameters

<i>pTXPacket-Successes</i>	<ul style="list-style-type: none"> • No of transmitted Packets without error. • Bit to check in ParamPresenceMask - 16
<i>pRXPacket-Successes</i>	<ul style="list-style-type: none"> • No of received Packets without error. • Bit to check in ParamPresenceMask - 17
<i>pTXPacketErrors</i>	<ul style="list-style-type: none"> • Number of outgoing packets with framing errors. • Bit to check in ParamPresenceMask - 18
<i>pRXPacket-Errors</i>	<ul style="list-style-type: none"> • Number of incoming packets with framing errors. • Bit to check in ParamPresenceMask - 19
<i>pTXPacket-Overflows</i>	<ul style="list-style-type: none"> • Number of packets dropped because Tx buffer overflowed (out of memory). • Bit to check in ParamPresenceMask - 20
<i>pRXPacket-Overflows</i>	<ul style="list-style-type: none"> • Number of packets dropped because Rx buffer overflowed (out of memory). • Bit to check in ParamPresenceMask - 21
<i>pTXOkBytes-Count</i>	<ul style="list-style-type: none"> • Number of bytes transmitted without error. • Bit to check in ParamPresenceMask - 25
<i>pRXOkBytes-Count</i>	<ul style="list-style-type: none"> • Number of bytes received without error. • Bit to check in ParamPresenceMask - 26
<i>pTXOKBytes-LastCall</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 27
<i>pRXOKBytes-LastCall</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 28

<i>pTXDropped-Count</i>	<ul style="list-style-type: none"> • Number of outgoing packets dropped. • Bit to check in ParamPresenceMask - 29
<i>pRXDropped-Count</i>	<ul style="list-style-type: none"> • Number of incoming packets dropped. • Bit to check in ParamPresenceMask - 30
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1006.2 Field Documentation

8.1006.2.1 `swi_uint256_t unpack_wds_GetPacketStatistics_t::ParamPresenceMask`

8.1006.2.2 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount`

8.1006.2.3 `uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount`

8.1006.2.4 `uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesLastCall`

8.1006.2.5 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors`

8.1006.2.6 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows`

8.1006.2.7 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses`

8.1006.2.8 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount`

8.1006.2.9 `uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount`

8.1006.2.10 `uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesLastCall`

8.1006.2.11 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors`

8.1006.2.12 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows`

8.1006.2.13 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses`

8.1007 `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

8.1007.1 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"> • No. of packets transmitted without error • Bit to check in ParamPresenceMask - 16
<i>rxPacket-Successes</i>	<ul style="list-style-type: none"> • No. of packets received without error • Bit to check in ParamPresenceMask - 17
<i>txPacketErrors</i>	<ul style="list-style-type: none"> • No. of outgoing packets with framing errors • Bit to check in ParamPresenceMask - 18
<i>rxPacketErrors</i>	<ul style="list-style-type: none"> • No. of incoming packets with framing errors • Bit to check in ParamPresenceMask - 19
<i>txPacket-Overflows</i>	<ul style="list-style-type: none"> • Number of packets dropped because Tx buffer overflowed • Bit to check in ParamPresenceMask - 20
<i>rxPacket-Overflows</i>	<ul style="list-style-type: none"> • Number of packets dropped because Rx buffer overflowed • Bit to check in ParamPresenceMask - 21
<i>txOkBytesCount</i>	<ul style="list-style-type: none"> • No of bytes transmitted without error. • Bit to check in ParamPresenceMask - 25
<i>rxOkBytesCount</i>	<ul style="list-style-type: none"> • No of bytes received without error. • Bit to check in ParamPresenceMask - 26
<i>txOKBytesLast-Call</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 27
<i>rxOKBytesLast-Call</i>	<ul style="list-style-type: none"> • 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 • Bit to check in ParamPresenceMask - 28

<i>txDroppedCount</i>	<ul style="list-style-type: none"> • Number of outgoing packets dropped. • Bit to check in ParamPresenceMask - 29
<i>rxDroppedCount</i>	<ul style="list-style-type: none"> • Number of incoming packets dropped. • Bit to check in ParamPresenceMask - 30
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1007.2 Field Documentation

8.1007.2.1 `swi_uint256_t unpack_wds_GetPacketStatus_t::ParamPresenceMask`

8.1007.2.2 `uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount`

8.1007.2.3 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount`

8.1007.2.4 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesLastCall`

8.1007.2.5 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors`

8.1007.2.6 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows`

8.1007.2.7 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses`

8.1007.2.8 `uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount`

8.1007.2.9 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount`

8.1007.2.10 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesLastCall`

8.1007.2.11 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors`

8.1007.2.12 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows`

8.1007.2.13 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses`

8.1008 `unpack_wds_GetSessionDuration_t` Struct Reference

Data Fields

- `uint64_t callDuration`
- `swi_uint256_t ParamPresenceMask`

8.1008.1 Detailed Description

This structure contains unpack get session duration information.

Parameters

<i>callDuration</i>	<ul style="list-style-type: none"> Duration of the current packet session in milliseconds Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1008.2 Field Documentation

8.1008.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

8.1008.2.2 swi_uint256_t unpack_wds_GetSessionDuration_t::ParamPresenceMask

8.1009 unpack_wds_GetSessionDurationV2_t Struct Reference

Data Fields

- uint64_t [callDuration](#)
- uint64_t * [pLastCallDuration](#)
- uint64_t * [pCallActiveDuration](#)
- uint64_t * [pLastCallActiveDuration](#)
- swi_uint256_t [ParamPresenceMask](#)

8.1009.1 Detailed Description

This structure contains unpack get session duration information -V2.

Parameters

<i>callDuration</i>	<ul style="list-style-type: none"> Duration of the current packet session in milliseconds Bit to check in ParamPresenceMask - 1
<i>pLastCallDuration</i>	<ul style="list-style-type: none"> Duration of the last data session in milliseconds Bit to check in ParamPresenceMask - 16
<i>pCallActiveDuration</i>	<ul style="list-style-type: none"> Duration of the active time of current data session in milliseconds Bit to check in ParamPresenceMask - 17
<i>pLastCallActiveDuration</i>	<ul style="list-style-type: none"> Duration of the active time of last data session in milliseconds Bit to check in ParamPresenceMask - 18
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1009.2 Field Documentation

8.1009.2.1 `uint64_t unpack_wds_GetSessionDurationV2_t::callDuration`

8.1009.2.2 `swi_uint256_t unpack_wds_GetSessionDurationV2_t::ParamPresenceMask`

8.1009.2.3 `uint64_t* unpack_wds_GetSessionDurationV2_t::pCallActiveDuration`

8.1009.2.4 `uint64_t* unpack_wds_GetSessionDurationV2_t::pLastCallActiveDuration`

8.1009.2.5 `uint64_t* unpack_wds_GetSessionDurationV2_t::pLastCallDuration`

8.1010 `unpack_wds_GetSessionState_t` Struct Reference

Data Fields

- `uint32_t connectionStatus`
- `swi_uint256_t ParamPresenceMask`

8.1010.1 Detailed Description

This structure contains unpack get session state information.

Parameters

<i>connection-Status</i>	- state of the current packet data session. <ul style="list-style-type: none"> • 1 - Disconnected. • 2 - Connected. • 3 - Suspended. • 4 - Authenticating. • Bit to check in ParamPresenceMask - 1
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1010.2 Field Documentation

8.1010.2.1 `uint32_t unpack_wds_GetSessionState_t::connectionStatus`

8.1010.2.2 `swi_uint256_t unpack_wds_GetSessionState_t::ParamPresenceMask`

8.1011 `unpack_wds_RMSetTransferStatistics_t` Struct Reference

Data Fields

- `swi_uint256_t ParamPresenceMask`

8.1011.1 Detailed Description

This structure contains unpack fetch current data system transfer statistics information.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none">bitmask representation to indicate valid parameters.
--------------------------	--

8.1011.2 Field Documentation

8.1011.2.1 `swi_uint256_t unpack_wds_RMSetTransferStatistics_t::ParamPresenceMask`

8.1012 unpack_wds_SetMobileIPProfile_t Struct Reference

Data Fields

- [swi_uint256_t ParamPresenceMask](#)

8.1012.1 Detailed Description

This structure contains set mobile IP profile information.

Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none">bitmask representation to indicate valid parameters.
--------------------------	--

8.1012.2 Field Documentation

8.1012.2.1 `swi_uint256_t unpack_wds_SetMobileIPProfile_t::ParamPresenceMask`

8.1013 unpack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

- [PackCreateProfileOut * pCreateProfileOut](#)
- `uint8_t * pProfileID`
- `uint16_t Tlvresult`
- [swi_uint256_t ParamPresenceMask](#)

8.1013.1 Detailed Description

This structure contains unpack create profile information.

Parameters

<i>pCreateProfileOut</i>	<ul style="list-style-type: none">SLQS Create profile InformationSee PackCreateProfileOut
<i>pProfileID</i>	<ul style="list-style-type: none">SLQS profile identifier information

<i>Tlvresult</i>	<ul style="list-style-type: none"> • TLV present flag, TRUE if presented, otherwise FALSE
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1013.2 Field Documentation

8.1013.2.1 `swi_uint256_t unpack_wds_SLQSCreateProfile_t::ParamPresenceMask`

8.1013.2.2 `PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut`

8.1013.2.3 `uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID`

8.1013.2.4 `uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult`

8.1014 unpack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- `uint16_t extendedErrorCode`
- `swi_uint256_t ParamPresenceMask`

8.1014.1 Detailed Description

This structure contains unpack delete profile information.

Parameters

<i>extendedErrorCode[OUT]</i>	<ul style="list-style-type: none"> • The extended error code received from DS Profile subsystem of type <code>eWDS_ERR_PROFILE_REG_xxx</code>. • Error code will only be present if error code <code>eQCWWAN_ERR_QMI_EXTENDED_INTERNAL</code> is returned by device. • See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description. • Bit to check in <code>ParamPresenceMask</code> - 224
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1014.2 Field Documentation

8.1014.2.1 `uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode`

8.1014.2.2 `swi_uint256_t unpack_wds_SLQSDeleteProfile_t::ParamPresenceMask`

8.1015 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

8.1015.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

Parameters

<i>CSTlv</i>	<ul style="list-style-type: none"> • Connection status tlv • See wds_ConnStatusTlv for more information • Bit to check in ParamPresenceMask - 16
<i>LMCERTlv</i>	<ul style="list-style-type: none"> • last modem call end reason tlv • see wds_LastMdmCallEndRsnTlv for more information • Bit to check in ParamPresenceMask - 17
<i>TXBOTlv</i>	<ul style="list-style-type: none"> • Tx bytes OK tlv • see wds_TXBytesOKTlv for more information • Bit to check in ParamPresenceMask - 18
<i>RXBOTlv</i>	<ul style="list-style-type: none"> • Rx bytes OK tlv • see wds_RXBytesOKTlv for more information • Bit to check in ParamPresenceMask - 19
<i>DSTlv</i>	<ul style="list-style-type: none"> • Dormancy status tlv • see wds_DormStatTlv for more information • Bit to check in ParamPresenceMask - 20
<i>DBTTlv</i>	<ul style="list-style-type: none"> • Data bear technology tlv • see wds_DataBearTechTlv for more information • Bit to check in ParamPresenceMask - 21
<i>CRTlv</i>	<ul style="list-style-type: none"> • Channel rate Tlv • see wds_ChannelRateTlv for more information • Bit to check in ParamPresenceMask - 22
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

Note

: None

8.1015.2 Field Documentation

8.1015.2.1 `wds_ChannelRateTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CRTlv`8.1015.2.2 `wds_ConnStatusTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CSTlv`8.1015.2.3 `wds_DataBearTechTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DBTTlv`8.1015.2.4 `wds_DormStatTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DSTlv`8.1015.2.5 `wds_LastMdmCallEndRsnTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::LMCERTlv`8.1015.2.6 `swi_uint256_t` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::ParamPresenceMask`8.1015.2.7 `wds_RXBytesOKTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::RXBOTlv`8.1015.2.8 `wds_TXBytesOKTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::TXBOTlv`8.1016 `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](#)

8.1016.1 Detailed Description

This structure contains unpack get 3GPP configure item information.

Parameters

	<i>LTEAttachProfile</i>	<ul style="list-style-type: none"> • LTE Attach Profile <ul style="list-style-type: none"> – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) • This setting is deprecated on MC/EM74xx • Bit to check in ParamPresenceMask - 16
--	-------------------------	--

	<i>profileList</i>	<p>Profile List</p> <ul style="list-style-type: none"> • an array of 4 profile configurations • Each element points to a single WORD value indicating profile • Optional parameter with possible values <ul style="list-style-type: none"> – 1 - 16 (MC/EM73xx and before) – 1 - 24 (MC/EM74xx and onwards) • function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present • Bit to check in ParamPresenceMask - 17 • Note: the 5th entry is currently ignored, please set it to zero
out	<i>defaultPDN-Enabled</i>	<ul style="list-style-type: none"> • 0 - disabled • 1 - enabled • Bit to check in ParamPresenceMask - 18
out	<i>_3gppRelease</i>	<p>3GPP release</p> <ul style="list-style-type: none"> • 0 - Release_99 • 1 - Release_5 • 2 - Release_6 • 3 - Release_7 • 4 - Release_8 • 5 - Release_9 (In 9x30 and towerads) • 6 - Release_10 (In 9x30 and towerads) • 7 - Release_11 (In 9x30 and towerads) • Bit to check in ParamPresenceMask - 19
out	<i>LTEAttach-ProfileList</i>	<ul style="list-style-type: none"> • pointer to WORD array indicating LTE Attach Profile List <ul style="list-style-type: none"> – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – Please provide attach profiles in order of decreasing priority in this list. • Bit to check in ParamPresenceMask - 20
in,out	<i>LTEAttach-ProfileListLen</i>	<ul style="list-style-type: none"> • Number of element in pLTEAttachProfileList <ul style="list-style-type: none"> – valid range: 1-24 – This setting is only supported for MC/EM74xx onwards • Bit to check in ParamPresenceMask - 20
	<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1016.2 Field Documentation

8.1016.2.1 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease

8.1016.2.2 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled

- 8.1016.2.3 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile`
- 8.1016.2.4 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]`
- 8.1016.2.5 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen`
- 8.1016.2.6 `swi_uint256_t unpack_wds_SLQSGet3GPPConfigItem_t::ParamPresenceMask`
- 8.1016.2.7 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]`

8.1017 `unpack_wds_SLQSGetCurrDataSystemStat_t` Struct Reference

Data Fields

- `uint8_t prefNetwork`
- `uint8_t networkInfoLen`
- `currNetworkInfo currNetworkInfo [255]`
- `swi_uint256_t ParamPresenceMask`

8.1017.1 Detailed Description

Data System Status

Parameters

<i>prefNetwork</i>	<ul style="list-style-type: none"> • Preferred Network • Values: <ul style="list-style-type: none"> – 0 - 3GPP – 1 - 3GPP2 • Bit to check in ParamPresenceMask - 16
<i>networkInfoLen</i>	<ul style="list-style-type: none"> • As input parameter size assigned to next parameter i.e. network information • As output the actual number of network information elements returned by the device • Bit to check in ParamPresenceMask - 16
<i>currNetworkInfo</i>	<ul style="list-style-type: none"> • Network information • See <code>currNetworkInfo</code> for more details • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1017.2 Field Documentation

- 8.1017.2.1 `currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]`
- 8.1017.2.2 `uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen`

8.1017.2.3 swi_uint256_t unpack_wds_SLQSGetCurrDataSystemStat_t::ParamPresenceMask

8.1017.2.4 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork

8.1018 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](#)

8.1018.1 Detailed Description

This structure contains unpack get current channel rate information.

Parameters

<i>current_channel_tx_rate</i>	<ul style="list-style-type: none"> • Current Channel Tx Rate. • Instantaneous channel Tx rate in bits per second. • In 9x15, this is the total current channel rate for all PDNs combined. • In 9x30 and later, this is the channel rate for a specific PDN. • Bit to check in ParamPresenceMask - 1
<i>current_channel_rx_rate</i>	<ul style="list-style-type: none"> • Current Channel Rx Rate. • Instantaneous channel Rx rate in bits per second. • In 9x15, this is the total current channel rate for all PDNs combined. • In 9x30 and later, this is the channel rate for a specific PDN • Bit to check in ParamPresenceMask - 1
<i>max_channel_tx_rate</i>	<ul style="list-style-type: none"> • Max Channel Tx Rate. • Maximum total Tx rate that modem is able to support in current serving system in bits per second. • In 9x15, this is a default hard coded value for the current serving system. • Bit to check in ParamPresenceMask - 1
<i>max_channel_rx_rate</i>	<ul style="list-style-type: none"> • Max Channel Rx Rate. • Maximum total Rx rate that modem is able to support in current serving system in bits per second. • In 9x15, this is a default hard coded value for the current serving system. • Bit to check in ParamPresenceMask - 1
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1018.2 Field Documentation

8.1018.2.1 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::current_channel_rx_rate`

8.1018.2.2 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::current_channel_tx_rate`

8.1018.2.3 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::max_channel_rx_rate`

8.1018.2.4 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::max_channel_tx_rate`

8.1018.2.5 `swi_uint256_t unpack_wds_SLQSGGetCurrentChannelRate_t::ParamPresenceMask`

8.1019 unpack_wds_SLQSGGetDataBearerTechnology_t Struct Reference

Data Fields

- `uint8_t dataBearerMask`
- `qmiWSDDataBearerTechnology curDataBearerTechnology`
- `qmiWSDDataBearerTechnology lastCallDataBearerTechnology`
- `swi_uint256_t ParamPresenceMask`

8.1019.1 Detailed Description

Structure to hold the data bearer technology values

Parameters

<i>dataBearerMask[OUT]</i>	<ul style="list-style-type: none"> • 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 liteQmiDataBearerMasks for bit-mask positions. • Bit to check in ParamPresenceMask - 1
<i>curDataBearerTechnology[OUT]</i>	<ul style="list-style-type: none"> • current data bearer technology value. • See qmiWSDDataBearerTechnology • Bit to check in ParamPresenceMask - 1
<i>lastCallDataBearerTechnology[OUT]</i>	<ul style="list-style-type: none"> • last call data bearer technology value. • See qmiWSDDataBearerTechnology <ul style="list-style-type: none"> – NULL if the parameter is not required • Bit to check in ParamPresenceMask - 16
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1019.2 Field Documentation

8.1019.2.1 `qmiWSDDataBearerTechnology unpack_wds_SLQSGGetDataBearerTechnology_t::curDataBearerTechnology`

8.1019.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

8.1019.2.3 qmiWSDDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology

8.1019.2.4 swi_uint256_t unpack_wds_SLQSGetDataBearerTechnology_t::ParamPresenceMask

8.1020 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

8.1020.1 Detailed Description

This structure contains the DUN Call Info response parameters

Parameters

connectionStatus	<ul style="list-style-type: none"> • See connectionStatus for more information • Bit to check in ParamPresenceMask - 16
callEndReason	<ul style="list-style-type: none"> • Last modem call end reason • See qaGobiApiTableCallEndReasons.h for Call End Reason • Only valid if the last call made was DUN, else zero is returned • Bit to check in ParamPresenceMask - 17
txOKBytesCount	<ul style="list-style-type: none"> • Number of bytes transmitted without error • Returned only if a data call is up • Bit to check in ParamPresenceMask - 18
rxOKBytesCount	<ul style="list-style-type: none"> • Number of bytes received without error • Returned only if a data call is up • Bit to check in ParamPresenceMask - 19

<i>dormancyStatus</i>	<ul style="list-style-type: none"> • Current traffic channel status • Returned if a data call is up <ul style="list-style-type: none"> – 0x01 - Traffic channel dormant – 0x02 - Traffic channel active • Bit to check in ParamPresenceMask - 20
<i>dataBearerTech</i>	<ul style="list-style-type: none"> • Current data bearer technology • Returned only if a data call is up <ul style="list-style-type: none"> – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO RevA) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA • Bit to check in ParamPresenceMask - 21
<i>channelRate</i>	<ul style="list-style-type: none"> • See dunchannelRate for more information • Bit to check in ParamPresenceMask - 22
<i>lastCallTXOK-BytesCnt</i>	<ul style="list-style-type: none"> • Number of bytes transmitted without error during the last data call (0 if no call was made). • Return only if not in a call and the previous call was made using DUN. • Bit to check in ParamPresenceMask - 23
<i>lastCallRXOK-BytesCnt</i>	<ul style="list-style-type: none"> • Number of bytes transmitted without error during the last data call (0 if no call was made). • Return only if not in a call and the previous call was made using DUN. • Bit to check in ParamPresenceMask - 24

<i>mdmCall-DurationActive</i>	<ul style="list-style-type: none"> • Duration that the call is active in milliseconds • If the modem connection status is connected, this represents the active duration of the current DUN call • 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) • Bit to check in ParamPresenceMask - 25
<i>lastCallData-BearerTech</i>	<ul style="list-style-type: none"> • Last Call Data Bearer Technology • Returned only if not in a call and when the previous call was made using DUN <ul style="list-style-type: none"> – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO Rev A) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA • Bit to check in ParamPresenceMask - 32
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1020.2 Field Documentation

8.1020.2.1 `uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason`

8.1020.2.2 `dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate`

8.1020.2.3 `connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus`

8.1020.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`

8.1020.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`

8.1020.2.6 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech`

8.1020.2.7 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt`

- 8.1020.2.8 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt`
- 8.1020.2.9 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive`
- 8.1020.2.10 `swi_uint256_t unpack_wds_SLQSGetDUNCallInfo_t::ParamPresenceMask`
- 8.1020.2.11 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount`
- 8.1020.2.12 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount`

8.1021 `unpack_wds_SLQSGetProfileSettings_t` Struct Reference

Data Fields

- [UnPackGetProfileSettingOut](#) * [pProfileSettings](#)
- `uint8_t` [ProfileType](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.1021.1 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"> • 0 - Profile 3GPP • 1 - Profile 3GPP2
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1021.2 Field Documentation

- 8.1021.2.1 `swi_uint256_t unpack_wds_SLQSGetProfileSettings_t::ParamPresenceMask`
- 8.1021.2.2 `UnPackGetProfileSettingOut`* `unpack_wds_SLQSGetProfileSettings_t::pProfileSettings`
- 8.1021.2.3 `uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType`
- 8.1021.2.4 `uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult`

8.1022 `unpack_wds_SLQSGetProfileSettingsV2_t` Struct Reference

Data Fields

- [UnPackGetProfileSettingOutV2](#) * [pProfileSettings](#)
- `uint8_t` [ProfileType](#)
- `uint16_t` [Tlvresult](#)

- [swi_uint256_t ParamPresenceMask](#)

8.1022.1 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"> • 0 - Profile 3GPP • 1 - Profile 3GPP2
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1022.2 Field Documentation

8.1022.2.1 [swi_uint256_t](#) unpack_wds_SLQSGetProfileSettingsV2_t::ParamPresenceMask

8.1022.2.2 [UnPackGetProfileSettingOutV2*](#) unpack_wds_SLQSGetProfileSettingsV2_t::pProfileSettings

8.1022.2.3 [uint8_t](#) unpack_wds_SLQSGetProfileSettingsV2_t::ProfileType

8.1022.2.4 [uint16_t](#) unpack_wds_SLQSGetProfileSettingsV2_t::Tlvresult

8.1023 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](#)

8.1023.1 Detailed Description

This structure contains unpack get runtime settings information.

Parameters

<i>IPv4</i>	ipv4 address <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 30
<i>ProfileName</i>	profile name <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 16
<i>PDPTtype</i>	PDP type <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 17
<i>APNName</i>	APN name <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 20
<i>PrimaryDNSV4</i>	primary dns IPV4 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 21
<i>SecondaryDNS-V4</i>	secondary dns IPV4 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 22
<i>UMTSGranted-QoS</i>	UMTS Granted QoS <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 23
<i>GPRSGranted-QoS</i>	GPRS Granted QoS <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 25
<i>Username</i>	username for authentication process <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 27
<i>Authentication</i>	authentication for authentication process <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 29
<i>ProfielID</i>	profile ID <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 31
<i>GWAddressV4</i>	Gateway IPv4 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 32
<i>SubnetMaskV4</i>	Subnet mask IPV4 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 33
<i>PCSCFAddrPC-O</i>	PCSCF address PCO <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 34

<i>ServerAddrList</i>	PCSCF server address list IPV4 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 35
<i>PCSCFFQDN-AddrList</i>	PCSCF FQDN address list IPV4 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 36
<i>PrimaryDNSV6</i>	Primary DNS IPV6 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 39
<i>SecondaryDNS-V6</i>	Secondary DNS IPV6 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 40
<i>Mtu</i>	actual (runtime) Maximum Transfer Unit <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 41
<i>DomainList</i>	domain list <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 42
<i>IPFamily-Preference</i>	ip family preference, it could be IPV4 or IPV6 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 43
<i>IMCNflag</i>	IM control flag, value: TRUE or FALSE <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 44
<i>Technology</i>	technology on which current packet data session is in progress <ul style="list-style-type: none"> • Values: <ul style="list-style-type: none"> – 32767 - CDMA – 32764 - UMTS – 30592 - EPC – 30584 - modem link local • Bit to check in ParamPresenceMask - 45
<i>IPV6AddrInfo</i>	address information IPV6 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 37
<i>IPV6GWAddr-Info</i>	gateway address information IPV6 <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 38
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1023.2 Field Documentation

8.1023.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

8.1023.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

8.1023.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.1023.2.4 struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

- 8.1023.2.5 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4`
- 8.1023.2.6 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag`
- 8.1023.2.7 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference`
- 8.1023.2.8 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4`
- 8.1023.2.9 `struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6AddrInfo`
- 8.1023.2.10 `struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6GWAddrInfo`
- 8.1023.2.11 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu`
- 8.1023.2.12 `swi_uint256_t unpack_wds_SLQSGetRuntimeSettings_t::ParamPresenceMask`
- 8.1023.2.13 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO`
- 8.1023.2.14 `struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList`
- 8.1023.2.15 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPTType`
- 8.1023.2.16 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4`
- 8.1023.2.17 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]`
- 8.1023.2.18 `struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID`
- 8.1023.2.19 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]`
- 8.1023.2.20 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4`
- 8.1023.2.21 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]`
- 8.1023.2.22 `struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList`
- 8.1023.2.23 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4`
- 8.1023.2.24 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology`
- 8.1023.2.25 `LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS`
- 8.1023.2.26 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]`

8.1024 `unpack_wds_SLQSModifyProfile_t` Struct Reference

Data Fields

- `uint16_t * pExtErrorCode`
- `swi_uint256_t ParamPresenceMask`

8.1024.1 Detailed Description

This structure contains out parameters for `unpack_wds_SLQSModifyProfile`

Parameters

<i>pExtErrorCode</i>	<ul style="list-style-type: none"> The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_REG_XXX. Error code will only be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description. Bit to check in ParamPresenceMask - 224
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1024.2 Field Documentation

8.1024.2.1 `swi_uint256_t` unpack_wds_SLQSModifyProfile_t::ParamPresenceMask8.1024.2.2 `uint16_t*` unpack_wds_SLQSModifyProfile_t::pExtErrorCode

8.1025 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.1025.1 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"> bitmask representation to indicate valid parameters.

8.1025.2 Field Documentation

8.1025.2.1 `swi_uint256_t` unpack_wds_SLQSSetIPFamilyPreference_t::ParamPresenceMask8.1025.2.2 `uint16_t` unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

8.1026 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](#)

8.1026.1 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"> • Current link status. Values: <ul style="list-style-type: none"> – 1 - DISCONNECTED – 2 - CONNECTED – 3 - SUSPENDED – 4 - AUTHENTICATING • Bit to check in ParamPresenceMask - 1
<i>reconfigReqd</i>	Indicates whether the network interface on the host needs to be reconfigured. <ul style="list-style-type: none"> • Values: <ul style="list-style-type: none"> – 0 - No need to reconfigure. – 1 - Reconfiguration required.

- Bit to check in ParamPresenceMask - **1**

Parameters

<i>sessionEnd-Reason</i>	Call End Reason <ul style="list-style-type: none"> • See qaGobiApiTableCallEndReasons.h for Call End Reason • Bit to check in ParamPresenceMask - 16
<i>verboseSessn-EndReasonType</i>	Verbose call end reason type <ul style="list-style-type: none"> • Call end reason type. Values: <ul style="list-style-type: none"> – 0 - Unspecified – 1 - Mobile IP – 2 - Internal – 3 - Call Manager defined – 6 - 3GPP Specification defined – 7 - PPP – 8 - EHRPD – 9 - IPv6 • Bit to check in ParamPresenceMask - 17
<i>verboseSessn-EndReason</i>	Reason the call ended (verbose) <ul style="list-style-type: none"> • See qaGobiApiTableCallEndReasons.h for Call End Reason • Bit to check in ParamPresenceMask - 17

<i>ipFamily</i>	<p>IP family of the packet data connection.</p> <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> 4 - IPv4 6 - IPv6 Bit to check in ParamPresenceMask - 18
<i>techName</i>	<p>Technology name of the packet data connection.</p> <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> 32767 - CDMA 32764 - UMTS 30592 - EPC 30590 - EMBMS 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. Bit to check in ParamPresenceMask - 19
<i>bearerID</i>	<ul style="list-style-type: none"> bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection. Valid Values - 0 to 16 <ul style="list-style-type: none"> Bit to check in ParamPresenceMask - 20
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1026.2 Field Documentation

8.1026.2.1 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID`

8.1026.2.2 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status`

8.1026.2.3 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily`

8.1026.2.4 `swi_uint256_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ParamPresenceMask`

8.1026.2.5 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd`

8.1026.2.6 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason`

8.1026.2.7 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName`

8.1026.2.8 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason`

8.1026.2.9 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType`

8.1027 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

8.1027.1 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"> • Number of bytes transmitted without error • Bit to check in ParamPresenceMask - 25
<i>rx_bytes</i>	<ul style="list-style-type: none"> • Number of bytes received without error • Bit to check in ParamPresenceMask - 26
<i>tx_pkts</i>	<ul style="list-style-type: none"> • Number of packets transmitted without error • Bit to check in ParamPresenceMask - 16
<i>rx_pkts</i>	<ul style="list-style-type: none"> • Number of packets received without error. • Bit to check in ParamPresenceMask - 17
<i>mipstatAvail</i>	Mobile IP status available <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 27
<i>mipStatus</i>	<ul style="list-style-type: none"> • Status of the last MIP call (or attempt). • Values <ul style="list-style-type: none"> – 0x00 - Success • Bit to check in ParamPresenceMask - 27

<i>dBTechAvail</i>	Data Bearer technology available <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 23
<i>dBTechnology</i>	<ul style="list-style-type: none"> • Data Bearer technology <ul style="list-style-type: none"> – Values <ul style="list-style-type: none"> * 0x01 - cdma2000 ® 1X * 0x02 - cdma2000 ® HRPD (1xEV-DO) * 0x03 - GSM * 0x04 - UMTS * 0x05 - cdma2000 ® HRPD (1xEV-DO RevA) * 0x06 - EDGE * 0x07 - HSDPA and WCDMA * 0x08 - WCDMA and HSUPA * 0x09 - HSDPA and HSUPA * 0x0A - LTE * 0x0B - cdma2000 ® EHRPD * 0x0C - HSDPA+ and WCDMA * 0x0D - HSDPA+ and HSUPA * 0x0E - DC_HSDPA+ and WCDMA * 0x0F - DC_HSDPA+ and HSUPA * 0x10 - HSDPA+ and 64QAM * 0x11 - HSDPA+, 64QAM and HSUPA * 0x12 - TDSCDMA * 0x13 - TDSCDMA and HSDPA * 0x14 - TDSCDMA and HSUPA * -1 - Unknown • Bit to check in ParamPresenceMask - 23
<i>dormancyStat-Avail</i>	Dormancy status available <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 24
<i>dormancyStatus</i>	<ul style="list-style-type: none"> • Dormancy status • Values <ul style="list-style-type: none"> – 1 - Traffic channel dormant – 2 - Traffic channel active • Bit to check in ParamPresenceMask - 24
<i>currDBTechAvail</i>	<ul style="list-style-type: none"> • Current Data Bearer technology available • Bit to check in ParamPresenceMask - 29

<i>ratMask</i>	<ul style="list-style-type: none"> • RAT mask to indicate type of technology • Values <ul style="list-style-type: none"> – 0x00 - DONT_CARE – 0x8000 - NULL_BEARER • CDMA RAT mask <ul style="list-style-type: none"> – 0x01 - CDMA_1X – 0x02 - EVDO_REV0 – 0x05 - HRPD – 0x0B - EHRPD • UMTS RAT mask <ul style="list-style-type: none"> – 0x03 - GPRS – 0x04 - WCDMA – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA • Bit to check in ParamPresenceMask - 29
<i>soMask</i>	<ul style="list-style-type: none"> • SO Mask • Bit to check in ParamPresenceMask - 29
<i>dataSysStatAvail</i>	<ul style="list-style-type: none"> • Data System Status available • Bit to check in ParamPresenceMask - 36
<i>prefNetwork</i>	<ul style="list-style-type: none"> • preferred network • Values <ul style="list-style-type: none"> – 0 - 3GPP – 1 - 3GPP2
<i>currNWInfo</i>	<ul style="list-style-type: none"> • Current Network Info • see wds_currNetworkInfo for more info
<i>dBtechExtAvail</i>	<p>Data bearer technology extended available</p> <ul style="list-style-type: none"> • Bit to check in ParamPresenceMask - 42

<i>dBtechnologyExt</i>	<ul style="list-style-type: none"> • Data bearer technology Extended. • Values <ul style="list-style-type: none"> – WDS_BEARER_TECH_NETWORK_3GPP (0) - 3GPP – WDS_BEARER_TECH_NETWORK_3GPP2 (1) - 3GPP2 • Bit to check in ParamPresenceMask - 42
<i>dBTechExtRat-Value</i>	<ul style="list-style-type: none"> • Data bearer technology Extended RAT Value • Values <ul style="list-style-type: none"> – WDS_BEARER_TECH_RAT_EX_NULL_BEARER (0x00) - NULL bearer – WDS_BEARER_TECH_RAT_EX_3GPP_WCDMA (0x01) - 3GPP WCDMA – WDS_BEARER_TECH_RAT_EX_3GPP_GERAN (0x02) - 3GPP GERAN – WDS_BEARER_TECH_RAT_EX_3GPP_LTE (0x03) - 3GPP LTE – WDS_BEARER_TECH_RAT_EX_3GPP_TDSCDMA (0x04) - 3GPP TDSCDMA – WDS_BEARER_TECH_RAT_EX_3GPP_WLAN (0x05) - 3GPP WLAN – WDS_BEARER_TECH_RAT_EX_3GPP_MAX (0x64) - 3GPP maximum – WDS_BEARER_TECH_RAT_EX_3GPP2_1X (0x65) - 3GPP2 1X – WDS_BEARER_TECH_RAT_EX_3GPP2_HRPD (0x66) - 3GPP2 HRPD – WDS_BEARER_TECH_RAT_EX_3GPP2_EHRPD (0x67) - 3GPP2 EHRPD – WDS_BEARER_TECH_RAT_EX_3GPP2_WLAN (0x68) - 3GPP2 WLAN – WDS_BEARER_TECH_RAT_EX_3GPP2_MAX (0xC8) - 3GPP2 maximum • Bit to check in ParamPresenceMask - 42

<i>dBTechExtSo-Mask</i>	<ul style="list-style-type: none"> • Data bearer technology Extended SO Mask • 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. • Values <ul style="list-style-type: none"> – 0x00 - SO mask unspecified – 3GPP SO mask <ul style="list-style-type: none"> * 0x01 - WCDMA * 0x02 - HSDPA * 0x04 - HSUPA * 0x08 - HSDPAPLUS * 0x10 - DC HSDPAPLUS * 0x20 - 64 QAM * 0x40 - HSPA * 0x80 - GPRS * 0x100 - EDGE * 0x200 - GSM * 0x400 - S2B * 0x800 - LTE limited service * 0x1000 - LTE FDD * 0x2000 - LTE TDD – 3GPP2 SO mask <ul style="list-style-type: none"> * 0x01000000 - 1X IS95 * 0x02000000 - 1X IS2000 * 0x04000000 - 1X IS2000 REL A * 0x08000000 - HDR REV0 DPA * 0x10000000 - HDR REVA DPA * 0x20000000 - HDR REVB DPA * 0x40000000 - HDR REVA MPA * 0x80000000 - HDR REVB MPA * 0x100000000 - HDR REVA EMPA * 0x200000000 - HDR REVB EMPA * 0x400000000 - HDR REVB MMPA * 0x800000000 - HDR EVDO FMC • Bit to check in ParamPresenceMask - 42
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1027.2 Field Documentation

8.1027.2.1 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail`

8.1027.2.2 `wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]`

8.1027.2.3 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail`

8.1027.2.4 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail`

- 8.1027.2.5 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechExtAvail
- 8.1027.2.6 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtRatValue
- 8.1027.2.7 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtSoMask
- 8.1027.2.8 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::DBTechnology
- 8.1027.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechnologyExt
- 8.1027.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail
- 8.1027.2.11 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus
- 8.1027.2.12 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail
- 8.1027.2.13 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus
- 8.1027.2.14 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen
- 8.1027.2.15 swi_uint256_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ParamPresenceMask
- 8.1027.2.16 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork
- 8.1027.2.17 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask
- 8.1027.2.18 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes
- 8.1027.2.19 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts
- 8.1027.2.20 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask
- 8.1027.2.21 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes
- 8.1027.2.22 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts
- 8.1027.2.23 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

8.1028 unpack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

- [wdsDhcpv4HwConfig](#) * [pHwConfig](#)
- [wdsDhcpv4OptionList](#) * [pRequestOptionList](#)
- [swi_uint256_t](#) [ParamPresenceMask](#)

8.1028.1 Detailed Description

WDS SWI DHCPv4 Config Structure

Parameters

<i>pHwConfig</i>	<ul style="list-style-type: none"> • pointer to HW Config structure • See wdsDhcpv4HwConfig • Bit to check in ParamPresenceMask - 16
<i>pRequestOption-List</i>	<ul style="list-style-type: none"> • pointer to Option List structure to be sent in DHCP request • See wdsDhcpv4OptionList • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1028.2 Field Documentation

8.1028.2.1 `swi_uint256_t unpack_wds_SLQSSGetDHCPv4ClientConfig_t::ParamPresenceMask`

8.1028.2.2 `wdsDhcpv4HwConfig* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig`

8.1028.2.3 `wdsDhcpv4OptionList* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList`

8.1029 `unpack_wds_SLQSSGetLoopback_t` Struct Reference

Data Fields

- `uint8_t` [ByteLoopbackMode](#)
- `uint8_t` [ByteLoopbackMultiplier](#)
- `swi_uint256_t` [ParamPresenceMask](#)

8.1029.1 Detailed Description

This structure contains unpack Get loopback information.

Parameters

<i>ByteLoopback-Mode</i>	<ul style="list-style-type: none"> • Loopback Mode. <ul style="list-style-type: none"> – 0 - Disable – 1 - Enable • Bit to check in ParamPresenceMask - 16
<i>ByteLoopback-Multiplier</i>	<ul style="list-style-type: none"> • Loopback multiplier. Number of downlink bytes to send for each uplink byte. • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1029.2 Field Documentation

8.1029.2.1 `uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode`

8.1029.2.2 `uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier`

8.1029.2.3 `swi_uint256_t unpack_wds_SLQSSGetLoopback_t::ParamPresenceMask`

8.1030 unpack_wds_SLQSSStartDataSession_t Struct Reference

Data Fields

- `uint32_t * psid`
- `uint32_t * pFailureReason`
- `uint32_t * pVerboseFailReasonType`
- `uint32_t * pVerboseFailureReason`
- `swi_uint256_t ParamPresenceMask`

8.1030.1 Detailed Description

This structure contains unpack Start Data Session Information.

Parameters

<i>psid</i>	<ul style="list-style-type: none"> • Assigned session ID when starting a data session • Bit to check in ParamPresenceMask - 1
<i>pFailureReason</i>	<ul style="list-style-type: none"> • Reason data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason • Bit to check in ParamPresenceMask - 16
<i>pVerboseFail-ReasonType</i>	<ul style="list-style-type: none"> • Parameter describing type of verbose failure reason • See qaGobiApiTableCallEndReasons.h for Call End Reason Type • Bit to check in ParamPresenceMask - 17
<i>pVerboseFailure-Reason</i>	<ul style="list-style-type: none"> • Verbose reason explaining why call failed. Depends on verbFailReasonType parameter • See qaGobiApiTableCallEndReasons.h for Call End Reason • Bit to check in ParamPresenceMask - 17
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> • bitmask representation to indicate valid parameters.

8.1030.2 Field Documentation

8.1030.2.1 `swi_uint256_t unpack_wds_SLQSSStartDataSession_t::ParamPresenceMask`

8.1030.2.2 `uint32_t* unpack_wds_SLQSSStartDataSession_t::pFailureReason`

8.1030.2.3 uint32_t* unpack_wds_SLQSSStartDataSession_t::psid

8.1030.2.4 uint32_t* unpack_wds_SLQSSStartDataSession_t::pVerboseFailReasonType

8.1030.2.5 uint32_t* unpack_wds_SLQSSStartDataSession_t::pVerboseFailureReason

8.1031 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

8.1031.1 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"> Context Identifier Bit to check in ParamPresenceMask - 16
<i>bearerId</i>	(optional) <ul style="list-style-type: none"> Bearer Identity 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. Bit to check in ParamPresenceMask - 17
<i>apnName</i>	(optional) <ul style="list-style-type: none"> APN name associated with the context id Bit to check in ParamPresenceMask - 18
<i>ipv4Address</i>	(optional) <ul style="list-style-type: none"> IPv4 Address Bit to check in ParamPresenceMask - 19

<i>ipv4GWAddress</i>	(optional) <ul style="list-style-type: none"> IPv4 Gateway Address Bit to check in ParamPresenceMask - 20
<i>prDNSIPv4-Address</i>	(optional) <ul style="list-style-type: none"> Primary DNS IPv4 Address Bit to check in ParamPresenceMask - 21
<i>seDNSIPv4-Address</i>	(optional) <ul style="list-style-type: none"> Secondary DNS IPv4 Address Bit to check in ParamPresenceMask - 22
<i>ipv6Address</i>	(optional) <ul style="list-style-type: none"> IPv6 Address Bit to check in ParamPresenceMask - 23
<i>ipv6GWAddress</i>	(optional) <ul style="list-style-type: none"> IPv6 Gateway Address See ipv6AddressInfo for more information Bit to check in ParamPresenceMask - 24
<i>prDNSIPv6-Address</i>	(optional) <ul style="list-style-type: none"> Primary IPv6 DNS Address(in network byte order) This is an 8-element array of 16-bit numbers, each of which is in big-endian format Bit to check in ParamPresenceMask - 25
<i>seDNSIPv6-Address</i>	(optional) <ul style="list-style-type: none"> Secondary IPv6 DNS Address(in network byte order) This is an 8-element array of 16-bit numbers, each of which is in big-endian format Bit to check in ParamPresenceMask - 26
<i>prPCSCFIPv4-Address</i>	(optional) <ul style="list-style-type: none"> Primary PCSCF IPv4 Address Bit to check in ParamPresenceMask - 27
<i>sePCSCFIPv4-Address</i>	(optional) <ul style="list-style-type: none"> Secondary PCSCF IPv4 Address Bit to check in ParamPresenceMask - 28
<i>prPCSCFIPv6-Address</i>	(optional) <ul style="list-style-type: none"> Primary PCSCF IPv6 Address This is an 8-element array of 16-bit numbers, each of which is in big-endian format Bit to check in ParamPresenceMask - 29
<i>sePCSCFIPv6-Address</i>	(optional) <ul style="list-style-type: none"> Secondary PCSCF IPv6 Address This is an 8-element array of 16-bit numbers, each of which is in big-endian format Bit to check in ParamPresenceMask - 30
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> bitmask representation to indicate valid parameters.

8.1031.2 Field Documentation

- 8.1031.2.1 `int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]`
- 8.1031.2.2 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId`
- 8.1031.2.3 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`
- 8.1031.2.4 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address`
- 8.1031.2.5 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress`
- 8.1031.2.6 `ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address`
- 8.1031.2.7 `ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress`
- 8.1031.2.8 `swi_uint256_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ParamPresenceMask`
- 8.1031.2.9 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address`
- 8.1031.2.10 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`
- 8.1031.2.11 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`
- 8.1031.2.12 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`
- 8.1031.2.13 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`
- 8.1031.2.14 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`
- 8.1031.2.15 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`
- 8.1031.2.16 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

8.1032 UnPackGetProfileSettingOut Struct Reference

Data Fields

- [UnpackQmiProfileInfo curProfile](#)
- `uint16_t * pExtErrCode`

8.1032.1 Detailed Description

This structure contains the profile settings output

Parameters

<i>curProfile</i>	<ul style="list-style-type: none"> • Structure containing details of the current profile • See UnpackQmiProfileInfo
-------------------	---

<i>pExtErrCode</i>	<ul style="list-style-type: none"> • pointer to a 2 byte extended error code • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. • See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description. • Bit to check in ParamPresenceMask - 224
--------------------	--

8.1032.2 Field Documentation

8.1032.2.1 **UnpackQmiProfileInfo** UnPackGetProfileSettingOut::curProfile

8.1032.2.2 **uint16_t*** UnPackGetProfileSettingOut::pExtErrCode

8.1033 UnPackGetProfileSettingOutV2 Struct Reference

Data Fields

- [UnpackQmiProfileInfoV2](#) curProfile
- **uint16_t *** [pExtErrCode](#)

8.1033.1 Detailed Description

This structure contains the profile settings output

Parameters

<i>curProfile</i>	<ul style="list-style-type: none"> • Structure containing details of the current profile • See UnpackQmiProfileInfoV2
<i>pExtErrCode</i>	<ul style="list-style-type: none"> • pointer to a 2 byte extended error code • Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device. • See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description. • Bit to check in ParamPresenceMask - 224

8.1033.2 Field Documentation

8.1033.2.1 **UnpackQmiProfileInfoV2** UnPackGetProfileSettingOutV2::curProfile

8.1033.2.2 **uint16_t*** UnPackGetProfileSettingOutV2::pExtErrCode

8.1034 UnpackSviAvmsEventReportBinaryUpdateSessionInfo 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 [bSerity](#)
- uint16_t [wVersionLength](#)
- uint8_t [szVersion](#) [128]
- uint16_t [wNameLength](#)
- uint8_t [szName](#) [128]
- uint16_t [wDescriptionLength](#)
- uint8_t [szDescription](#) [1024]
- uint8_t [TlvPresent](#)

8.1034.1 Detailed Description

This structure contains unpack event report binary update session information parameters.

Parameters

<i>bBinaryType</i>	: Type <ul style="list-style-type: none"> • 1 - Firmware. • 2 - User App. • 3 - Legato Framework.
<i>bStat</i>	: State <ul style="list-style-type: none"> • 0x01 - No binary update available. • 0x02 - Query binary Download. • 0x03 - Binary Downloading. • 0x04 - Binary downloaded. • 0x05 -Query Binary Update. • 0x06 - Binary updating. • 0x07 - Binary updated.
<i>bUserInput-Request</i>	Bit mask of available user inputs. <ul style="list-style-type: none"> • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 0x02 - Reject.
<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>bSerity</i>	Serity. <ul style="list-style-type: none"> • 0x01 - Mandatory. • 0x02 - Optional.
<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

8.1034.2 Field Documentation

- 8.1034.2.1 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bBinaryType`
- 8.1034.2.2 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bSerity`
- 8.1034.2.3 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bState`
- 8.1034.2.4 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bUserInputRequest`
- 8.1034.2.5 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szDescription[1024]`
- 8.1034.2.6 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szName[128]`
- 8.1034.2.7 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szVersion[128]`
- 8.1034.2.8 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::TlvPresent`
- 8.1034.2.9 `uint32_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::ulPkgDownloadComplete`
- 8.1034.2.10 `uint32_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::ulPkgDownloadSize`
- 8.1034.2.11 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wDescriptionLength`
- 8.1034.2.12 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wNameLength`
- 8.1034.2.13 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wUpdateCompeteStatus`
- 8.1034.2.14 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wUserInputTimeout`
- 8.1034.2.15 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wVersionLength`

8.1035 UnpackSwiAvmsEventReportConfig Struct Reference

Data Fields

- `uint8_t bState`
- `uint8_t bUserInputRequest`
- `uint16_t wUserInputTimeout`
- `uint16_t wAlertMsgLength`
- `uint8_t szAlertMsg [200]`
- `uint8_t TlvPresent`

8.1035.1 Detailed Description

This structure contains unpack event report configure parameters.

Parameters

<i>bState</i>	State. <ul style="list-style-type: none"> • 0x01 - AVMS Read Request. • 0x02 - AVMS Change Request. • 0x03 - AVMS Config Complete.
<i>bUserInput-Request</i>	Bit mask of available user inputs. <ul style="list-style-type: none"> • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 0x02 - Reject.
<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

8.1035.2 Field Documentation

- 8.1035.2.1 `uint8_t UnpackSwiAvmsEventReportConfig::bState`
- 8.1035.2.2 `uint8_t UnpackSwiAvmsEventReportConfig::bUserInputRequest`
- 8.1035.2.3 `uint8_t UnpackSwiAvmsEventReportConfig::szAlertMsg[200]`
- 8.1035.2.4 `uint8_t UnpackSwiAvmsEventReportConfig::TlvPresent`
- 8.1035.2.5 `uint16_t UnpackSwiAvmsEventReportConfig::wAlertMsgLength`
- 8.1035.2.6 `uint16_t UnpackSwiAvmsEventReportConfig::wUserInputTimeout`

8.1036 UnpackSwiAvmsEventReportConnectionRequest Struct Reference

Data Fields

- `uint8_t bUserInputRequest`
- `uint16_t wUserInputTimeout`
- `uint8_t TlvPresent`

8.1036.1 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"> • 0x00 - No user input required. Informational indication. • 0x01 - Accept. • 0x02 - Reject.
<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

8.1036.2 Field Documentation

8.1036.2.1 uint8_t UnpackSwiAvmsEventReportConnectionRequest::bUserInputRequest

8.1036.2.2 uint8_t UnpackSwiAvmsEventReportConnectionRequest::TlvPresent

8.1036.2.3 uint16_t UnpackSwiAvmsEventReportConnectionRequest::wUserInputTimeout

8.1037 UnpackSwiAvmsEventReportDataSessionStatus Struct Reference

Data Fields

- uint8_t [bType](#)
- uint16_t [wErrorCode](#)
- uint8_t [TlvPresent](#)

8.1037.1 Detailed Description

This structure contains unpack evnet report data session status parameters.

Parameters

<i>bType</i>	Notification type. <ul style="list-style-type: none"> • 0: Data session closed. • 1: Data session activated. • 2: Register made. -3: Data session error.
<i>wErrorCode</i>	LWM2M Session error code. <ul style="list-style-type: none"> • 0x0000: none
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

8.1037.2 Field Documentation

8.1037.2.1 uint8_t UnpackSwiAvmsEventReportDataSessionStatus::bType

8.1037.2.2 uint8_t UnpackSwiAvmsEventReportDataSessionStatus::TlvPresent

8.1037.2.3 uint16_t UnpackSwiAvmsEventReportDataSessionStatus::wErrorCode

8.1038 UnpackSwiAvmsEventReportHTTPStatus Struct Reference

Data Fields

- uint16_t [wHTTPStatus](#)
- uint8_t [TlvPresent](#)

8.1038.1 Detailed Description

This structure contains unpack event report HTTP status parameters.

Parameters

<i>wHTTPStatus</i>	<p>See RFC 7231.</p> <ul style="list-style-type: none"> • 100 - Continue • 101 - Switching Protocols • 200 - OK • 201 - Created • 202 - Accepted • 203 - Non-Authoritative Information • 204 - No Content • 205 - Reset Content • 206 - Partial Content • 300 - Multiple Choices • 301 - Moved Permanently • 302 - Found • 303 - See Other • 304 - Not Modified • 305 - Use Proxy • 307 - Temporary Redirect • 400 - Bad Request • 401 - Unauthorized • 402 - Payment Required • 403 - Forbidden • 404 - Not Found • 405 - Method Not Allowed • 406 - Not Acceptable • 407 - Proxy Authentication Required • 408 - Request Timeout • 409 - Conflict • 410 - Gone • 411 - Length Required • 412 - Precondition Failed • 413 - Payload Too Large • 414 - URI Too Long • 415 - Unsupported Media Type • 416 - Range Not Satisfiable • 417 - Expectation Failed • 426 - Upgrade Required • 500 - Internal Server Error • 501 - Not Implemented • 502 - Bad Gateway • 503 - Service Unavailable • 504 - Gateway Timeout • 505 - HTTP Version Not Supported
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

8.1038.2 Field Documentation

8.1038.2.1 uint8_t UnpackSwiAvmsEventReportHTTPStatus::TlvPresent

8.1038.2.2 uint16_t UnpackSwiAvmsEventReportHTTPStatus::wHTTPStatus

8.1039 UnpackSwiAvmsEventReportNotification Struct Reference

Data Fields

- uint8_t [bNotification](#)
- uint16_t [wSessionStatus](#)
- uint8_t [TlvPresent](#)

8.1039.1 Detailed Description

This structure contains unpack event report notification parameters.

Parameters

<i>bNotification</i>	Notification. <ul style="list-style-type: none"> • 0x14 - Module starts sending data to server. • 0x15 - Authentication with the server. • 0x16 - session with the server is ended.
<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"> • 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
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

8.1039.2 Field Documentation

8.1039.2.1 uint8_t UnpackSviAvmsEventReportNotification::bNotification

8.1039.2.2 uint8_t UnpackSviAvmsEventReportNotification::TlvPresent

8.1039.2.3 uint16_t UnpackSviAvmsEventReportNotification::wSessionStatus

8.1040 UnpackSviAvmsEventReportPackageID Struct Reference

Data Fields

- uint8_t [bPackageID](#)
- uint8_t [TlvPresent](#)

8.1040.1 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

8.1040.2 Field Documentation

8.1040.2.1 uint8_t UnpackSviAvmsEventReportPackageID::bPackageID

8.1040.2.2 uint8_t UnpackSviAvmsEventReportPackageID::TlvPresent

8.1041 UnpackSviAvmsEventReportRegStatus Struct Reference

Data Fields

- uint8_t [bRegStatus](#)
- uint8_t [TlvPresent](#)

8.1041.1 Detailed Description

This structure contains unpack event report registration status parameters.

Parameters

<i>bRegStatus</i>	LWM2M Registration status. <ul style="list-style-type: none">• 0: Need Bootstrap.• 1: Bootstrap made.• 2: Register made. -3: Update made.
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

8.1041.2 Field Documentation

8.1041.2.1 uint8_t UnpackSviAvmsEventReportRegStatus::bRegStatus

8.1041.2.2 `uint8_t UnpackSwiAvmsEventReportRegStatus::TlvPresent`

8.1042 UnpackSwiAvmsEventReportSessionType Struct Reference

Data Fields

- `uint8_t bType`
- `uint8_t TlvPresent`

8.1042.1 Detailed Description

This structure contains unpack event report session type parameters.

Parameters

<i>bType</i>	Session Type. <ul style="list-style-type: none">• 0: Bootstrap session.• 1. DM session
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

8.1042.2 Field Documentation

8.1042.2.1 `uint8_t UnpackSwiAvmsEventReportSessionType::bType`

8.1042.2.2 `uint8_t UnpackSwiAvmsEventReportSessionType::TlvPresent`

8.1043 UnpackSwiAvmsEventReportWAMSParamChange Struct Reference

Data Fields

- `uint16_t wWamsChangeMask`
- `uint8_t TlvPresent`

8.1043.1 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">• By default set to 0xFF for all changes.• 0x01 - device_login• 0x02 - device_MD5_key• 0x04 - server_login• 0x08 - server_MD5_key• 0x10 - server_URL• 0x20 - Nonce• 0x40 - Application key
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

8.1043.2 Field Documentation

8.1043.2.1 `uint8_t UnpackSwiAvmsEventReportWAMSPParamChange::TlvPresent`

8.1043.2.2 `uint16_t UnpackSwiAvmsEventReportWAMSPParamChange::wWamsChangeMask`

8.1044 unpackWdsProfileParam Union Reference

Data Fields

- [LibpackProfile3GPP SlqsProfile3GPP](#)
- [LibpackProfile3GPP2 SlqsProfile3GPP2](#)

8.1044.1 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"> • See LibpackProfile3GPP
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> • See LibpackProfile3GPP2

8.1044.2 Field Documentation

8.1044.2.1 `LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP`

8.1044.2.2 `LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2`

8.1045 unpackWdsProfileParamV2 Union Reference

Data Fields

- [LibpackProfile3GPPV2 SlqsProfile3GPP](#)
- [LibpackProfile3GPP2 SlqsProfile3GPP2](#)

8.1045.1 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"> • See LibpackProfile3GPP
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> • See LibpackProfile3GPP2

8.1045.2 Field Documentation

8.1045.2.1 LibpackProfile3GPPV2 unpackWdsProfileParamV2::SlqsProfile3GPP

8.1045.2.2 LibpackProfile3GPP2 unpackWdsProfileParamV2::SlqsProfile3GPP2

8.1046 voice_airTimer Struct Reference

Data Fields

- [uint8_t namID](#)
- [uint32_t airTimerValue](#)

8.1046.1 Detailed Description

This structure contains information about the Air Timer.

Parameters

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

8.1046.2 Field Documentation

8.1046.2.1 [uint32_t voice_airTimer::airTimerValue](#)

8.1046.2.2 [uint8_t voice_airTimer::namID](#)

8.1047 voice_allCallsAlphaIDInfo Struct Reference

Data Fields

- [uint8_t callID](#)
- [voice_alphaIDInfo AlphaIDInfo](#)

8.1047.1 Detailed Description

This structure contains information for Alpha Identifier for All Calls

Parameters

<i>callID</i>	<ul style="list-style-type: none"> • Unique call identifier for the call.
---------------	--

<i>AlphaDInfo</i>	<ul style="list-style-type: none"> • See voice_alphaDInfo for more information.
-------------------	--

8.1047.2 Field Documentation

8.1047.2.1 [voice_alphaDInfo](#) [voice_allCallsAlphaDInfo::AlphaDInfo](#)

8.1047.2.2 [uint8_t](#) [voice_allCallsAlphaDInfo::callID](#)

8.1048 [voice_allCallsDiagInfo](#) Struct Reference

Data Fields

- [uint8_t](#) [callID](#)
- [voice_diagInfo](#) [DiagInfo](#)

8.1048.1 Detailed Description

This structure contains Diagnostic Information for All Calls

Parameters

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

8.1048.2 Field Documentation

8.1048.2.1 [uint8_t](#) [voice_allCallsDiagInfo::callID](#)

8.1048.2.2 [voice_diagInfo](#) [voice_allCallsDiagInfo::DiagInfo](#)

8.1049 [voice_allCallsUUSInfo](#) Struct Reference

Data Fields

- [uint8_t](#) [callID](#)
- [voice_UUSInfo](#) [uusInfo](#)

8.1049.1 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"> • Unique call identifier for the call.
---------------	--

<i>uusInfo</i>	<ul style="list-style-type: none"> See voice_UUSInfo for more information.
----------------	---

8.1049.2 Field Documentation

8.1049.2.1 `uint8_t voice_allCallsUUSInfo::callID`

8.1049.2.2 `voice_UUSInfo voice_allCallsUUSInfo::uusInfo`

8.1050 voice_alphalDInfo Struct Reference

Data Fields

- `uint8_t alphaDcs`
- `uint8_t alphaLen`
- `uint8_t alphaText [255]`

8.1050.1 Detailed Description

This structure contains information about the Alpha Identifier.

Parameters

<i>alphaDcs</i>	<ul style="list-style-type: none"> Alpha coding scheme <ul style="list-style-type: none"> 0x01 - GSM Default_Char 0x02 - UCS2 0xFF - Not Available
<i>alphaLen</i>	<ul style="list-style-type: none"> Number of sets of the following elements: <ul style="list-style-type: none"> pAlpha_text If zero(0) then no further information exists.
<i>alphaText[MAX_VOICE_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> Data encoded as per the alpha_dcs

8.1050.2 Field Documentation

8.1050.2.1 `uint8_t voice_alphalDInfo::alphaDcs`

8.1050.2.2 `uint8_t voice_alphalDInfo::alphaLen`

8.1050.2.3 `uint8_t voice_alphalDInfo::alphaText[255]`

8.1051 voice_arrAlertingPattern Struct Reference

Data Fields

- uint8_t [numInstances](#)
- uint8_t [callID](#) [20]
- uint32_t [alertingPattern](#) [20]

8.1051.1 Detailed Description

This structure contains an array of Alerting Pattern.

Parameters

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

8.1051.2 Field Documentation

8.1051.2.1 uint32_t voice_arrAlertingPattern::alertingPattern[20]

8.1051.2.2 uint8_t voice_arrAlertingPattern::callID[20]

8.1051.2.3 uint8_t voice_arrAlertingPattern::numInstances

8.1052 voice_arrAlertingType Struct Reference

Data Fields

- uint8_t [numInstances](#)
- uint8_t [callID](#) [20]
- uint8_t [AlertingType](#) [20]

8.1052.1 Detailed Description

This structure contains an array of Alerting Type.

Parameters

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

8.1052.2 Field Documentation

8.1052.2.1 uint8_t voice_arrAlertingType::AlertingType[20]

8.1052.2.2 uint8_t voice_arrAlertingType::callID[20]

8.1052.2.3 uint8_t voice_arrAlertingType::numInstances

8.1053 voice_arrAlphaID Struct Reference

Data Fields

- uint8_t [numInstances](#)
- [voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr](#) [20]

8.1053.1 Detailed Description

This structure contains an array of Alpha ID Info

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • Number of allCallsAlphaIDInfo that follow. • If zero(0) then no further information exists.
<i>allCallsAlphaIDInfo[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> • Array of allCallsAlphaIDInfo. • See voice_allCallsAlphaIDInfo for more information.

8.1053.2 Field Documentation

8.1053.2.1 [voice_allCallsAlphaIDInfo](#) [voice_arrAlphaID::allCallsAlphaIDInfoArr](#)[20]8.1053.2.2 uint8_t [voice_arrAlphaID::numInstances](#)

8.1054 voice_arrCalledPartyNum Struct Reference

Data Fields

- [uint8_t numInstances](#)
- [voice_peerNumberInfo CalledPartyNum](#) [20]

8.1054.1 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"> • Number of calledPartyNum that follow. • If zero(0) then no further information exists.
<i>CalledParty-Num[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> • Array of CalledPartyNum. • See voice_peerNumberInfo for more information.

8.1054.2 Field Documentation

8.1054.2.1 [voice_peerNumberInfo](#) [voice_arrCalledPartyNum::CalledPartyNum](#)[20]

8.1054.2.2 [uint8_t](#) [voice_arrCalledPartyNum::numInstances](#)

8.1055 voice_arrCallEndReason Struct Reference

Data Fields

- [uint8_t numInstances](#)
- [uint8_t callID](#) [20]
- [uint16_t callEndReason](#) [20]

8.1055.1 Detailed Description

This structure contains an array of Call End Reasons.

Parameters

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

8.1055.2 Field Documentation

8.1055.2.1 uint16_t voice_arrCallEndReason::callEndReason[20]

8.1055.2.2 uint8_t voice_arrCallEndReason::callID[20]

8.1055.2.3 uint8_t voice_arrCallEndReason::numInstances

8.1056 voice_arrCallInfo Struct Reference

Data Fields

- uint8_t [numInstances](#)
- [voice_getAllCallInformation](#) [getAllCallInfo](#) [20]

8.1056.1 Detailed Description

This structure contains an array of Call Info

Parameters

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

8.1056.2 Field Documentation

8.1056.2.1 [voice_getAllCallInformation](#) [voice_arrCallInfo::getAllCallInfo](#)[20]

8.1056.2.2 uint8_t voice_arrCallInfo::numInstances

8.1057 voice_arrConnectPartyNum Struct Reference

Data Fields

- uint8_t [numInstances](#)
- [voice_peerNumberInfo](#) [ConnectedPartyNum](#) [20]

8.1057.1 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">• Number of ConnectedPartyNum that follow.• If zero(0) then no further information exists.
---------------------	---

<i>ConnectedPartyNum</i> [VOICE_MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> • Array of ConnectedPartyNum. • See voice_peerNumberInfo for more information.
--	---

8.1057.2 Field Documentation

8.1057.2.1 [voice_peerNumberInfo](#) [voice_arrConnectPartyNum::ConnectedPartyNum](#)[20]

8.1057.2.2 [uint8_t](#) [voice_arrConnectPartyNum::numInstances](#)

8.1058 [voice_arrDiagInfo](#) Struct Reference

Data Fields

- [uint8_t](#) [numInstances](#)
- [voice_allCallsDiagInfo](#) [DiagInfo](#) [20]

8.1058.1 Detailed Description

This structure contains an array of Diagnostic Information.

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • Number of DiagInfo that follow. • If zero(0) then no further information exists.
<i>DiagInfo</i> [VOICE_MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> • Array of DiagInfo. • See voice_allCallsDiagInfo for more information.

8.1058.2 Field Documentation

8.1058.2.1 [voice_allCallsDiagInfo](#) [voice_arrDiagInfo::DiagInfo](#)[20]

8.1058.2.2 [uint8_t](#) [voice_arrDiagInfo::numInstances](#)

8.1059 [voice_arrRedirPartyNum](#) Struct Reference

Data Fields

- [uint8_t](#) [numInstances](#)
- [voice_peerNumberInfo](#) [RedirPartyNum](#) [20]

8.1059.1 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"> • Number of redirPartyNum that follow. • If zero(0) then no further information exists.
<i>RedirPartyNum[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> • Array of RedirPartyNum. • See voice_peerNumberInfo for more information.

8.1059.2 Field Documentation

8.1059.2.1 uint8_t voice_arrRedirPartyNum::numInstances

8.1059.2.2 voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]

8.1060 voice_arrRemotePartyName Struct Reference

Data Fields

- uint8_t [numInstances](#)
- [voice_getAllCallRmtPtyName GetAllCallRmtPtyName](#) [20]

8.1060.1 Detailed Description

This structure contains an array of Remote Party Names

Parameters

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

8.1060.2 Field Documentation

8.1060.2.1 voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]

8.1060.2.2 uint8_t voice_arrRemotePartyName::numInstances

8.1061 voice_arrRemotePartyNum Struct Reference

Data Fields

- uint8_t [numInstances](#)
- [voice_getAllCallRmtPtyNum RmtPtyNum](#) [20]

8.1061.1 Detailed Description

This structure contains an array of Remote Party Numbers

Parameters

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

8.1061.2 Field Documentation

8.1061.2.1 `uint8_t voice_arrRemotePartyNum::numInstances`

8.1061.2.2 `voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]`

8.1062 voice_arrSvcOption Struct Reference

Data Fields

- `uint8_t numInstances`
- `uint8_t callID [20]`
- `uint16_t srvOption [20]`

8.1062.1 Detailed Description

This structure contains array an of Servicing option.

Parameters

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

8.1062.2 Field Documentation

8.1062.2.1 `uint8_t voice_arrSvcOption::callID[20]`

8.1062.2.2 `uint8_t voice_arrSvcOption::numInstances`

8.1062.2.3 `uint16_t voice_arrSvcOption::srvOption[20]`

8.1063 voice_arrUUSInfo Struct Reference

Data Fields

- `uint8_t numInstances`
- `voice_allCallsUUSInfo AllCallsUUSInfo [20]`

8.1063.1 Detailed Description

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

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • Number of allCallsUUSInfo that follow. • If zero(0) then no further information exists.
<i>AllCallsUUS-Info[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> • Array of allCallsUUSInfo. • See voice_allCallsUUSInfo for more information.

8.1063.2 Field Documentation

8.1063.2.1 `voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]`

8.1063.2.2 `uint8_t voice_arrUUSInfo::numInstances`

8.1064 voice_burstDTMFInfo Struct Reference

Data Fields

- `uint8_t * pCallID`
- `uint8_t digitCnt`
- `uint8_t pDigitBuff [255]`

8.1064.1 Detailed Description

This structure contains Voice Burst DTMF Information

Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> • 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. • 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. • If the call ID value received is 0, no value has been returned by the device • NULL pointer - Invalid data.
----------------	--

<i>digitCnt</i>	<ul style="list-style-type: none"> Length of DTMF digit buffer which follows
<i>pDigitBuff[MAX_VOICE_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> DTMF digit buffer in ASCII, NULL terminated

8.1064.2 Field Documentation

8.1064.2.1 `uint8_t voice_burstDTMFInfo::digitCnt`

8.1064.2.2 `uint8_t* voice_burstDTMFInfo::pCallID`

8.1064.2.3 `uint8_t voice_burstDTMFInfo::pDigitBuff[255]`

8.1065 `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]`

8.1065.1 Detailed Description

This structure contains Called party Number Information

Parameters

<i>PI</i>	<ul style="list-style-type: none"> Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.
<i>SI</i>	<ul style="list-style-type: none"> Number of sets of following elements <ul style="list-style-type: none"> Caller Id
<i>SI</i>	<ul style="list-style-type: none"> Number screening indicator. Values: <ul style="list-style-type: none"> 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network

<i>numType</i>	<ul style="list-style-type: none"> • Number type. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
<i>numPlan</i>	<ul style="list-style-type: none"> • Number plan. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension
<i>numLen</i>	<ul style="list-style-type: none"> • Provides the length of number which follow.
<i>number[255]</i>	<ul style="list-style-type: none"> • number of numLen length, NULL terminated.

8.1065.2 Field Documentation

8.1065.2.1 `uint8_t voice_calledPartyInfo::number[255]`

8.1065.2.2 `uint8_t voice_calledPartyInfo::numLen`

8.1065.2.3 `uint8_t voice_calledPartyInfo::numPlan`

8.1065.2.4 `uint8_t voice_calledPartyInfo::numType`

8.1065.2.5 `uint8_t voice_calledPartyInfo::PI`

8.1065.2.6 `uint8_t voice_calledPartyInfo::SI`

8.1066 voice_calledPartySubAdd Struct Reference

Data Fields

- `uint8_t extBit`

- uint8_t [subAddrType](#)
- uint8_t [oddEvenInd](#)
- uint8_t [subAddrLen](#)
- uint8_t [subAddr](#) [255]

8.1066.1 Detailed Description

This structure contains information about the Called Sub Party Addresses.

Parameters

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

8.1066.2 Field Documentation

8.1066.2.1 uint8_t voice_calledPartySubAdd::extBit

8.1066.2.2 uint8_t voice_calledPartySubAdd::oddEvenInd

8.1066.2.3 uint8_t voice_calledPartySubAdd::subAddr[255]

8.1066.2.4 uint8_t voice_calledPartySubAdd::subAddrLen

8.1066.2.5 uint8_t voice_calledPartySubAdd::subAddrType

8.1067 voice_callerIDInfo Struct Reference

Data Fields

- uint8_t [PI](#)
- uint8_t [callerIDLen](#)
- uint8_t [callerID](#) [255]

8.1067.1 Detailed Description

This structure contains Caller ID Information

Parameters

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

8.1067.2 Field Documentation

8.1067.2.1 `uint8_t voice_callerIDInfo::callerID[255]`

8.1067.2.2 `uint8_t voice_callerIDInfo::callerIDLen`

8.1067.2.3 `uint8_t voice_callerIDInfo::PI`

8.1068 voice_callFwdTypeAndPlan Struct Reference

Data Fields

- `uint8_t numberType`
- `uint8_t numberPlan`

8.1068.1 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">• Call forwarding number type<ul style="list-style-type: none">– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN Unknown– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL International– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL National– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC Network-specific– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER Subscriber– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED Reserved– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED Abbreviated– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION Reserved extension
-------------------	--

<i>numberPlan</i>	<ul style="list-style-type: none"> • Call forwarding number plan <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION Reserved extension
-------------------	---

8.1068.2 Field Documentation

8.1068.2.1 `uint8_t voice_callFwdTypeAndPlan::numberPlan`

8.1068.2.2 `uint8_t voice_callFwdTypeAndPlan::numberType`

8.1069 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]

8.1069.1 Detailed Description

This structure contains information for Get Call Forwarding Extended Information.

Parameters

<i>SvcStatus</i>	<ul style="list-style-type: none"> • Service status. Values: <ul style="list-style-type: none"> – 0x00 - SERVICE_STATUS_INACTIVE - Inactive – 0x01 - SERVICE_STATUS_ACTIVE - Active
<i>SvcClass</i>	<ul style="list-style-type: none"> • Service Class is a combination (sum) of information class constants • See qaGobiApiTableSupServiceInfoClasses.h for service classes.
<i>noReplyTimer</i>	<ul style="list-style-type: none"> • No reply timer value in seconds • A value of 0 indicates that no_reply_timer is ignored.

<i>PI</i>	<ul style="list-style-type: none"> • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.
<i>SI</i>	<ul style="list-style-type: none"> • Number screening indicator. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
<i>numType</i>	<ul style="list-style-type: none"> • Number type. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
<i>numPlan</i>	<ul style="list-style-type: none"> • Number plan. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension
<i>numLen</i>	<ul style="list-style-type: none"> • Provides the length of number which follow.
<i>number[255]</i>	<ul style="list-style-type: none"> • number of numLen length, NULL terminated.

8.1069.2 Field Documentation

8.1069.2.1 `uint8_t voice_callFWExtInfo::noReplyTimer`

8.1069.2.2 `uint8_t voice_callFWExtInfo::number[255]`

8.1069.2.3 uint8_t voice_callFWExtInfo::numLen

8.1069.2.4 uint8_t voice_callFWExtInfo::numPlan

8.1069.2.5 uint8_t voice_callFWExtInfo::numType

8.1069.2.6 uint8_t voice_callFWExtInfo::PI

8.1069.2.7 uint8_t voice_callFWExtInfo::SI

8.1069.2.8 uint8_t voice_callFWExtInfo::SvcClass

8.1069.2.9 uint8_t voice_callFWExtInfo::SvcStatus

8.1070 voice_callFWInfo Struct Reference

Data Fields

- uint8_t [SvcStatus](#)
- uint8_t [SvcClass](#)
- uint8_t [numLen](#)
- uint8_t [number](#) [255]
- uint8_t [noReplyTimer](#)

8.1070.1 Detailed Description

This structure contains information for Get Call Forwarding Information.

Parameters

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

8.1070.2 Field Documentation

8.1070.2.1 uint8_t voice_callFWInfo::noReplyTimer

8.1070.2.2 `uint8_t voice_callFWInfo::number[255]`

8.1070.2.3 `uint8_t voice_callFWInfo::numLen`

8.1070.2.4 `uint8_t voice_callFWInfo::SvcClass`

8.1070.2.5 `uint8_t voice_callFWInfo::SvcStatus`

8.1071 voice_callInfo Struct Reference

Data Fields

- `uint8_t callID`
- `uint8_t callState`
- `uint8_t callType`
- `uint8_t direction`
- `uint8_t mode`

8.1071.1 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">• Call identifier for the call queried for information.• If zero(0) then invalid.
<i>callState</i>	<ul style="list-style-type: none">• Call state.<ul style="list-style-type: none">– 0x01 - CALL_STATE_ORIGINATION - Origination– 0x02 - CALL_STATE_INCOMING - Incoming– 0x03 - CALL_STATE_CONVERSATION - Conversation– 0x04 - CALL_STATE_CC_IN_PROGRESS - Call is originating but waiting for call control to complete– 0x05 - CALL_STATE_ALERTING - Alerting– 0x06 - CALL_STATE_HOLD - Hold– 0x07 - CALL_STATE_WAITING - Waiting– 0x08 - CALL_STATE_DISCONNECTING - Disconnecting– 0x09 - CALL_STATE_END - End– 0x0A - CALL_STATE_SETUP - MT call is in Setup state in 3GPP– 0xFF - Not Available

<i>callType</i>	<ul style="list-style-type: none"> • Call type. <ul style="list-style-type: none"> – 0x00 - CALL_TYPE_VOICE - Voice – 0x02 - CALL_TYPE_VOICE_IP - Voice over IP – 0x06 - CALL_TYPE_OTAPA - OTAPA – 0x07 - CALL_TYPE_STD_OTASP - Standard OTASP – 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP – 0x09 - CALL_TYPE_EMERGENCY - Emergency – 0xFF - Not Available
<i>direction</i>	<ul style="list-style-type: none"> • Direction. <ul style="list-style-type: none"> – 0x01 - CALL_DIRECTION_MO - MO call – 0x02 - CALL_DIRECTION_MT - MT call – 0xFF - Not Available
<i>mode</i>	<ul style="list-style-type: none"> • Mode. • 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"> – 0x01 - CALL_MODE_CDMA - CDMA – 0x02 - CALL_MODE_GSM - GSM – 0x03 - CALL_MODE_UMTS - UMTS – 0x04 - CALL_MODE_LTE - LTE – 0x05 - CALL_MODE_TDS - TD-SCDMA – 0xFF - Not Available

8.1071.2 Field Documentation

8.1071.2.1 `uint8_t voice_callInfo::callID`

8.1071.2.2 `uint8_t voice_callInfo::callState`

8.1071.2.3 `uint8_t voice_callInfo::callType`

8.1071.2.4 `uint8_t voice_callInfo::direction`

8.1071.2.5 `uint8_t voice_callInfo::mode`

8.1072 `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]

8.1072.1 Detailed Description

This structure contains Calling party Number Information

Parameters

<i>PI</i>	<ul style="list-style-type: none"> • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.
<i>SI</i>	<ul style="list-style-type: none"> • Number of sets of following elements <ul style="list-style-type: none"> – Caller Id
<i>SI</i>	<ul style="list-style-type: none"> • Number screening indicator. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
<i>numType</i>	<ul style="list-style-type: none"> • Number type. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
<i>numPlan</i>	<ul style="list-style-type: none"> • Number plan. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension
<i>numLen</i>	<ul style="list-style-type: none"> • Provides the length of number which follow.

<i>number[255]</i>	<ul style="list-style-type: none"> • number of numLen length, NULL terminated.
--------------------	---

8.1072.2 Field Documentation

8.1072.2.1 uint8_t voice_callingPartyInfo::number[255]

8.1072.2.2 uint8_t voice_callingPartyInfo::numLen

8.1072.2.3 uint8_t voice_callingPartyInfo::numPlan

8.1072.2.4 uint8_t voice_callingPartyInfo::numType

8.1072.2.5 uint8_t voice_callingPartyInfo::PI

8.1072.2.6 uint8_t voice_callingPartyInfo::SI

8.1073 voice_ccSUPSType Struct Reference

Data Fields

- uint8_t [svcType](#)
- uint8_t [reason](#)

8.1073.1 Detailed Description

This structure contains information about the Call Control Supplementary Service Types

Parameters

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

8.1073.2 Field Documentation

8.1073.2.1 uint8_t voice_ccSUPSType::reason

8.1073.2.2 uint8_t voice_ccSUPSType::svcType

8.1074 voice_CLIPResp Struct Reference

Data Fields

- uint8_t [ActiveStatus](#)
- uint8_t [ProvisionStatus](#)

8.1074.1 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">• Active status.• Values:<ul style="list-style-type: none">– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive– 0x01 - ACTIVE_STATUS_ACTIVE - Active– 0xFF - Not Available
<i>ProvisionStatus</i>	<ul style="list-style-type: none">• Provisioned status.• Values:<ul style="list-style-type: none">– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned– 0xFF - Not Available

8.1074.2 Field Documentation

8.1074.2.1 uint8_t voice_CLIPResp::ActiveStatus

8.1074.2.2 uint8_t voice_CLIPResp::ProvisionStatus

8.1075 voice_CLIRResp Struct Reference

Data Fields

- uint8_t [ActiveStatus](#)
- uint8_t [ProvisionStatus](#)

8.1075.1 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"> • Active status. • Values: <ul style="list-style-type: none"> – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
<i>ProvisionStatus</i>	<ul style="list-style-type: none"> • Provisioned status. • Values: <ul style="list-style-type: none"> – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED_PERMANENT - Permanently provisioned – 0x02 - PROVISION_STATUS_PRESENTATION_RESTRICTED - Restricted presentation – 0x03 - PROVISION_STATUS_PRESENTATION_ALLOWED - Allowed presentation – 0xFF - Not Available

8.1075.2 Field Documentation

8.1075.2.1 uint8_t voice_CLIRResp::ActiveStatus

8.1075.2.2 uint8_t voice_CLIRResp::ProvisionStatus

8.1076 voice_CNAPResp Struct Reference

Data Fields

- uint8_t [ActiveStatus](#)
- uint8_t [ProvisionStatus](#)

8.1076.1 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"> • Active status. • Values: <ul style="list-style-type: none"> – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
<i>ProvisionStatus</i>	<ul style="list-style-type: none"> • Provisioned status. • Values: <ul style="list-style-type: none"> – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.1076.2 Field Documentation

8.1076.2.1 `uint8_t voice_CNAPResp::ActiveStatus`

8.1076.2.2 `uint8_t voice_CNAPResp::ProvisionStatus`

8.1077 voice_COLPResp Struct Reference

Data Fields

- `uint8_t` [ActiveStatus](#)
- `uint8_t` [ProvisionStatus](#)

8.1077.1 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">• Active status.• Values:<ul style="list-style-type: none">– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive– 0x01 - ACTIVE_STATUS_ACTIVE - Active– 0xFF - Not Available
<i>ProvisionStatus</i>	<ul style="list-style-type: none">• Provisioned status.• Values:<ul style="list-style-type: none">– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned– 0xFF - Not Available

8.1077.2 Field Documentation

8.1077.2.1 `uint8_t voice_COLPResp::ActiveStatus`

8.1077.2.2 `uint8_t voice_COLPResp::ProvisionStatus`

8.1078 voice_COLRResp Struct Reference

Data Fields

- `uint8_t` [ActiveStatus](#)
- `uint8_t` [ProvisionStatus](#)

8.1078.1 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"> • Active status. • Values: <ul style="list-style-type: none"> – 0x00 - ACTIVE_STATUS_INACTIVE - Inactive – 0x01 - ACTIVE_STATUS_ACTIVE - Active – 0xFF - Not Available
<i>ProvisionStatus</i>	<ul style="list-style-type: none"> • Provisioned status. • Values: <ul style="list-style-type: none"> – 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned – 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned – 0xFF - Not Available

8.1078.2 Field Documentation

8.1078.2.1 uint8_t voice_COLRResp::ActiveStatus

8.1078.2.2 uint8_t voice_COLRResp::ProvisionStatus

8.1079 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]

8.1079.1 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"> • Presentation indicator <ul style="list-style-type: none"> – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
-------------------	--

<i>screeningInd</i>	<ul style="list-style-type: none"> Screening indicator. <ul style="list-style-type: none"> 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network 0xFF - Not Available
<i>numType</i>	<ul style="list-style-type: none"> Number type. <ul style="list-style-type: none"> 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension 0xFF - Not Available
<i>numPlan</i>	<ul style="list-style-type: none"> Number plan. <ul style="list-style-type: none"> 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension 0xFF - Not Available
<i>callerIDLen</i>	<ul style="list-style-type: none"> Provides the length of caller ID which follow. If zero(0) then no further information exists.
<i>callerID[MAX_VOICE_CALL_NO_LEN]</i>	<ul style="list-style-type: none"> callerID of numLen length, NULL terminated.

8.1079.2 Field Documentation

8.1079.2.1 `uint8_t voice_connectNumInfo::callerID[81]`

8.1079.2.2 `uint8_t voice_connectNumInfo::callerIDLen`

8.1079.2.3 `uint8_t voice_connectNumInfo::numPlan`

8.1079.2.4 `uint8_t voice_connectNumInfo::numPresInd`

8.1079.2.5 `uint8_t voice_connectNumInfo::numType`

8.1079.2.6 `uint8_t voice_connectNumInfo::screeningInd`

8.1080 `voice_CUGInfo` Struct Reference

Data Fields

- `uint16_t` [CUGIndex](#)
- `uint8_t` [SuppPrefCUG](#)
- `uint8_t` [SuppOA](#)

8.1080.1 Detailed Description

This structure contains Closed User Group Information

Parameters

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

8.1080.2 Field Documentation

8.1080.2.1 `uint16_t voice_CUGInfo::CUGIndex`

8.1080.2.2 `uint8_t voice_CUGInfo::SuppOA`

8.1080.2.3 `uint8_t voice_CUGInfo::SuppPrefCUG`

8.1081 `voice_curAMRConfig` Struct Reference

Data Fields

- `uint8_t` [gsmAmrStat](#)
- `uint8_t` [wcdmaAmrStat](#)

8.1081.1 Detailed Description

This structure contains the Current Adaptive Multi Rate Configuration Information.

Parameters

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

8.1081.2 Field Documentation

8.1081.2.1 uint8_t voice_curAMRConfig::gsmAmrStat

8.1081.2.2 uint8_t voice_curAMRConfig::wcdmaAmrStat

8.1082 voice_diagInfo Struct Reference

Data Fields

- uint8_t [diagInfoLen](#)
- uint8_t [diagnosticInfo](#) [255]

8.1082.1 Detailed Description

This structure contains Diagnostic Information

Parameters

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

8.1082.2 Field Documentation

8.1082.2.1 uint8_t voice_diagInfo::diagInfoLen

8.1082.2.2 uint8_t voice_diagInfo::diagnosticInfo[255]

8.1083 voice_DTMFInfo Struct Reference

Data Fields

- uint8_t [callID](#)
- uint8_t [DTMFEvent](#)
- uint8_t [digitCnt](#)
- uint8_t [digitBuff](#) [255]

8.1083.1 Detailed Description

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

Parameters

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

8.1083.2 Field Documentation

8.1083.2.1 uint8_t voice_DTMFInfo::callID

8.1083.2.2 uint8_t voice_DTMFInfo::digitBuff[255]

8.1083.2.3 uint8_t voice_DTMFInfo::digitCnt

8.1083.2.4 uint8_t voice_DTMFInfo::DTMFEvent

8.1084 voice_DTMFLengths Struct Reference

Data Fields

- uint8_t [DTMFPulseWidth](#)
- uint8_t [DTMFInterdigitInterval](#)

8.1084.1 Detailed Description

This structure contains Voice Burst DTMF pulse length information

Parameters

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

8.1084.2 Field Documentation

8.1084.2.1 uint8_t voice_DTMFLengths::DTMFInterdigitInterval

8.1084.2.2 uint8_t voice_DTMFLengths::DTMFPulseWidth

8.1085 voice_ECTNum Struct Reference

Data Fields

- uint8_t [ECTCallState](#)
- uint8_t [presentationInd](#)
- uint8_t [number](#) [81]

8.1085.1 Detailed Description

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

Parameters

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

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

8.1085.2 Field Documentation

8.1085.2.1 `uint8_t voice_ECTNum::ECTCallState`

8.1085.2.2 `uint8_t voice_ECTNum::number[81]`

8.1085.2.3 `uint8_t voice_ECTNum::presentationInd`

8.1086 voice_extDispRecInfo Struct Reference

Data Fields

- `uint8_t dispType`
- `uint8_t extDispInfoLen`
- `uint8_t extDispInfo [255]`

8.1086.1 Detailed Description

This structure contains Line Control Information

Parameters

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

8.1086.2 Field Documentation

8.1086.2.1 `uint8_t voice_extDispRecInfo::dispType`

8.1086.2.2 `uint8_t voice_extDispRecInfo::extDispInfo[255]`

8.1086.2.3 `uint8_t voice_extDispRecInfo::extDispInfoLen`

8.1087 voice_getAllCallInformation Struct Reference

Data Fields

- [voice_callInfo](#) `Callinfo`
- `uint8_t isEmpty`
- `uint8_t ALS`

8.1087.1 Detailed Description

This structure contains information related to call state change.

Parameters

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

8.1087.2 Field Documentation

8.1087.2.1 `uint8_t voice_getAllCallInformation::ALS`

8.1087.2.2 `voice_callInfo voice_getAllCallInformation::Callinfo`

8.1087.2.3 `uint8_t voice_getAllCallInformation::isEmpty`

8.1088 voice_getAllCallRmtPtyName Struct Reference

Data Fields

- `uint8_t callID`
- [voice_remotePartyName](#) `RemotePartyName`

8.1088.1 Detailed Description

This structure contains information for All Call Remote Party Names

Parameters

<i>callID</i>	<ul style="list-style-type: none"> Unique call identifier for the call.
<i>RemoteParty-Name</i>	<ul style="list-style-type: none"> See voice_remotePartyName for more information.

8.1088.2 Field Documentation8.1088.2.1 `uint8_t voice_getAllCallRmtPtyName::callID`8.1088.2.2 `voice_remotePartyName voice_getAllCallRmtPtyName::RemotePartyName`**8.1089 voice_getAllCallRmtPtyNum Struct Reference****Data Fields**

- `uint8_t callID`
- `voice_remotePartyNum RemotePartyNum`

8.1089.1 Detailed Description

This structure contains information for All Call Remote Party Numbers

Parameters

<i>callID</i>	<ul style="list-style-type: none"> Unique call identifier for the call.
<i>RemoteParty-Num</i>	<ul style="list-style-type: none"> See voice_remotePartyNum for more information.

8.1089.2 Field Documentation8.1089.2.1 `uint8_t voice_getAllCallRmtPtyNum::callID`8.1089.2.2 `voice_remotePartyNum voice_getAllCallRmtPtyNum::RemotePartyNum`**8.1090 voice_getCallFWExtInfo Struct Reference****Data Fields**

- `uint8_t numInstances`
- `voice_callFWExtInfo CallFWExtInfo [20]`

8.1090.1 Detailed Description

This structure contains an array of Call Forwarded Extended Information.

Parameters

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

8.1090.2 Field Documentation

8.1090.2.1 [voice_callFWExtInfo](#) [voice_getCallFWExtInfo::CallFWExtInfo](#)[20]8.1090.2.2 [uint8_t](#) [voice_getCallFWExtInfo::numInstances](#)

8.1091 voice_getCallFWInfo Struct Reference

Data Fields

- [uint8_t](#) [numInstances](#)
- [voice_callFWInfo](#) [CallFWInfo](#) [20]

8.1091.1 Detailed Description

This structure contains an array of Call Forwarded Information.

Parameters

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

8.1091.2 Field Documentation

8.1091.2.1 [voice_callFWInfo](#) [voice_getCallFWInfo::CallFWInfo](#)[20]8.1091.2.2 [uint8_t](#) [voice_getCallFWInfo::numInstances](#)

8.1092 voice_lineCtrlInfo Struct Reference

Data Fields

- [uint8_t](#) [polarityIncluded](#)
- [uint8_t](#) [toggleMode](#)
- [uint8_t](#) [revPolarity](#)
- [uint8_t](#) [pwrDenialTime](#)

8.1092.1 Detailed Description

This structure contains Line Control Information

Parameters

<i>polarityIncluded</i>	<ul style="list-style-type: none"> Included Polarity; Boolean Value
<i>toggleMode</i>	<ul style="list-style-type: none"> Toggle mode; Boolean Value
<i>revPolarity</i>	<ul style="list-style-type: none"> Reverse Polarity; Boolean Value
<i>pwrDenialTime</i>	<ul style="list-style-type: none"> Power denial time; refer to [S1, Section 3.7.5.15 Line Control] for valid values

8.1092.2 Field Documentation

8.1092.2.1 `uint8_t voice_lineCtrlInfo::polarityIncluded`

8.1092.2.2 `uint8_t voice_lineCtrlInfo::pwrDenialTime`

8.1092.2.3 `uint8_t voice_lineCtrlInfo::revPolarity`

8.1092.2.4 `uint8_t voice_lineCtrlInfo::toggleMode`

8.1093 `voice_newPwdData` Struct Reference

Data Fields

- `uint8_t newPwd` [4]
- `uint8_t newPwdAgain` [4]

8.1093.1 Detailed Description

This structure contains New Password Data.

Parameters

<i>newPwd[BARRING_PASSWORD_LENGTH]</i>	<ul style="list-style-type: none"> New password. <ul style="list-style-type: none"> Password consists of 4 ASCII digits. Range: 0000 to 9999.
<i>newPwdAgain[BARRING_PASSWORD_LENGTH]</i>	<ul style="list-style-type: none"> New password again. <ul style="list-style-type: none"> Password consists of 4 ASCII digits. Range: 0000 to 9999.

8.1093.2 Field Documentation

8.1093.2.1 uint8_t voice_newPwdData::newPwd[4]

8.1093.2.2 uint8_t voice_newPwdData::newPwdAgain[4]

8.1094 voice_NSSAudioCtrl Struct Reference

Data Fields

- uint8_t [upLink](#)
- uint8_t [downLink](#)

8.1094.1 Detailed Description

This structure contains National Supplementary Services - Audio Control Information

Parameters

<i>upLink</i>	<ul style="list-style-type: none">• Values as per[S24, 4.10 Reservation response].
<i>downLink</i>	<ul style="list-style-type: none">• Values as per[S24, 4.10 Reservation response].

8.1094.2 Field Documentation

8.1094.2.1 uint8_t voice_NSSAudioCtrl::downLink

8.1094.2.2 uint8_t voice_NSSAudioCtrl::upLink

8.1095 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]

8.1095.1 Detailed Description

This structure contains information for Connected Peer Numbers.

Parameters

<i>callID</i>	<ul style="list-style-type: none">• Unique call identifier for the call.
---------------	--

<i>numPI</i>	<ul style="list-style-type: none"> • Number presentation indicator. <ul style="list-style-type: none"> – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
<i>numSI</i>	<ul style="list-style-type: none"> • Number screening indicator. <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network
<i>numType</i>	<ul style="list-style-type: none"> • Number type. <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
<i>numPlan</i>	<ul style="list-style-type: none"> • Number plan. <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension
<i>numLen</i>	<ul style="list-style-type: none"> • Provides the length of number which follow.
<i>number</i> [MAX_VOICE_CALL_NO_LEN]	<ul style="list-style-type: none"> • number of numLen length, NULL terminated.

8.1095.2 Field Documentation

- 8.1095.2.1 `uint8_t voice_peerNumberInfo::callID`
- 8.1095.2.2 `uint8_t voice_peerNumberInfo::number[81]`
- 8.1095.2.3 `uint8_t voice_peerNumberInfo::numLen`
- 8.1095.2.4 `uint8_t voice_peerNumberInfo::numPI`
- 8.1095.2.5 `uint8_t voice_peerNumberInfo::numPlan`
- 8.1095.2.6 `uint8_t voice_peerNumberInfo::numSI`
- 8.1095.2.7 `uint8_t voice_peerNumberInfo::numType`

8.1096 voice_prefVoiceSO Struct Reference

Data Fields

- `uint8_t namID`
- `uint8_t evrcCapability`
- `uint16_t homePageVoiceSO`
- `uint16_t homeOrigVoiceSO`
- `uint16_t roamOrigVoiceSO`

8.1096.1 Detailed Description

This structure contains information about the Preferred Voice Service Options.

Parameters

<i>namID</i>	<ul style="list-style-type: none">• Index of the NAM(Number Assignment Module) to be configured.• Range 0 to 3.• Some modems support only 1 or 2 NAMs.• 0xFF,if not available.
<i>evrcCapability</i>	<ul style="list-style-type: none">• EVRC capability.• Values:<ul style="list-style-type: none">– 0x00 - Disable– 0x01 - Enable– 0xFF - Not Available

<i>homePageVoice-SO</i>	<ul style="list-style-type: none"> • 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. • Values: <ul style="list-style-type: none"> – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 0xFFFF - Not Available
<i>homeOrigVoice-SO</i>	<ul style="list-style-type: none"> • 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. • Values: <ul style="list-style-type: none"> – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 0xFFFF - Not Available
<i>roamOrigVoice-SO</i>	<ul style="list-style-type: none"> • 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. • Values: <ul style="list-style-type: none"> – 0x0000 - VOICE_SO_WILD - Any service option – 0x0001 - VOICE_SO_IS_96A - IS-96A – 0x0003 - VOICE_SO_EVRC - EVRC – 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733 – 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder – 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband – 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband – 0x8000 - VOICE_SO_13K - 13K – 0x8001 - VOICE_SO_IS_96 - IS-96 – 0x8023 - VOICE_SO_WVRC - WVRC – 0xFFFF - Not Available

8.1096.2 Field Documentation

- 8.1096.2.1 `uint8_t` voice_prefVoiceSO::evrcCapability
- 8.1096.2.2 `uint16_t` voice_prefVoiceSO::homeOrigVoiceSO
- 8.1096.2.3 `uint16_t` voice_prefVoiceSO::homePageVoiceSO
- 8.1096.2.4 `uint8_t` voice_prefVoiceSO::namID
- 8.1096.2.5 `uint16_t` voice_prefVoiceSO::roamOrigVoiceSO

8.1097 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]

8.1097.1 Detailed Description

This structure contains Redirecting Number Information

Parameters

<i>PI</i>	<ul style="list-style-type: none"> • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values.
<i>SI</i>	<ul style="list-style-type: none"> • Number of sets of following elements <ul style="list-style-type: none"> – Caller Id
<i>SI</i>	<ul style="list-style-type: none"> • Number screening indicator. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened – 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification – 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification – 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network

<i>numType</i>	<ul style="list-style-type: none"> • Number type. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International – 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National – 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific – 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber – 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved – 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated – 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension
<i>numPlan</i>	<ul style="list-style-type: none"> • Number plan. • Values: <ul style="list-style-type: none"> – 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown – 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN – 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data – 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex – 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National – 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private – 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system – 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension
<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"> • Provides the length of number which follow.
<i>number[255]</i>	<ul style="list-style-type: none"> • number of numLen length, NULL terminated.

8.1097.2 Field Documentation

8.1097.2.1 `uint8_t voice_redirNumInfo::number[255]`

8.1097.2.2 `uint8_t voice_redirNumInfo::numLen`

8.1097.2.3 `uint8_t voice_redirNumInfo::numPlan`

8.1097.2.4 `uint8_t voice_redirNumInfo::numType`

8.1097.2.5 `uint8_t voice_redirNumInfo::PI`

8.1097.2.6 `uint8_t voice_redirNumInfo::reason`

8.1097.2.7 `uint8_t voice_redirNumInfo::SI`

8.1098 voice_remotePartyName Struct Reference

Data Fields

- uint8_t [namePI](#)
- uint8_t [codingScheme](#)
- uint8_t [nameLen](#)
- uint8_t [callerName](#) [255]

8.1098.1 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"> • Name presentation indicator. <ul style="list-style-type: none"> – 0x00 - PRESENTATION_NAME_PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_NAME_PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NAME_UNAVAILABLE - Unavailable presentation – 0x03 - PRESENTATION_NAME_NAME_PRESENTATION_RESTRICTED - Restricted name presentation – 0xFF - Not Available
<i>codingScheme</i>	<ul style="list-style-type: none"> • Refer to Table10 qaGobiApiTableCodingScheme.h for coding schemes • 0xFF - Not Available
<i>nameLen</i>	<ul style="list-style-type: none"> • Provides the length of name which follow. • If zero(0) then no further information exists.
<i>callerName[MAX_VOICE_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • Name in ASCII, NULL ending.

8.1098.2 Field Documentation

8.1098.2.1 uint8_t voice_remotePartyName::callerName[255]

8.1098.2.2 uint8_t voice_remotePartyName::codingScheme

8.1098.2.3 uint8_t voice_remotePartyName::nameLen

8.1098.2.4 uint8_t voice_remotePartyName::namePI

8.1099 voice_remotePartyNum Struct Reference

Data Fields

- uint8_t [presentationInd](#)
- uint8_t [numLen](#)
- uint8_t [remPartyNumber](#) [81]

8.1099.1 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"> • Presentation indicator. <ul style="list-style-type: none"> – 0x00 - PRESENTATION_ALLOWED - Allowed presentation – 0x01 - PRESENTATION_RESTRICTED - Restricted presentation – 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation – 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific) – 0xFF - Not Available
<i>numLen</i>	<ul style="list-style-type: none"> • Provides the length of number which follow. • If zero(0) then no further information exists.
<i>remParty- Number[MAX_ CALL_NO_LEN]</i>	<ul style="list-style-type: none"> • Array of numbers in ASCII, NULL ending.

8.1099.2 Field Documentation

8.1099.2.1 uint8_t voice_remotePartyNum::numLen

8.1099.2.2 uint8_t voice_remotePartyNum::presentationInd

8.1099.2.3 uint8_t voice_remotePartyNum::remPartyNumber[81]

8.1100 voice_roamTimer Struct Reference

Data Fields

- uint8_t [namID](#)
- uint32_t [roamTimerValue](#)

8.1100.1 Detailed Description

This structure contains information about the Roam Timer.

Parameters

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

8.1100.2 Field Documentation

8.1100.2.1 `uint8_t voice_roamTimer::namID`8.1100.2.2 `uint32_t voice_roamTimer::roamTimerValue`

8.1101 voice_signalInfo Struct Reference

Data Fields

- `uint8_t` [signalType](#)
- `uint8_t` [alertPitch](#)
- `uint8_t` [signal](#)

8.1101.1 Detailed Description

This structure contains Signal Information

Parameters

<i>signalType</i>	<ul style="list-style-type: none">• Call identifier for the call.
<i>alertPitch</i>	<ul style="list-style-type: none">• Signal Information
<i>signal</i>	<ul style="list-style-type: none">• Caller ID Information

8.1101.2 Field Documentation

8.1101.2.1 `uint8_t voice_signalInfo::alertPitch`8.1101.2.2 `uint8_t voice_signalInfo::signal`8.1101.2.3 `uint8_t voice_signalInfo::signalType`

8.1102 voice_SUPSInfo Struct Reference

Data Fields

- uint8_t [svcType](#)
- uint8_t [isModByCC](#)

8.1102.1 Detailed Description

This structure contains information about the Supplementary Services.

Parameters

<i>svcType</i>	<ul style="list-style-type: none">• Service type.<ul style="list-style-type: none">– 0x01 - SERVICE_TYPE_ACTIVATE - Activate– 0x02 - SERVICE_TYPE_DEACTIVATE - Deactivate– 0x03 - SERVICE_TYPE_REGISTER - Register– 0x04 - SERVICE_TYPE_ERASE - Erase– 0x05 - SERVICE_TYPE_INTERROGATE - Interrogate– 0x06 - SERVICE_TYPE_REGISTER_PASSWORD - Register password– 0x07 - SERVICE_TYPE_USSD - USSD
<i>isModByCC</i>	<ul style="list-style-type: none">• 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">– 0 - False– 1 - True

8.1102.2 Field Documentation

8.1102.2.1 uint8_t voice_SUPSInfo::isModByCC

8.1102.2.2 uint8_t voice_SUPSInfo::svcType

8.1103 voice_USSDNotificationNetworkInfo Struct Reference

Data Fields

- uint8_t [tlvPresent](#)
- struct [voice_USSInfo](#) [networkInfo](#)

8.1103.1 Detailed Description

Structure for storing the USS info present in USSDNotification callback.

Parameters

<i>tlvPresent</i>	Values: <ul style="list-style-type: none">• 0 - Not present• 1 - Tlv is present
-------------------	--

<i>NetworkInfo</i>	<ul style="list-style-type: none">• USS Data from Network (See structure voice_USSInfo)• NULL pointer - Invalid data.
--------------------	--

8.1103.2 Field Documentation

8.1103.2.1 struct [voice_USSInfo](#) [voice_USSDNotificationNetworkInfo::networkInfo](#)

8.1103.2.2 uint8_t [voice_USSDNotificationNetworkInfo::tlvPresent](#)

8.1104 voice_USSInfo Struct Reference

Data Fields

- uint8_t [ussDCS](#)
- uint8_t [ussLen](#)
- uint8_t [ussData](#) [182]

8.1104.1 Detailed Description

This structure contains USS Information

Parameters

<i>ussDCS</i>	<ul style="list-style-type: none">• 1 - ASCII coding scheme• 2 - 8-BIT coding scheme• 3 - UCS2
<i>ussLen</i>	<ul style="list-style-type: none">• Range 1 to 182
<i>ussData</i>	<ul style="list-style-type: none">• Data encoded as per the DCS

8.1104.2 Field Documentation

8.1104.2.1 uint8_t [voice_USSInfo::ussData](#)[182]

8.1104.2.2 uint8_t [voice_USSInfo::ussDCS](#)

8.1104.2.3 uint8_t [voice_USSInfo::ussLen](#)

8.1105 voice_UUSInfo Struct Reference

Data Fields

- uint8_t [UUSType](#)
- uint8_t [UUSDcs](#)

- uint8_t [UUSDatalen](#)
- uint8_t [UUSData](#) [255]

8.1105.1 Detailed Description

This structure contains User to User Signaling Service Information.

Parameters

<i>UUSType</i>	<ul style="list-style-type: none"> • UUS type values are: <ul style="list-style-type: none"> – 0x00 - UUS_DATA – 0x01 - UUS_TYPE1_IMPLICIT – 0x02 - UUS_TYPE1_REQUIRED – 0x03 - UUS_TYPE1_NOT_REQUIRED – 0x04 - UUS_TYPE2_REQUIRED – 0x05 - UUS_TYPE2_NOT_REQUIRED – 0x06 - UUS_TYPE3_REQUIRED – 0x07 - UUS_TYPE3_NOT_REQUIRED – 0xFF - Not Available
<i>UUSDcs</i>	<ul style="list-style-type: none"> • UUS data coding scheme values are: <ul style="list-style-type: none"> – 0x01 - UUS_DCS_USP – 0x02 - UUS_DCS_OHLP – 0x03 - UUS_DCS_X244 – 0x04 - UUS_DCS_SMCf – 0x05 - UUS_DCS_IA5 – 0x06 - UUS_DCS_RV12RD – 0x07 - UUS_DCS_Q931UNCCM – 0xFF - Not Available
<i>UUSDatalen</i>	<ul style="list-style-type: none"> • Number of sets of the following elements. <ul style="list-style-type: none"> – UUSData • If zero(0) then no further information exists.
<i>UUSData[<small>MAX_VOICE_DESCRIPTION_LENGTH</small>]</i>	<ul style="list-style-type: none"> • UUS data encoded as per coding scheme

8.1105.2 Field Documentation

8.1105.2.1 uint8_t voice_UUSInfo::UUSData[255]

8.1105.2.2 uint8_t voice_UUSInfo::UUSDatalen

8.1105.2.3 uint8_t voice_UUSInfo::UUSDcs

8.1105.2.4 uint8_t voice_UUSInfo::UUSType

8.1106 wds_channelRate Struct Reference

Data Fields

- uint32_t [CurrChanTxRate](#)
- uint32_t [CurrChanRxRate](#)

8.1106.1 Detailed Description

This structure contains Channel Rate

Parameters

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

8.1106.2 Field Documentation

8.1106.2.1 uint32_t wds_channelRate::CurrChanRxRate

8.1106.2.2 uint32_t wds_channelRate::CurrChanTxRate

8.1107 wds_ChannelRateTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [wds_channelRate](#) [ChannelRate](#)

8.1107.1 Detailed Description

This structure stores information about channel rate TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• value is 1 if this TLV is present in indication, otherwise 0
<i>ChannelRate</i>	<ul style="list-style-type: none">• see wds_channelRate for more info

8.1107.2 Field Documentation

8.1107.2.1 wds_channelRate wds_ChannelRateTlv::ChannelRate

8.1107.2.2 uint8_t wds_ChannelRateTlv::TlvPresent

8.1108 wds_ConnStatusTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [MDMConnStatus](#)

8.1108.1 Detailed Description

This structure stores information about modem connection status TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• value is 1 if this TLV is present in indication, otherwise 0
<i>MDMConn- Status</i>	<ul style="list-style-type: none">-Modem connecton status• 0x01 - disconnected• 0x02 - connected

8.1108.2 Field Documentation

8.1108.2.1 uint8_t wds_ConnStatusTlv::MDMConnStatus

8.1108.2.2 uint8_t wds_ConnStatusTlv::TlvPresent

8.1109 wds_currNetworkInfo Struct Reference

Data Fields

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

8.1109.1 Detailed Description

Network information structure

Parameters

<i>NetworkType</i>	<ul style="list-style-type: none"> Values: <ul style="list-style-type: none"> 0 - 3GPP 1 - 3GPP2
<i>RATMask</i>	<ul style="list-style-type: none"> RAT mask to indicate type of technology Values <ul style="list-style-type: none"> 0x00 - DONT_CARE 0x8000 - NULL_BEARER CDMA RAT mask <ul style="list-style-type: none"> 0x01 - CDMA_1X 0x02 - EVDO_REV0 0x05 - HRPD 0x0B - EHRPD UMTS RAT mask <ul style="list-style-type: none"> 0x03 - GPRS 0x04 - WCDMA 0x06 - EDGE 0x07 - HSDPA and WCDMA 0x08 - WCDMA and HSUPA 0x09 - HSDPA and HSUPA 0x0A - LTE 0x0C - HSDPA+ and WCDMA 0x0D - HSDPA+ and HSUPA 0x0E - DC_HSDPA+ and WCDMA 0x0F - DC_HSDPA+ and HSUPA
<i>SOMask</i>	<ul style="list-style-type: none"> SO Mask

8.1109.2 Field Documentation

8.1109.2.1 uint8_t wds_currNetworkInfo::NetworkType

8.1109.2.2 uint32_t wds_currNetworkInfo::RATMask

8.1109.2.3 uint32_t wds_currNetworkInfo::SOMask

8.1110 wds_DataBearerTechTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [DataBearerTech](#)

8.1110.1 Detailed Description

This structure hold parameters about data bearer technology TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • value is 1 if this TLV is present in indication, otherwise 0
<i>DataBearerTech</i>	<ul style="list-style-type: none"> • Data bear technology <ul style="list-style-type: none"> – 0x01 - cdma2000 1X – 0x02 - cdma2000 HRPD (1xEV-DO) – 0x03 - GSM – 0x04 - UMTS – 0x05 - cdma200 HRPD (1xEV-DO RevA) – 0x06 - EDGE – 0x07 - HSDPA and WCDMA – 0x08 - WCDMA and HSUPA – 0x09 - HSDPA and HSUPA – 0x0A - LTE – 0x0B - cdma2000 EHRPD – 0x0C - HSDPA+ and WCDMA – 0x0D - HSDPA+ and HSUPA – 0x0E - DC_HSDPA+ and WCDMA – 0x0F - DC_HSDPA+ and HSUPA – 0x10 - HSDPA+ and 64QAM – 0x11 - HSDPA+, 64QAM and HSUPA – 0x12 - TDSCDMA – 0x13 - TDSCDMA and HSDPA – 0xFF - Unknown

8.1110.2 Field Documentation

8.1110.2.1 `uint8_t wds_DataBearTechTlv::DataBearerTech`8.1110.2.2 `uint8_t wds_DataBearTechTlv::TlvPresent`8.1111 `wds_DataUlongLongTlv` Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint64_t ullData`

8.1111.1 Detailed Description

This structure contains Data unsigned long long TLV data.

Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>ullData</i>	unsigend long long data in 64bit

8.1111.2 Field Documentation

8.1111.2.1 `uint8_t wds_DataULongLongTlv::TlvPresent`

8.1111.2.2 `uint64_t wds_DataULongLongTlv::ulldata`

8.1112 wds_DataULongTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint32_t ulData`

8.1112.1 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

8.1112.2 Field Documentation

8.1112.2.1 `uint8_t wds_DataULongTlv::TlvPresent`

8.1112.2.2 `uint32_t wds_DataULongTlv::ulData`

8.1113 wds_DHCPLeaseOptTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t numOpt`
- `wds_DHCPOpt optList` [30]
- `uint8_t optListData` [2048]

8.1113.1 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

8.1113.2 Field Documentation

8.1113.2.1 `uint8_t wds_DHCPLeaseOptTlv::numOpt`

8.1113.2.2 `wds_DHCPOpt` `wds_DHCPLeaseOptTlv::optList[30]`

8.1113.2.3 `uint8_t` `wds_DHCPLeaseOptTlv::optListData[2048]`

8.1113.2.4 `uint8_t` `wds_DHCPLeaseOptTlv::TlvPresent`

8.1114 `wds_DHCPLeaseStateTlv` Struct Reference

Data Fields

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

8.1114.1 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">• 0 - active, newly acquired• 1 - active, renewed• 2 - active, renewing• 3 - active, rebinding• 4 - inactive, expired• 5 - inactive, renew refused• 6 - inactive, rebind refused• 7 - inactive, other

8.1114.2 Field Documentation

8.1114.2.1 `uint8_t` `wds_DHCPLeaseStateTlv::leaseState`

8.1114.2.2 `uint8_t` `wds_DHCPLeaseStateTlv::TlvPresent`

8.1115 `wds_DHCPOpt` Struct Reference

Data Fields

- `uint8_t` [optCode](#)
- `uint8_t` [optValLen](#)
- `uint8_t *` [pOptVal](#)

8.1115.1 Detailed Description

DHCP option code information

Parameters

<i>optCode</i>	option code <ul style="list-style-type: none"> values: <ul style="list-style-type: none"> 0 - 255
<i>optValLen</i>	length of option code <ul style="list-style-type: none"> values: <ul style="list-style-type: none"> 0 - 255
<i>pOptVal</i>	option value

8.1115.2 Field Documentation

8.1115.2.1 `uint8_t wds_DHCPOpt::optCode`8.1115.2.2 `uint8_t wds_DHCPOpt::optValLen`8.1115.2.3 `uint8_t* wds_DHCPOpt::pOptVal`

8.1116 wds_DHCPProfileIdTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t profileType`
- `uint8_t profileId`

8.1116.1 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"> 0 - 3GPP
<i>profileId</i>	index identifying the profile <ul style="list-style-type: none"> 1-24 valid for 3GPP profile type (9x30 and onwards)

8.1116.2 Field Documentation

8.1116.2.1 `uint8_t wds_DHCPProfileIdTlv::profileId`8.1116.2.2 `uint8_t wds_DHCPProfileIdTlv::profileType`8.1116.2.3 `uint8_t wds_DHCPProfileIdTlv::TlvPresent`

8.1117 wds_DHCPv4HWConfig Struct Reference

Data Fields

- uint8_t [hwType](#)
- uint8_t [chaddrLen](#)
- uint8_t [chaddr](#) [16]

8.1117.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

Parameters

<i>hwType</i>	<ul style="list-style-type: none"> • HW Type 1 - Ethernet 20 - Serial
<i>chaddrlen</i>	<ul style="list-style-type: none"> • chaddrlen
<i>chaddr</i>	<ul style="list-style-type: none"> • chaddr. Max size 16 bytes

8.1117.2 Field Documentation

8.1117.2.1 uint8_t wds_DHCPv4HWConfig::chaddr[16]

8.1117.2.2 uint8_t wds_DHCPv4HWConfig::chaddrLen

8.1117.2.3 uint8_t wds_DHCPv4HWConfig::hwType

8.1118 wds_DHCPv4Option Struct Reference

Data Fields

- uint8_t [optCode](#)
- uint8_t [optValLen](#)
- uint8_t [optVal](#) [255]

8.1118.1 Detailed Description

WDS SWI DHCPv4 Option Structure

Parameters

<i>optCode</i>	<ul style="list-style-type: none"> • Option code <ul style="list-style-type: none"> – 0 - 255
<i>optValLen</i>	<ul style="list-style-type: none"> • Option value length <ul style="list-style-type: none"> – 0 - 255

<i>optVal</i>	<ul style="list-style-type: none"> Option value
---------------	--

8.1118.2 Field Documentation

8.1118.2.1 `uint8_t wds_DHCPv4Option::optCode`

8.1118.2.2 `uint8_t wds_DHCPv4Option::optVal[255]`

8.1118.2.3 `uint8_t wds_DHCPv4Option::optValLen`

8.1119 wds_DHCPv4OptionList Struct Reference

Data Fields

- `uint8_t numOpt`
- `wds_DHCPv4Option * pOptList`

8.1119.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

Parameters

<i>numOpt</i>	<ul style="list-style-type: none"> number of options <ul style="list-style-type: none"> – 0 - 255
<i>pOptList</i>	<ul style="list-style-type: none"> pointer to list of DHCP Options

8.1119.2 Field Documentation

8.1119.2.1 `uint8_t wds_DHCPv4OptionList::numOpt`

8.1119.2.2 `wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList`

8.1120 wds_DHCPv4ProfileId Struct Reference

Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

8.1120.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

Parameters

<i>profileType</i>	<ul style="list-style-type: none"> • 0 for 3GPP
<i>profileId</i>	<ul style="list-style-type: none"> • 1 to 24 for 3GPP profile

8.1120.2 Field Documentation8.1120.2.1 `uint8_t wds_DHCPv4ProfileId::profileId`8.1120.2.2 `uint8_t wds_DHCPv4ProfileId::profileType`**8.1121 wds_Domain Struct Reference****Data Fields**

- `uint16_t domainLen`
- `uint8_t domainName [256]`

8.1121.1 Detailed Description

This structure contains the DomainName Information

Parameters

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

8.1121.2 Field Documentation8.1121.2.1 `uint16_t wds_Domain::domainLen`8.1121.2.2 `uint8_t wds_Domain::domainName[256]`**8.1122 wds_DomainNameList Struct Reference****Data Fields**

- `uint8_t numInstances`
- `struct wds_Domain domain [10]`

8.1122.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

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

Note

Data invalid when [numInstances](#) equal 0xff.

8.1122.2 Field Documentation

8.1122.2.1 struct wds_Domain wds_DomainNameList::domain[10]

8.1122.2.2 uint8_t wds_DomainNameList::numInstances

8.1123 wds_DormStatTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [DormancyStat](#)

8.1123.1 Detailed Description

This structure stores information about dormancy status TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• value is 1 if this TLV is present in indication, otherwise 0
<i>DormancyStat</i>	<ul style="list-style-type: none">-Dormancy status<ul style="list-style-type: none">• 0x01 - Traffic channel dormant• 0x02 - Traffic channel active

8.1123.2 Field Documentation

8.1123.2.1 uint8_t wds_DormStatTlv::DormancyStat

8.1123.2.2 uint8_t wds_DormStatTlv::TlvPresent

8.1124 wds_GPRSQoS Struct Reference

Data Fields

- uint32_t [precedenceClass](#)
- uint32_t [delayClass](#)
- uint32_t [reliabilityClass](#)

- uint32_t [peakThroughputClass](#)
- uint32_t [meanThroughputClass](#)

8.1124.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

Parameters

<i>precedence-Class</i>	<ul style="list-style-type: none"> • Precedence class
<i>delayClass</i>	<ul style="list-style-type: none"> • Delay class
<i>reliabilityClass</i>	<ul style="list-style-type: none"> • Reliability class
<i>peak-Throughput-Class</i>	<ul style="list-style-type: none"> • Peak throughput class
<i>mean-Throughput-Class</i>	<ul style="list-style-type: none"> • Mean throughput class

Note

Data invalid when all parameters are equal to 0xffffffff.

8.1124.2 Field Documentation

8.1124.2.1 uint32_t wds_GPRSQoS::delayClass

8.1124.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

8.1124.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

8.1124.2.4 uint32_t wds_GPRSQoS::precedenceClass

8.1124.2.5 uint32_t wds_GPRSQoS::reliabilityClass

8.1125 wds_IPv4AdTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [IPv4Addr](#)

8.1125.1 Detailed Description

IPv4 address TLV information

Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>IPv4Addr</i>	IPv4 address

8.1125.2 Field Documentation

8.1125.2.1 uint32_t wds_IPv4AdTlv::IPv4Addr

8.1125.2.2 uint8_t wds_IPv4AdTlv::TlvPresent

8.1126 wds_IPV6AddressInfo Struct Reference

Data Fields

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

8.1126.1 Detailed Description

This structure contains the IPV6 Address Information

Parameters

<i>IPv6PrefixLen</i>	<ul style="list-style-type: none"> • Length of the received IPv6 address in no. of bits; can take value between 0 and 128 – 0xFF - Not Available
<i>IPAddressV6</i>	<ul style="list-style-type: none"> • IPv6 address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format.

8.1126.2 Field Documentation

8.1126.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

8.1126.2.2 uint8_t wds_IPV6AddressInfo::IPv6PrefixLen

8.1127 wds_IPV6GWAddressInfo Struct Reference

Data Fields

- uint8_t [gwV6PrefixLen](#)
- uint16_t [gwAddressV6](#) [8]

8.1127.1 Detailed Description

This structure contains the IPV6 Gateway Address Information

Parameters

<i>gwV6PrefixLen</i>	<ul style="list-style-type: none"> Length of the received IPv6 Gateway address in no. of bits; can take value between 0 and 128
<i>IPAddressV6</i>	<ul style="list-style-type: none"> 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.

Note

Data invalid when IPV6PrefixLen equal 0xff.

8.1127.2 Field Documentation

8.1127.2.1 `uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]`

8.1127.2.2 `uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen`

8.1128 wds_LastMdmCallEndRsnTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint16_t CallEndReason`

8.1128.1 Detailed Description

This structure stores information about modem last call end reason TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> value is 1 if this TLV is present in indication, otherwise 0
<i>CallEndReason</i>	-Modem call end reason <ul style="list-style-type: none"> See qaGobiApiTableCallEndReasons.h for Call End Reason

8.1128.2 Field Documentation

8.1128.2.1 `uint16_t wds_LastMdmCallEndRsnTlv::CallEndReason`

8.1128.2.2 `uint8_t wds_LastMdmCallEndRsnTlv::TlvPresent`

8.1129 wds_PCSCFFQDNAddress Struct Reference

Data Fields

- `uint16_t fqdnLen`
- `uint8_t fqdnAddr [256]`

8.1129.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

Parameters

<i>fqdnLen</i>	<ul style="list-style-type: none">length of the received FQDN address
<i>fqdnAddr</i>	<ul style="list-style-type: none">FQDN address(Max 256 characters)

8.1129.2 Field Documentation

8.1129.2.1 `uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]`

8.1129.2.2 `uint16_t wds_PCSCFFQDNAddress::fqdnLen`

8.1130 wds_PCSCFFQDNAddressList Struct Reference

Data Fields

- `uint8_t numInstances`
- struct `wds_PCSCFFQDNAddress pcsfFQDNAddress` [10]

8.1130.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

Parameters

<i>numInstances</i>	<ul style="list-style-type: none">Number of FQDN addresses received
<i>pcsfFQDN-Address</i>	<ul style="list-style-type: none">FQDN address information(Max 10 addresses)

Note

Data invalid when `numInstances` equal 0xff.

8.1130.2 Field Documentation

8.1130.2.1 `uint8_t wds_PCSCFFQDNAddressList::numInstances`

8.1130.2.2 `struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]`

8.1131 wds_PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- `uint8_t numInstances`

- uint32_t [pccsfIPV4Addr](#) [64]

8.1131.1 Detailed Description

This structure contains the PCSCFIPv4ServerAddressList Information

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • number of address following
<i>pccsfIPV4Addr</i>	<ul style="list-style-type: none"> • P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)

Note

Data invalid when [numInstances](#) equal 0xff and [pccsfIPV4Addr](#) equal to 0xffffffff.

8.1131.2 Field Documentation

8.1131.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

8.1131.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pccsfIPV4Addr[64]

8.1132 wds_ProfileIdentifier Struct Reference

Data Fields

- uint8_t [profileType](#)
- uint8_t [profileIndex](#)

8.1132.1 Detailed Description

This structure contains the Profile Identifier Information

Parameters

<i>profileType</i>	<ul style="list-style-type: none"> • Identifies the type of profile 0x00 = 3GPP
<i>profileIndex</i>	<ul style="list-style-type: none"> • 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

Note

Data invalid when all parameters are equal to 0xff.

8.1132.2 Field Documentation

8.1132.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.1132.2.2 uint8_t wds_ProfileIdentifier::profileType

8.1133 wds_profileInfo Union Reference

Data Fields

- [LibPackprofile_3GPP](#) SlqsProfile3GPP
- [LibPackprofile_3GPP2](#) SlqsProfile3GPP2

8.1133.1 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 LibPackprofile_3GPP
<i>SlqsProfile3GPP2</i>	3GPP2 profile See LibPackprofile_3GPP2

8.1133.2 Field Documentation

8.1133.2.1 [LibPackprofile_3GPP](#) wds_profileInfo::SlqsProfile3GPP8.1133.2.2 [LibPackprofile_3GPP2](#) wds_profileInfo::SlqsProfile3GPP2

8.1134 wds_RXBytesOKTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint64_t [RxByteOKCnt](#)

8.1134.1 Detailed Description

This structure stores information about Rx Bytes OK TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> • value is 1 if this TLV is present in indication, otherwise 0
<i>RxByteOKCnt</i>	-Number of bytes received without error

8.1134.2 Field Documentation

8.1134.2.1 uint64_t wds_RXBytesOKTlv::RxByteOKCnt

8.1134.2.2 uint8_t wds_RXBytesOKTlv::TlvPresent

8.1135 wds_transferStatInd Struct Reference

Data Fields

- uint8_t [StatsPeriod](#)
- uint32_t [StatsMask](#)

8.1135.1 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">• 0 - Do not report.• 1 - Other - Period between reports (in seconds).
<i>StatsMask</i>	requested statistic bit mask. <ul style="list-style-type: none">• 0x00000040 - Tx bytes OK• 0x00000080 - Rx bytes OK

8.1135.2 Field Documentation

8.1135.2.1 uint32_t wds_transferStatInd::StatsMask

8.1135.2.2 uint8_t wds_transferStatInd::StatsPeriod

8.1136 wds_TrStatInd Struct Reference

Data Fields

- uint8_t [statsPeriod](#)
- uint32_t [statsMask](#)

8.1136.1 Detailed Description

This structure contains the information about the Transfer Statistics Indicator parameters.

Parameters

<i>statsPeriod</i>	<ul style="list-style-type: none"> Period between transfer statistics reports. <ul style="list-style-type: none"> 0 - Do not report Other - Period between reports (seconds)
<i>statsMask</i>	<ul style="list-style-type: none"> Requested statistic bit mask. <ul style="list-style-type: none"> 0x00000001 - Tx packets OK 0x00000002 - Rx packets OK 0x00000004 - Tx packet errors 0x00000008 - Rx packet errors 0x00000010 - Tx overflows 0x00000020 - Rx overflows 0x00000040 - Tx bytes OK 0x00000080 - Rx bytes OK Each bit set causes the corresponding optional information to be sent in SLQSWdsEventReport-Callback. All unlisted bits are reserved for future use and must be set to zero.

8.1136.2 Field Documentation

8.1136.2.1 uint32_t wds_TrStatInd::statsMask

8.1136.2.2 uint8_t wds_TrStatInd::statsPeriod

8.1137 wds_TXBytesOKTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint64_t [TxByteOKCnt](#)

8.1137.1 Detailed Description

This structure stores information about Tx Bytes OK TLV.

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> value is 1 if this TLV is present in indication, otherwise 0
<i>TxByteOKCnt</i>	<ul style="list-style-type: none"> Number of bytes transmitted without error

8.1137.2 Field Documentation

8.1137.2.1 uint8_t wds_TXBytesOKTlv::TlvPresent

8.1137.2.2 uint64_t wds_TXBytesOKTlv::TxByteOKCnt

8.1138 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](#)

8.1138.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> • 0x00 - Subscribed • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 0x04 - Background
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> • Maximum uplink bit rate in bits/sec
<i>maxDownlink- Bitrate</i>	<ul style="list-style-type: none"> • Maximum downlink bit rate in bits/sec
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> • Guaranteed uplink bit rate in bits/sec
<i>grntDownlink- Bitrate</i>	<ul style="list-style-type: none"> • Guaranteed downlink bit rate in bits/sec
<i>qosDelivery- Order</i>	<ul style="list-style-type: none"> - Qos delivery order • 0x00 - Subscribe • 0x01 - Delivery order on • 0x02 - Delivery order off
<i>maxSDUSize</i>	<ul style="list-style-type: none"> • Maximum SDU size

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

8.1138.2 Field Documentation

8.1138.2.1 uint8_t wds_UMTSMinQoS::deliveryErrSDU

- 8.1138.2.2 `uint32_t wds_UMTSMInQoS::grntDownlinkBitrate`
- 8.1138.2.3 `uint32_t wds_UMTSMInQoS::grntUplinkBitrate`
- 8.1138.2.4 `uint32_t wds_UMTSMInQoS::maxDownlinkBitrate`
- 8.1138.2.5 `uint32_t wds_UMTSMInQoS::maxSDUSize`
- 8.1138.2.6 `uint32_t wds_UMTSMInQoS::maxUplinkBitrate`
- 8.1138.2.7 `uint8_t wds_UMTSMInQoS::qosDeliveryOrder`
- 8.1138.2.8 `uint8_t wds_UMTSMInQoS::resBerRatio`
- 8.1138.2.9 `uint8_t wds_UMTSMInQoS::sduErrorRatio`
- 8.1138.2.10 `uint8_t wds_UMTSMInQoS::trafficClass`
- 8.1138.2.11 `uint32_t wds_UMTSMInQoS::trafficPriority`
- 8.1138.2.12 `uint32_t wds_UMTSMInQoS::transferDelay`

8.1139 wdsDhcpv4HwConfig Struct Reference

Data Fields

- `uint8_t hwType`
- `uint8_t chaddrLen`
- `uint8_t chaddr [16]`

8.1139.1 Detailed Description

Structure contain DHCP V4 Hardware Configuration.

Parameters

<i>hwType</i>	DHCP HW Type, examples: <ul style="list-style-type: none"> • 0 - Ethernet • 20 - Serial
<i>chaddrLen</i>	Length of chaddr field, examples: <ul style="list-style-type: none"> • 6 for Ethernet MAC address
<i>chaddr</i>	Client hardware address

8.1139.2 Field Documentation

- 8.1139.2.1 `uint8_t wdsDhcpv4HwConfig::chaddr[16]`
- 8.1139.2.2 `uint8_t wdsDhcpv4HwConfig::chaddrLen`
- 8.1139.2.3 `uint8_t wdsDhcpv4HwConfig::hwType`

8.1140 wdsDhcpv4Option Struct Reference

Data Fields

- `uint8_t` [optCode](#)
- `uint8_t` [optValLen](#)
- `uint8_t` [optVal](#) [255]

8.1140.1 Detailed Description

This structure contains DHCP V4 Option information.

Parameters

<i>optCode</i>	Option code <ul style="list-style-type: none">• 0 - 255
<i>optValLen</i>	Option value length <ul style="list-style-type: none">• 0 - 255
<i>optVal</i>	Option Value

8.1140.2 Field Documentation

8.1140.2.1 `uint8_t` [wdsDhcpv4Option::optCode](#)

8.1140.2.2 `uint8_t` [wdsDhcpv4Option::optVal](#)[255]

8.1140.2.3 `uint8_t` [wdsDhcpv4Option::optValLen](#)

8.1141 wdsDhcpv4OptionList Struct Reference

Data Fields

- `uint8_t` [numOpt](#)
- [wdsDhcpv4Option](#) * [pOptList](#)

8.1141.1 Detailed Description

Structure contain DHCP V4 Option List

Parameters

<i>numOpt</i>	number of options <ul style="list-style-type: none">• 0 - 255
<i>pOptList</i>	pointer to list of DHCP Options <ul style="list-style-type: none">• See wdsDhcpv4Option

8.1141.2 Field Documentation

8.1141.2.1 `uint8_t wdsDhcpv4OptionList::numOpt`

8.1141.2.2 `wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList`

8.1142 wdsDhcpv4ProfileId Struct Reference

Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

8.1142.1 Detailed Description

This structure contains DHCP v4 ProfileID Information.

Parameters

<i>profileType</i>	profile type <ul style="list-style-type: none">• 0 - 3GPP
<i>profileId</i>	profile index <ul style="list-style-type: none">• index identifying the profile 1-24 valid for 3GPP profile type (EM74xx and onwards)

8.1142.2 Field Documentation

8.1142.2.1 `uint8_t wdsDhcpv4ProfileId::profileId`

8.1142.2.2 `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_TXPCMIIRFiltr](#)
- struct [audio_RXPCMIIRFiltr](#)
- 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	<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

9.2.1.6 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	<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

9.2.1.7 `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`}

Typedefs

- typedef void(* `logger`)(uint8_t lvl, const char *buff)

Enumerations

- enum `eLOG_LEVEL` {
`eLOG_INFO`,
`eLOG_DEBUG`,
`eLOG_WARN`,
`eLOG_FATAL` }
- enum `eTimeout` {
`eTIMEOUT_2_S` = 2000,
`eTIMEOUT_5_S` = 5000,
`eTIMEOUT_8_S` = 8000,
`eTIMEOUT_10_S` = 10000,
`eTIMEOUT_20_S` = 20000,
`eTIMEOUT_30_S` = 30000,
`eTIMEOUT_60_S` = 60000,
`eTIMEOUT_300_S` = 300000,
`eTIMEOUT_DEFAULT` = `eTIMEOUT_8_S` }
- enum `eQMI_SVC` {
`eCTL`,
`eWDS`,
`eDMS`,
`eNAS` =3,
`eQOS`,
`eSMS` =5,
`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 Typedef Documentation

9.4.2.1 typedef void(* logger)(uint8_t lvl, const char *buff)

9.4.3 Enumeration Type Documentation

9.4.3.1 enum eLOG_LEVEL

log levels

Enumerator

eLOG_INFO
eLOG_DEBUG
eLOG_WARN
eLOG_FATAL

9.4.3.2 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

9.4.3.3 enum eTimeout

eTimeout

Enumerator

eTIMEOUT_2_S
eTIMEOUT_5_S
eTIMEOUT_8_S
eTIMEOUT_10_S
eTIMEOUT_20_S
eTIMEOUT_30_S
eTIMEOUT_60_S
eTIMEOUT_300_S
eTIMEOUT_DEFAULT

9.4.3.4 enum msgtype

qmi message type

Enumerator

eREQ
eRSP
eIND

9.4.4 Function Documentation

9.4.4.1 void fill_pack_ctx (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t svc, int timeout)

9.4.4.2 void fill_sdu_hdr (pack_qmi_t * pCtx, uint8_t * pReqBuf)

9.4.4.3 char* get_version ()

Returns

version string

9.4.4.4 uint16_t helper_get_error_code (uint8_t * qmi_resp)

9.4.4.5 char* helper_get_error_reason (uint16_t retVal)

9.4.4.6 const char* helper_get_req_str (uint8_t svc, uint8_t * req, uint32_t len)

extract msgid string from modem req

Parameters

in	svc	qmi service
in	req	qmi request
in	len	request length

Returns

qmi message string

9.4.4.7 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	svc	qmi service
in	pbuf	qmi response/indication
in	len	response/indication length
out	pCtx	unpacked context

Returns

qmi message string

9.4.4.8 uint16_t helper_get_xid (uint8_t * qmi_resp)

9.4.4.9 int helper_isBootLoader_DebugEnabled (const char * szPath, const char * pQsn)

Parameters

in	szPath	the full path of the provided bootloader or firmware file
in	pQsn	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

9.4.4.10 int helper_set_log_func (logger * func)

set log function

9.4.4.11 int helper_set_log_lvl (uint8_t lvl)

set log level

9.4.4.12 char* liteqmi_GetVersion ()**Returns**

version string

9.4.4.13 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

9.4.4.14 void liteqmi_log (uint8_t lvl, const char * fmt, ...)**9.4.4.15 unsigned unpack_result_code_only (uint8_t * pMdmResp)**

common handler for unpacking response with TLV type 0x02 only

9.4.5 Variable Documentation**9.4.5.1 logger* glog****9.4.5.2 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_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_SLQSDmsSwiGetResetInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSDmsSwiGetResetInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSDmsSwiGetResetInfo_t](#) *pOutput)
- int [unpack_dms_SLQSDmsSwiGetResetInfo_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t](#) *pOutput)
- int [pack_dms_SLQSDmsSwiIndicationRegister](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SLQSDmsSwiIndicationRegister_t](#) *reqArg)
- int [unpack_dms_SLQSDmsSwiIndicationRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSDmsSwiIndicationRegister_t](#) *pOutput)
- int [pack_dms_SLQSSwiGetFwUpdateStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSSwiGetFwUpdateStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiGetFwUpdateStatus_t](#) *pOutput)
- 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>	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.2.2 `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_SLQSDmsSwilIndicationRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSDmsSwilIndicationRegister_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 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	<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

9.5.2.66 `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	<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

9.5.2.67 `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>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.2.68 `int pack_dms_UIMUnblockControlKey (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMUnblockControlKey_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>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.2.69 `int pack_dms_UIMUnblockPIN (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMUnblockPIN_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>	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.2.71 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>	requeset parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.2.72 `int unpack_dms_ActivateAutomatic (uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactory-Defaults_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_SetIndicationRegister_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_SLQSDmsSwiIndicationRegister (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiIndicationRegister_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_SLQSSwiGetCrashInfo_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_SLQSUIMGetState (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSUIMGetState_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_UIMSetPINProtection_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_UIMUnblockControlKey_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 [FMSImageElement](#)

- 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"> image list returned from GetStoredImages See FMSImageList
in, out	<i>pValidCombinationSize</i>	<ul style="list-style-type: none"> number of combination passed in and returned
out	<i>pValidCombinations</i>	<ul style="list-style-type: none"> valid combinations returned See CarrierImage_t

Returns

- eQCWWAN_ERR_INVALID_ARG - Invalid parameters
- eQCWWAN_ERR_BUFFER_SZ - No enough element to store combinations returned

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.2.2 `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_SLQSSetIMSSMSConfig_t](#)
- struct [unpack_ims_SLQSSetIMSSMSConfig_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_SLQSGetIMSSMSConfig_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_TmrTfInfo](#)
- 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_SLQSUserCfgCallBack_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_SLQSVoIPCfgCallBack_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_SLQSUerCfgCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_ims_SLQSUerCfgCallBack_ind_t](#) *pOutput)
- int [unpack_ims_SLQSVolPCfgCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_ims_SLQSVolPCfgCallBack_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_SLQSGetIMSSMS-Config_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_SLQSGetIMSUser-Config_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_SLQSGetIMSVoIPConfig (uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSGetIMSVoIP-Config_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_SLQSSetIMSUser-
Config_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_SLQSSetIMSVolPConfig (uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSSetIMSVolPConfig_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_SLQSSUserCfgCallBack_ind (uint8_t * pResp, uint16_t respLen,
unpack_ims_SLQSSUserCfgCallBack_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_SLQSRegisterIMSIndication_t](#)
- struct [unpack_imsa_SLQSGetIMSRegStatus_t](#)
- struct [unpack_imsa_SLQSGetIMSAServiceStatus_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] • See pack_qmi_t 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

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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[IN/OUT] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[IN/OUT] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed

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_SLQSRegisterIMSASIndication (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_imsa_SLQSRegisterIMSASIndication_t * reqParam)

Function to pack Register IMSA Indication command

Parameters

<i>pCtx</i>	[OUT] <ul style="list-style-type: none"> See pack_qmi_t for more information
<i>pReqBuf</i>	[IN/OUT] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[IN/OUT] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[IN] <ul style="list-style-type: none"> See pack_imsa_SLQSRegisterIMSASIndication_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.8.3.4 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">Response from modem
<i>respLen</i>	[IN] <ul style="list-style-type: none">Length of pResp from modem
<i>pOutput</i>	[OUT] <ul style="list-style-type: none">See unpack_imsa_SLQSGetIMSARegStatus_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.8.3.5 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] <ul style="list-style-type: none">Response from modem
<i>respLen</i>	[IN] <ul style="list-style-type: none">Length of pResp from modem
<i>pOutput</i>	[OUT] <ul style="list-style-type: none">See unpack_imsa_SLQSGetIMSAServiceStatus_t 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 unpack_imsa_SLQSRegisterIMSAIndication_t 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` SetPrefRspFromModem)
- 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 lBlockSize)
- 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"> • full path of the file
<i>imagemask</i>	<ul style="list-style-type: none"> • Bitmask provides type of file <ul style="list-style-type: none"> – bit0 - cwe – bit1 - nvu – bit2 - spk

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

9.9.3.4 typedef struct _litefw_FirmwareInfo_litefw_FirmwareInfo

This structure contains information of the provided firmware file

Parameters

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

<i>szCarrier-Priversion_str</i>	<ul style="list-style-type: none"> Carrier PRI Version String.
---------------------------------	---

9.9.3.5 typedef struct _litefw_FirmwarePartNo _litefw_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"> Part Number of the provided firmware image
---------------------	--

9.9.3.6 typedef void(* litefwlogger)(uint8_t lvl, const char *buff)

This Custom Log prototype.

Parameters

in	<i>lvl</i>	<ul style="list-style-type: none"> Log level.
in	<i>buff</i>	<ul style="list-style-type: none"> Log String.

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"> See litefw_FirmwareInfo for more information.
out	<i>pack</i>	<ul style="list-style-type: none"> See pack_fms_SetImagesPreference_t for more information.

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>SetPrefRspFromModem</i>	<ul style="list-style-type: none"> See unpack_fms_SetImagesPreference_t for more information.
----	----------------------------	--

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	info	<ul style="list-style-type: none"> • See litefw_FirmwareInfo for more information.
----	------	---

Returns

0 on success, -1 error value otherwise

9.9.5.4 unsigned int litefw_DownloadFW (char * plImagePath, char * szTTYPath, int iFWImageType, int image_mask, int iModelFamily)

This API Download Firmware.

Parameters

in	plImagePath	<ul style="list-style-type: none"> • Firmware Folder Path.
in	szTTYPath	<ul style="list-style-type: none"> • QDL Device Path.
in	iFWImageType	<ul style="list-style-type: none"> • Firmware Type. • See litefw_Fw_Type
in	image_mask	<ul style="list-style-type: none"> • Image Mask. <ul style="list-style-type: none"> – IMG_MASK_MDM IMG_MASK_PRI
in	iModelFamily	<ul style="list-style-type: none"> • Modem Family. • See litefw_Models

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 Firmware Parameters From Path.

Parameters

in	<i>plmagePath</i>	<ul style="list-style-type: none">• Firmware Folder Path.
out	<i>info</i>	<ul style="list-style-type: none">• See litefw_FirmwareInfo for more information.

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 Extrac Firmware Part Number From Path.

Parameters

in	<i>plmagePath</i>	<ul style="list-style-type: none">• Firmware Folder Path.
out	<i>partno</i>	<ul style="list-style-type: none">• See litefw_FirmwarePartNo for more information.

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">• See litefw_FirmwareInfo for more information.
----	---------------	---

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 Family from a model string.

Parameters

<i>in</i>	<i>pModelString</i>	<ul style="list-style-type: none">Model String.
-----------	---------------------	---

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 download 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

in	<i>log_en</i>	
		<ul style="list-style-type: none">• Logs enable/disable flag.

9.9.5.13 int litefw_set_log_func (litefwlogger * *func*)

This API Set Custom Log function.

Parameters

in	<i>func</i>	
		<ul style="list-style-type: none">• See litefwlogger for more information.

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">• QDL Download Mode.<ul style="list-style-type: none">– litefw_QDL_MODEs

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">• litefw_QDL_FLOW_CONTROLS

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"> Firmware Read Block Size.
----	-------------------	---

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"> file path or directory of the firmware image
out	<i>info</i>	<ul style="list-style-type: none"> See litefw_FirmwareFileInfo, application should allocate the memory for the struct array, this API will fill in the details in the provided storage
out	<i>pNumOfItems</i>	<ul style="list-style-type: none"> number of litefw_FirmwareFileInfo

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"> QDL Device Path.
----	------------------	--

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"> • QDL Device Path.
----	------------------	--

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](#)

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](#)

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)

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 LOCEVENTMASKINJECTTIMEREQ 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_SLQSLOCGetServer_t**

9.10.2.3 **typedef unpack_result_t unpack_loc_SLQSLOCInjectPosition_t**

9.10.2.4 **typedef unpack_result_t unpack_loc_SLQSLOCInjectSensorData_t**

9.10.2.5 **typedef unpack_result_t unpack_loc_SLQSLOCInjectUTCTime_t**

9.10.2.6 **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_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

9.10.4.4 `int pack_loc_SetOperationMode (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetOperationMode_t * reqArg)`

Set Operation Mode pack

Parameters

in, 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_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

9.10.4.6 `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

9.10.4.7 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

9.10.4.8 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

9.10.4.9 `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

9.10.4.10 `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

9.10.4.11 `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

9.10.4.12 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

9.10.4.13 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

9.10.4.14 `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

9.10.4.15 `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

9.10.4.16 `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

9.10.4.17 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

9.10.4.18 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

9.10.4.19 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

9.10.4.20 `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

9.10.4.21 `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

9.10.4.22 `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

9.10.4.23 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

9.10.4.24 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

9.10.4.25 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

9.10.4.26 `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

9.10.4.27 `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

9.10.4.28 `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

9.10.4.29 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

9.10.4.30 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

9.10.4.31 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

9.10.4.32 `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

9.10.4.33 `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

9.10.4.34 `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

9.10.4.35 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

9.10.4.36 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

9.10.4.37 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

9.10.4.38 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

9.10.4.39 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

9.10.4.40 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

9.10.4.41 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

9.10.4.42 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

9.10.4.43 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

9.10.4.44 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

9.10.4.45 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

9.10.4.46 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 [RFBandInfoElements](#)
- struct [unpack_nas_GetRFInfo_t](#)
- struct [cdmaSSInfo](#)
- struct [hdrSSInfo](#)
- struct [lteSSInfo](#)
- struct [tdscdmaSigInfoExt](#)
- struct [unpack_nas_SLQSNasGetSigInfo_t](#)
- struct [unpack_nas_SLQSNasSigInfoCallback_ind_t](#)
- struct [unpack_nas_GetHomeNetwork_t](#)
- struct [nas_SrvStatusInfo](#)
- struct [nas_GSMSrvStatusInfo](#)
- struct [nas_sysInfoCommon](#)
- struct [nas_CDMASysInfo](#)
- struct [nas_HDRSysInfo](#)
- struct [nas_GSMSysInfo](#)
- struct [nas_WCDMASysInfo](#)
- struct [nas_LTESysInfo](#)
- struct [nas_AddCDMASysInfo](#)
- struct [nas_AddSysInfo](#)
- struct [nas_CallBarringSysInfo](#)
- struct [nas_LteCiotOpModeTlv](#)
- struct [nas_NR5GSerStatTlv](#)
- struct [nas_NR5GSystemInfoTlv](#)
- struct [nas_NR5GCellStatusTlv](#)
- struct [unpack_nas_SLQSGetSysInfo_t](#)
- struct [unpack_nas_SLQSSysInfoCallback_ind_t](#)
- struct [unpack_nas_GetServingNetwork_t](#)
- struct [unpack_nas_GetServingNetworkCapabilities_t](#)
- struct [nas_QmiNas3GppNetworkInfo](#)
- struct [nas_QmiNas3GppNetworkRAT](#)

- struct [nas_QmisNasPcsDigit](#)
- struct [nas_QmisNasSlqsNasPCICellInfo](#)
- struct [nas_QmisNasSlqsNasPCIIInfo](#)
- struct [nas_lteOpModeTlv](#)
- struct [unpack_nas_PerformNetworkScan_t](#)
- struct [unpack_nas_SLQSSwiGetLteCQI_t](#)
- struct [nas_CommInfo](#)
- struct [nas_LTEInfo](#)
- struct [unpack_nas_SLQSNasSwiModemStatus_t](#)
- struct [nas_servSystem](#)
- struct [nas_dataSrvCapabilities](#)
- struct [nas_currentPLMN](#)
- struct [nas_roamIndList](#)
- struct [nas_qaQmi3Gpp2TimeZone](#)
- struct [nas_detailSvcInfo](#)
- struct [nas_CDMASysInfoExt](#)
- struct [nas_callBarStatus](#)
- struct [unpack_nas_SLQSGetServingSystem_t](#)
- struct [nas_rxSignalStrengthListElement](#)
- struct [nas_ecioListElement](#)
- struct [nas_errorRateListElement](#)
- struct [nas_rsrqInformation](#)
- struct [nas_lteSnrinformation](#)
- struct [nas_lteRsrpinformation](#)
- struct [unpack_nas_SLQSGetSignalStrength_t](#)
- struct [nas_SLQSSignalStrengthsIndReq](#)
- struct [pack_nas_SLQSSetSignalStrengthsCallback_t](#)
- struct [nas_SLQSSignalStrengthsInformation](#)
- struct [nas_RejectReasonTlv](#)
- struct [nas_SignalStrengthTlv](#)
- struct [nas_RFInfoTlv](#)
- struct [nas_SLQSSignalStrengthsTlv](#)
- struct [unpack_nas_SetEventReportInd_t](#)
- struct [unpack_nas_GetCDMANetworkParameters_t](#)
- struct [unpack_nas_GetANAAAAAuthenticationStatus_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_TDSCDMARSCPTthresh](#)
- struct [nas_TDSCDMARSSIThresh](#)
- struct [nas_TDSCDMAECIOThresh](#)
- struct [nas_TDSCDMASINRThresh](#)

- struct [pack_nas_SLQSNasConfigSigInfo2_t](#)
- struct [unpack_nas_SetDataCapabilitiesCallback_ind_t](#)
- struct [unpack_nas_GetNetworkPreference_t](#)
- struct [pack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetRoamingIndicatorCallback_ind_t](#)
- struct [NAServingSystemInfo](#)
- struct [unpack_nas_SetServingSystemCallback_ind_t](#)
- struct [NASPhyCaAggScellIndType](#)
- struct [NASPhyCaAggScellIDBw](#)
- struct [NASPhyCaAggScellInfo](#)
- struct [NASPhyCaAggPcellInfo](#)
- struct [NASPhyCaAggScellIndex](#)
- struct [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_timeInfo](#)
- struct [unpack_nas_SLQSGetNetworkTime_t](#)
- struct [nas_UniversalTime](#)
- struct [unpack_nas_SLQSNasNetworkTimeCallback_ind_t](#)
- struct [nas_PhyCaAggScellIndType](#)
- struct [nas_PhyCaAggScellIDBw](#)
- struct [nas_PhyCaAggScellInfo](#)
- struct [nas_PhyCaAggPcellInfo](#)
- struct [nas_PhyCaAggScellIndex](#)
- 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 [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_TDSCDMASINRCONFThresh](#)
- 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_SLQSSwiRankIndicatorCallback_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 [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](#)

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_SCELL_ARRRY_SIZE](#) 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_SLQSSetSignalStrengthsCallback_t](#)
- typedef [unpack_result_t](#) [unpack_nas_SetRFInfoCallback_t](#)
- typedef [unpack_result_t](#) [unpack_nas_SetLURRejectCallback_t](#)
- typedef [unpack_result_t](#) [unpack_nas_SetACCOLC_t](#)
- typedef [unpack_result_t](#) [unpack_nas_SLQSNasConfigSigInfo2_t](#)
- typedef [unpack_result_t](#) [unpack_nas_SLQSInitiateNetworkRegistration_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_CELL_STATE {`
 `eLITEQMI_NAS_LTE_CPHY_CELL_STATE_DECONFIGURED =0x00,`
 `eLITEQMI_NAS_LTE_CPHY_CELL_STATE_CONFIGURED_DEACTIVATED =0x01,`
 `eLITEQMI_NAS_LTE_CPHY_CELL_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_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 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_SLQSNasSysInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSysInfoCallback_ind_t *pOutput)
- int pack_nas_GetServingNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t *pOutput)
- int pack_nas_GetServingNetworkCapabilities (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetworkCapabilities (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t *pOutput)

- int [pack_nas_PerformNetworkScan](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_PerformNetworkScan](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_PerformNetworkScan_t](#) *pOutput)
- int [pack_nas_SLQSSwiGetLteCQI](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSSwiGetLteCQI](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSwiGetLteCQI_t](#) *pOutput)
- int [pack_nas_SLQSNasSmiModemStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSNasSmiModemStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSmiModemStatus_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_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_SetLURejectCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int [unpack_nas_SetLURejectCallback](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetLURejectCallback_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_GetANAAAAAuthenticationStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetANAAAAAuthenticationStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetANAAAAAuthenticationStatus_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 pack_nas_SLQSGetNetworkTime](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_nas_SLQSGetNetworkTime](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetNetworkTime_t](#) *pOutput)
- [int unpack_nas_SLQSNasNetworkTimeCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasNetworkTimeCallBack_ind_t](#) *pOutput)
- [int unpack_nas_SetNasLTECphyCalndCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetNasLTECphyCalndCallback_ind_t](#) *pOutput)
- [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_SLQSNASSwiGetChannelLock_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_SLQSNASGeteDRXParams_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_SLQSNASGeteDRXParamsExt_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_SLQSNASSeteDRXParams_t](#) *pOutput)
- int [unpack_nas_SLQSSwiHDRPersonalityCallback_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t](#) *pOutput)
- int [unpack_nas_SLQSSwiRandIndicatorCallback_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSwiRandIndicatorCallback_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_SLQSGetSysSelectionPrefExt_t](#) *pOutput)
- 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_SLQSNasEdrxChangeInfoCallback_Ind_t](#) *pOutput)
- int [unpack_nas_SLQSNasNetworkRejectCallback_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasNetworkRejectCallback_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_SLQSNASGetForbiddenNetworks_t](#) *pOutput)

9.11.1 Macro Definition Documentation

9.11.1.1 `#define LITE_NW_SCAN_LTE_OP_MODE_MAX_LENGTH 255`

9.11.1.2 `#define NAS_IMSI_M_S1_LENGTH 7`

9.11.1.3 `#define NAS_IMSI_M_S2_LENGTH 3`

9.11.1.4 `#define NAS_MAX_DESCRIPTION_LENGTH 255`

9.11.1.5 `#define NAS_MAX_NUM_NETWORKS 30`

9.11.1.6 `#define NAS_MAX_PHY_CA_AGG_SELL_ARRRY_SIZE 255`

9.11.1.7 `#define NAS_MAX_SCC_RX_INFO_INSTANCES 255`

9.11.1.8 `#define NAS_MAX_SLQS_NAS_PCI_INFO_LENGTH 255`

9.11.1.9 `#define NAS_MAX_SLQS_NAS_PCI_INFO_PLMN_LENGTH 255`

9.11.1.10 `#define NAS_MCC_MNC_INST_LENGTH 255`

9.11.1.11 `#define NAS_NAM_NAME_LENGTH 12`

9.11.1.12 `#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048`

9.11.1.13 `#define NAS_PLMN_LENGTH 3`

9.11.1.14 `#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_SLQSNASSeteDRXParams_t
- 9.11.2.13 typedef unpack_result_t unpack_nas_SLQSNasSwiIndicationRegister_t
- 9.11.2.14 typedef unpack_result_t unpack_nas_SLQSNASSwiSetChannelLock_t
- 9.11.2.15 typedef unpack_result_t unpack_nas_SLQSSetBandPreference_t
- 9.11.2.16 typedef unpack_result_t unpack_nas_SLQSSetSignalStrengthsCallback_t
- 9.11.2.17 typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPref_t
- 9.11.2.18 typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPrefExt_t
- 9.11.2.19 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_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	<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

9.11.4.16 `int pack_nas_SetLURRejectCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable)`

Pack set registration reject 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

9.11.4.17 `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_SLQSConfigSigInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSConfigSigInfo_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_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

9.11.4.22 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

9.11.4.23 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

9.11.4.24 int pack_nas_SLQSGetPLMNName (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SLQSGetPLMNName_t * *reqArg*)

Pack get operator name for specifced 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

9.11.4.25 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

9.11.4.26 `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 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

9.11.4.27 `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

9.11.4.28 `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

9.11.4.29 `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

9.11.4.30 `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

9.11.4.31 `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

9.11.4.32 `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

9.11.4.33 `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

9.11.4.34 `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

9.11.4.35 `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

9.11.4.36 `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

9.11.4.37 `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

9.11.4.38 `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

9.11.4.39 `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

9.11.4.40 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

9.11.4.41 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

9.11.4.42 `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

9.11.4.43 `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

9.11.4.44 `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

9.11.4.45 `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

9.11.4.46 `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

9.11.4.47 `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

9.11.4.48 `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

9.11.4.49 `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

9.11.4.50 `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

9.11.4.51 `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

9.11.4.52 `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

9.11.4.53 `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

9.11.4.54 `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

9.11.4.55 `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

9.11.4.56 `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

9.11.4.57 `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

9.11.4.58 `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

9.11.4.59 `int unpack_nas_GetANAAAAAuthenticationStatus (uint8_t * pResp, uint16_t respLen,
unpack_nas_GetANAAAAAuthenticationStatus_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

9.11.4.60 `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

9.11.4.61 `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

9.11.4.62 `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

9.11.4.63 `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

9.11.4.64 `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

9.11.4.65 `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

9.11.4.66 `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

9.11.4.67 `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

9.11.4.68 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

9.11.4.69 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

9.11.4.70 `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

9.11.4.71 `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

9.11.4.72 `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

9.11.4.73 `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

9.11.4.74 `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

9.11.4.75 `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

9.11.4.76 `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

9.11.4.77 `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

9.11.4.78 `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

9.11.4.79 `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

9.11.4.80 `int unpack_nas_SLQSCfgSigInfo (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSCfgSigInfo_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

9.11.4.81 `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

9.11.4.82 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

9.11.4.83 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

9.11.4.84 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

9.11.4.85 `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

9.11.4.86 `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

9.11.4.87 `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

9.11.4.88 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

9.11.4.89 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

9.11.4.90 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

9.11.4.91 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

9.11.4.92 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

9.11.4.93 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

9.11.4.94 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

9.11.4.95 `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

9.11.4.96 `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

9.11.4.97 `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

9.11.4.98 `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

9.11.4.99 `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

9.11.4.100 `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

9.11.4.101 `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

9.11.4.102 `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

9.11.4.103 `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

9.11.4.104 `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

9.11.4.105 `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

9.11.4.106 `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

9.11.4.107 `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

9.11.4.108 `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

9.11.4.109 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

9.11.4.110 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

9.11.4.111 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

9.11.4.112 `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

9.11.4.113 `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

9.11.4.114 `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

9.11.4.115 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

9.11.4.116 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

9.11.4.117 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

9.11.4.118 `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

9.11.4.119 `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

9.11.4.120 `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

9.11.4.121 `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

9.11.4.122 `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

9.11.4.123 `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

9.11.4.124 `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

9.11.4.125 `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

9.11.4.126 `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

9.11.4.127 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

9.11.4.128 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

9.11.4.129 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 unpack_nas_GetCDMANetworkParameters_t param valid. <ul style="list-style-type: none"> • 0 - SCI • 1 - SCM • 2 - RegHomeSID, RegForeignSID and RegForeignNID • 3 - ForceRev0 • 4 - CustomSCP, Protocol ,Broadcast and Application • 5 - Roaming

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.11.4.130 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 unpack_nas_SLQSGetServingSystem_t param valid. <ul style="list-style-type: none"> • 0 - RoamIndicatorVal • 1 - DataSrvCapabilities • 2 - CurrentPLMN • 3 - SystemID and NetworkID • 4 - BaseStationID, BaseStationLatitude and BasestationLongitude • 5 - RoamingIndicatorList • 6 - DefaultRoamInd • 7 - Gpp2TimeZone • 8 - CDMA_P_Rev • 9 - GppTimeZone • 10 - GppNetworkDSTAdjustment • 11 - Lac • 12 - CellID • 13 - ConcSvcInfo • 14 - PRLInd • 15 - DTMIInd • 16 - DetailedSvcInfo • 17 - CDMASystemInfoExt • 18 - HdrPersonality • 19 - TrackAreaCode • 20 - CallBarStatus

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.11.4.131 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 unpack_nas_SLQSGetSignalStrength_t param valid. <ul style="list-style-type: none"> • 0 - rxSignalStrengthList and rxSignalStrengthListLen • 1 - ecioList and ecioListLen • 2 - lo • 3 - sinr • 4 - errorRateListLen, errorRateList • 5 - rsrqInfo • 6 - ltesnr • 7 - ltersrp

Returns

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.11.4.132 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 unpack_nas_SLQSNasGetSigInfo_t param valid. <ul style="list-style-type: none"> • 0 - CDMASSTInfo • 1 - HDRSSInfo. • 2 - GSMSSInfo. • 3 - WCDMASSInfo. • 4 - LTESSTInfo.

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_GetPDSDefaults_t](#)
- struct [pack_pds_SetPDSDefaults_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_SetPDSDefaults_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_GetPDSDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_pds_GetPDSDefaults (uint8_t *pResp, uint16_t respLen, unpack_pds_GetPDSDefaults_t *pOutput)`
- `int pack_pds_SetPDSDefaults (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetPDSDefaults_t *pReqParam)`
- `int unpack_pds_SetPDSDefaults (uint8_t *pResp, uint16_t respLen, unpack_pds_SetPDSDefaults_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_StopPDSTrackingSession_t *pOutput)`
- `int pack_pds_PDSInjectTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_PDSInjectTimeReference_t *pReqParam)`
- `int unpack_pds_PDSInjectTimeReference (uint8_t *pResp, uint16_t respLen, unpack_pds_PDSInjectTimeReference_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_GetXTRAAutomaticDownload_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_SetXTRAAutomaticDownload_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 *pOutput)`
- `int pack_pds_SetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRANetwork_t *pReqParam)`
- `int unpack_pds_SetXTRANetwork (uint8_t *pResp, uint16_t respLen, unpack_pds_SetXTRANetwork_t *pOutput)`
- `int pack_pds_GetXTRAVality (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_pds_GetXTRAVality (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAVality_t *pOutput)`
- `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_ForceXTRADownload_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_GetServiceAutomaticTracking_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_SetServiceAutomaticTracking_t](#) *pOutput)
- int [pack_pds_ResetPDSData](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_pds_ResetPDSData_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_SLQSSetAGPSConfig_t](#) *pReqParam)
- int [unpack_pds_SLQSSetAGPSConfig](#) (uint8_t *pResp, uint16_t respLen, [unpack_pds_SLQSSetAGPSConfig_t](#) *pOutput)
- int [pack_pds_SLQSPDSInjectAbsoluteTimeReference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_pds_SLQSPDSInjectAbsoluteTimeReference_t](#) *pReqParam)
- int [unpack_pds_SLQSPDSInjectAbsoluteTimeReference](#) (uint8_t *pResp, uint16_t respLen, [unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t](#) *pOutput)
- int [pack_pds_SLQSGetAGPSConfig](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_pds_SLQSGetAGPSConfig_t](#) *pReqParam)
- int [unpack_pds_SLQSGetAGPSConfig](#) (uint8_t *pResp, uint16_t respLen, [unpack_pds_SLQSGetAGPSConfig_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_SLQSPDSInjectPositionData_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_SLQSPDSDeterminePosition_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_ResetPDSData_t](#)

9.12.1.4 typedef [unpack_result_t](#) [unpack_pds_SetEventReportCallback_t](#)

9.12.1.5 typedef [unpack_result_t](#) [unpack_pds_SetPDSDefaults_t](#)

9.12.1.6 typedef [unpack_result_t](#) [unpack_pds_SetPDSState_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_GetXTRAVality (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_SLQSGetGPSSStateInfo (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_SLQSPDSDeterminePosition (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	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

9.12.2.24 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	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

9.12.2.25 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_SetPDSDefaults (uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetPDSDefaults_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_SLQSGetGPSState-Info_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: 1850-19195; DL: 1930-1995; E-UTRA Operating Band 25
- 149 - TDD LTE UL: 2496-2690; DL: 2496-2690; E-UTRA Operating Band 41
- 150 - TDD LTE UL: 3400-3600; DL: 3400-3600; E-UTRA Operating Band 42
- 151 - TDD LTE UL: 3600-3800; DL: 3600-3800; E-UTRA Operating Band 43
- 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_OUTGOINGINTEXTOTHOME - Outgoing external to home
- 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming
- 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred
- 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI_VOICE_REASON_CALLWAITING - Call waiting
- 0x10 - VOICE_CC_SUPS_RESULT_REASON_CLIP - CLIP
- 0x11 - VOICE_CC_SUPS_RESULT_REASON_CLIR - CLIR
- 0x12 - VOICE_CC_SUPS_RESULT_REASON_COLP - COLP
- 0x13 - VOICE_CC_SUPS_RESULT_REASON_COLR - COLR
- 0x14 - VOICE_CC_SUPS_RESULT_REASON_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

9.15 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 SMDCP(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 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRRC(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRRC(LTE Radio Resource Control) CONN REL RLF
- 1063 - LRRRC(LTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRRC(LTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRRC(LTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRRC(LTE Radio Resource Control) CONN REL SIB READ ERROR
- 1067 - DETACH WITH REATTACH LTE NW DETACH
- 1068 - DETACH WITH OUT REATTACH LTE NW DETACH
- 1069 - ESM(EPS Session Management) PROC TIME OUT
- 1070 - MESSAGE EXCEED MAX L2 LIMIT
- 1500 - CD GEN OR BUSY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of general or network busy
- 1501 - CD BILL OR AUTH, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of billing failure or authentication failure
- 1502 - CHG HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is a change to HDR system due to redirection or PRL not preferred

- 1503 - EXIT HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device exited HDR due to redirection or PRL not preferred
- 1504 - HDR NO SESSION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have a HDR session
- 1505 - HDR ORIG DURING GPS FIX, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since it is ending an HDR call origination in favor of a GPS fix
- 1506 - HDR CS TIMEOUT, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since connection setup on HDR system timed out
- 1507 - HDR RELEASED BY CM, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when it wants to release a HDR call so a 1X call can continue
- 1508 - COLLOC ACQ FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when device failed to acquire co-located HDR for origination
- 1509 - OTASP COMMIT IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since an OTASP commit is in progress
- 1510 - NO HYBR HDR SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device has no Hybrid HDR service
- 1511 - HDR NO LOCK GRANTED, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module could not get the RF lock
- 1512 - HOLD OTHER IN PROG, this error code is returned when data call is brought down by CM(Call Manager) because DBM or SMS is in progress
- 1513 - HDR FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module released the call due to fade
- 1514 - HDR ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to HDR system Access Failure
- 2000 - CLIENT END, this error code is returned when client ends the data call
- 2001 - NO SRV, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device has no service
- 2002 - FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device lost the system due to fade
- 2003 - REL NORMAL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with no reason
- 2004 - ACC IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access attempt already in progress
- 2005 - ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access Failure
- 2006 - REDIR OR HANDOFF, this error code is returned when data call is brought down because device is in the process of redirecting/handling 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 SNDP 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 qaGobiApiTableRadiolInterfaces.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_DATACIRCUITSYNC
- 0X20 - CLASS_DATACIRCUITASYNC
- 0X40 - CLASS_PACKETACCESS
- 0X80 - CLASS_PADACCESS

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 TND5) - 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

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_SLQSQosSviReadDataStats_t](#)
- struct [unpack_QosFlowStat_t](#)
- struct [unpack_qos_SLQSQosSviReadDataStats_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_BindDataPortSIODataPort_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_SLQSQosSviReadApnExtraParams \(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, \[pack_qos_SLQSQosSviReadApnExtraParams_t\]\(#\) reqParam\)](#)
- [int unpack_qos_SLQSQosSviReadApnExtraParams \(uint8_t *pResp, uint16_t respLen, \[unpack_qos_SLQSQosSviReadApnExtraParams_t\]\(#\) *pOutput\)](#)
- [int pack_qos_SLQSQosSviReadDataStats \(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, \[pack_qos_SLQSQosSviReadDataStats_t\]\(#\) reqParam\)](#)
- [int unpack_qos_SLQSQosSviReadDataStats \(uint8_t *pResp, uint16_t respLen, \[unpack_qos_SLQSQosSviReadDataStats_t\]\(#\) *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"> • See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
in	<i>req</i>	<ul style="list-style-type: none"> • See pack_qos_BindDataPort_t 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: 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"> • See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed

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"> • See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> • See pack_qos_SLQSQosSwiReadApnExtraParams_t 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: 2 seconds
- PDN Specific: Yes

9.29.2.4 `int pack_qos_SLQSQosSwiReadDataStats (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSQosSwiReadDataStats_t reqParam)`

Function to pack QMI command to query APN data statistics This maps to SLQSQosSwiReadDataStats

Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> • See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> • See pack_qos_SLQSQosSwiReadDataStats_t 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: 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"> • See pack_qmi_t for more information
-----	-------------	---

in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> • See pack_qos_SLQSSetQosEventCallback_t 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: 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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> • See unpack_qos_BindDataPort_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.2.7 `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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> • See unpack_qos_SLQSQosGetNetworkStatus_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.2.8 `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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> • See unpack_qos_SLQSQosSwiReadApnExtraParams_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.2.9 `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"> • Response from modem
----	--------------	---

in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> See unpack_qos_SLQSQosSwiReadDataStats_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.2.10 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> See unpack_qos_SLQSSetQosEventCallback_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.2.11 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem

out	<i>pOutput</i>	<ul style="list-style-type: none"> • See unpack_qos_SLQSSetQosEventCallback_ind_t 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

- 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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> • See unpack_qos_SLQSSetQosNWStatusCallback_ind_t 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

- 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> See unpack_qos_SLQSSetQosPriEventCallback_ind_t 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

- 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pOutput</i>	<ul style="list-style-type: none"> See unpack_qos_SLQSSetQosStatusCallback_ind_t 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

- 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_sendAsyncsmsParams](#)
- 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](#) *Output)
- [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 eqmiCbkJetStatus`

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	<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

9.32.3.10 `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	<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

9.32.3.11 `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 [UnpackSwiAvmsEventReportWAMSParmChange](#)
- 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_SLQSAVMSStartSession_t](#) reqParam)
- int [unpack_swiavms_SLQSAVMSStartSession](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_swiavms_SLQSAVMSStartSession_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"> See pack_qmi_t for more information
in, out	pReqBuf	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
in, out	pLen	<ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed

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"> See pack_qmi_t for more information
in, out	pReqBuf	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
in, out	pLen	<ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed

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_swia_vms_SLQSAVMSSendSelection (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swia_vms_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"> See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> See pack_swia_vms_SLQSAVMSSendSelection_t 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: 2 seconds

9.34.3.4 `int pack_swia_vms_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"> See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes

<i>in, out</i>	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
----------------	-------------	---

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"> • See pack_qmi_t for more information
<i>in, out</i>	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
<i>in, out</i>	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed

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"> See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> See pack_swiaavms_SLQSAVMSSetSettings_t 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: 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"> See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> See pack_swiaavms_SLQSAVMSSetSettings_v2_t 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: 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"> See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> On input, size of pReqBuf On ouptut, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> See pack_swiavms_SLQSAVMSSetSettings_t 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: 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"> See pack_qmi_t for more information
-----	-------------	---

in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> • See pack_swiaavms_SLQSAVMSSStartSession_t 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: 2 seconds

9.34.3.10 `int pack_swiaavms_SLQSAVMSSStopSession (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaavms_SLQSAVMSSStopSession_t reqParam)`

Function to pack cancel AVMS session command This maps to SLQSAVMSSStopSession

Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> • See pack_qmi_t for more information
in, out	<i>pReqBuf</i>	<ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
in, out	<i>pLen</i>	<ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
in	<i>reqParam</i>	<ul style="list-style-type: none"> • See pack_swiaavms_SLQSAVMSSStopSession_t 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: 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> See unpack_swiavms_SLQSAVMSEventReportInd_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.12 `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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> See unpack_swiavms_SLQSAVMSSetSettings_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.13 `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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiaavms_SLQSAVMSSetSettings_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.14 `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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiaavms_SLQSAVMSendSelection_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.15 `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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiavms_SLQSAVMSSessionGetInfo_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.16 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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiavms_SLQSAvmsSetEventReport_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.17 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"> • Response from modem
----	--------------	---

in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> See unpack_swiaavms_SLQSAVMSSetSettings_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.18 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> See unpack_swiaavms_SLQSAVMSSetSettings_v2_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.19 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"> Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> Length of pResp from modem

out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiavms_SLQSAVMSSession_t for more information
-----	------------------	---

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.20 int unpack_swiavms_SLQSAVMSSession (uint8_t * *pResp*, uint16_t *respLen*,
unpack_swiavms_SLQSAVMSSession_t * *pResponse*)

Function to pack stop AVMS session command This maps to SLQSAVMSSession

Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiavms_SLQSAVMSSession_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.34.3.21 int unpack_swiavms_SLQSAVMSSession_avc2 (uint8_t * *pResp*, uint16_t *respLen*,
unpack_swiavms_SLQSAVMSSession_t * *pResponse*)

Function to pack stop AVMS session command This maps to SLQSAVMSSession 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"> • Response from modem
in	<i>respLen</i>	<ul style="list-style-type: none"> • Length of pResp from modem
out	<i>pResponse</i>	<ul style="list-style-type: none"> • See unpack_swiavms_SLQSAVMSSession_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.35 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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed

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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[in] <ul style="list-style-type: none"> See pack_swioma_SLQSOMADMCancelSession_t 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

9.38.3.3 `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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[in] <ul style="list-style-type: none"> See pack_swima_SLQSOMADMGetSessionInfo_t 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

9.38.3.4 `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"> See pack_qmi_t for more information
-------------	--

<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed

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"> • See pack_qmi_t for more information
<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
<i>reqParam</i>	[in] <ul style="list-style-type: none"> • See pack_swioma_SLQSOMADMSendSelection_t 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

9.38.3.6 `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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[in] <ul style="list-style-type: none"> See pack_swima_SLQSOMADMSetSettings_t 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

9.38.3.7 `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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[in,out] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[in,out] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[in] <ul style="list-style-type: none"> See pack_swima_SLQSOMADMStartSession_t 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

9.38.3.8 `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"> See unpack_swioma_SLQSOMADMAAlertCallback_ind_t for more information
----------------	---

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.38.3.10 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"> Response from modem
<i>respLen</i>	[in] <ul style="list-style-type: none"> Length of pResp from modem
<i>pOutput</i>	[out] <ul style="list-style-type: none"> response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.38.3.11 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"> Response from modem
<i>respLen</i>	[in] <ul style="list-style-type: none"> Length of pResp from modem
<i>pOutput</i>	[out] <ul style="list-style-type: none"> See unpack_swioma_SLQSOMADMGetSessionInfo_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.38.3.12 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] • Response from modem
<i>respLen</i>	[in] • Length of pResp from modem
<i>pOutput</i>	[out] • See unpack_swioma_SLQSOMADMGetSettings_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.38.3.13 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] • 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

9.38.3.14 `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 unpack_swima_SLQSOMADMStartSession_t 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 pack_qmi_t 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 pack_swima_SLQSOMADMCancelSessionExt_t 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 pack_qmi_t 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 Thu Nov 21 2019 08:29:34 for LinuxQMISDK-Lite by Doxygen	

<i>pLen</i>	[IN/OUT] <ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
-------------	--

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"> • See pack_qmi_t for more information
<i>pReqBuf</i>	[IN/OUT] <ul style="list-style-type: none"> • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
<i>pLen</i>	[IN/OUT] <ul style="list-style-type: none"> • On input, size of pReqBuf • On output, number of bytes actually packed
<i>reqParam</i>	[IN] <ul style="list-style-type: none"> • See pack_swioma_SLQSOMADMSendSelectionExt_t 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

9.39.3.4 `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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[IN/OUT] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[IN/OUT] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[IN] <ul style="list-style-type: none"> See pack_swima_SLQSOMADMSetSettings_t 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

9.39.3.5 `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"> See pack_qmi_t for more information
<i>pReqBuf</i>	[IN/OUT] <ul style="list-style-type: none"> Buffer for packed QMI command to be provided by the host application Minimum expected size is 2048 bytes
<i>pLen</i>	[IN/OUT] <ul style="list-style-type: none"> On input, size of pReqBuf On output, number of bytes actually packed
<i>reqParam</i>	[IN] <ul style="list-style-type: none"> See pack_swima_SLQSOMADMStartSessionExt_t 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

9.39.3.6 `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 unpack_swioma_SLQSOMADMGetSessionInfoExt_t 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}}

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 Function Documentation

9.40.2.1 `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

9.40.2.2 `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

9.40.2.3 `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
----	-------------	----------

9.40.2.4 `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_SLQSUIMLGetSlotsStatusV2_t](#)
- struct [pack_uim_SLQSUIMLSwitchSlot_t](#)
- struct [unpack_uim_SetUimSlotStatusChangeCallback_ind_t](#)
- struct [pack_uim_SLQSUIMLPowerUp_t](#)
- struct [pack_uim_SLQSUIMLPowerDown_t](#)
- struct [pack_uim_SLQSUIMLRefreshOK_t](#)
- struct [uim_registerRefresh](#)
- struct [pack_uim_SLQSUIMLRefreshRegister_t](#)
- struct [pack_uim_SLQSUIMLRefreshComplete_t](#)
- struct [uim_refreshevent](#)
- struct [pack_uim_SLQSUIMLRefreshGetLastEvent_t](#)
- struct [unpack_uim_SLQSUIMLRefreshGetLastEvent_t](#)
- struct [pack_uim_SLQSUIMLGetFileAttributes_t](#)
- struct [uim_fileAttributes](#)
- struct [unpack_uim_SLQSUIMLGetFileAttributes_t](#)
- struct [uim_depersonalizationInformation](#)
- struct [pack_uim_SLQSUIMLDepersonalization_t](#)
- struct [unpack_uim_SLQSUIMLDepersonalization_t](#)
- struct [uim_authenticationData](#)
- struct [pack_uim_SLQSUIMLAuthenticate_t](#)
- struct [uim_authenticateResult](#)
- struct [unpack_uim_SLQSUIMLAuthenticate_t](#)
- struct [uim_personalizationStatus](#)
- struct [pack_uim_SLQSUIMLGetConfiguration_t](#)
- struct [unpack_uim_SLQSUIMLGetConfiguration_t](#)
- struct [unpack_uim_SLQSUIMLRefreshCallback_Ind_t](#)

Macros

- [#define UIM_UINT8_MAX_STRING_SZ 255](#)
- [#define UIM_MAX_DESCRIPTION_LENGTH 255](#)
- [#define UIM_MAX_NO_OF_SLOTS 5](#)
- [#define UIM_MAX_NO_OF_APPLICATIONS 10](#)
- [#define MAX_NO_OF_SLOTS 5](#)
- [#define MAX_NO_OF_APPLICATIONS 10](#)
- [#define MAX_DESCRIPTION_LENGTH 255](#)
- [#define MAX_SLOTS_STATUS 255](#)
- [#define MAX_ICCID_LENGTH 255](#)
- [#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_SLQSUIMLSwitchSlot_t](#)
- typedef [unpack_result_t](#) [unpack_uim_SLQSUIMLPowerUp_t](#)
- typedef [unpack_result_t](#) [unpack_uim_SLQSUIMLPowerDown_t](#)
- typedef [unpack_result_t](#) [unpack_uim_SLQSUIMLReset_t](#)
- typedef [unpack_result_t](#) [unpack_uim_SLQSUIMLRefreshOK_t](#)
- typedef [unpack_result_t](#) [unpack_uim_SLQSUIMLRefreshRegister_t](#)
- typedef [unpack_result_t](#) [unpack_uim_SLQSUIMLRefreshComplete_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_SLQSUIRefreshRegister](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIRefreshRegister_t](#) *reqArg)
- int [unpack_uim_SLQSUIRefreshRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIRefreshRegister_t](#) *pOutput)
- int [pack_uim_SLQSUIRefreshComplete](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIRefreshComplete_t](#) *reqArg)
- int [unpack_uim_SLQSUIRefreshComplete](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIRefreshComplete_t](#) *pOutput)
- int [pack_uim_SLQSUIRefreshGetLastEvent](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIRefreshGetLastEvent_t](#) *reqArg)
- int [unpack_uim_SLQSUIRefreshGetLastEvent](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIRefreshGetLastEvent_t](#) *pOutput)
- int [pack_uim_SLQSUIGetFileAttributes](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIGetFileAttributes_t](#) *reqArg)
- int [unpack_uim_SLQSUIGetFileAttributes](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIGetFileAttributes_t](#) *pOutput)
- int [pack_uim_SLQSUIDepersonalization](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIDepersonalization_t](#) *reqArg)
- int [unpack_uim_SLQSUIDepersonalization](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIDepersonalization_t](#) *pOutput)
- int [pack_uim_SLQSUIAuthenticate](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIAuthenticate_t](#) *reqArg)
- int [unpack_uim_SLQSUIAuthenticate](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIAuthenticate_t](#) *pOutput)
- int [pack_uim_SLQSUIGetConfiguration](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIGetConfiguration_t](#) *reqArg)
- int [unpack_uim_SLQSUIGetConfiguration](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIGetConfiguration_t](#) *pOutput)
- int [unpack_uim_SLQSUIRefreshCallback_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIRefreshCallback_Ind_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_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_SLQSUIMDepersonalization (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMDepersonalization_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_SLQSUIGetFileAttributes (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIGetFileAttributes_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_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

9.43.3.11 `int pack_uim_SLQSUIMPowerDown (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIMPowerDown_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

9.43.3.12 `int pack_uim_SLQSUIMPowerUp (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIMPowerUp_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

9.43.3.13 `int pack_uim_SLQSUIMRefreshComplete (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIMRefreshComplete_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

9.43.3.14 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

9.43.3.15 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	<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_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	<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_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

9.43.3.18 int pack_uim_SLQSUISwitchSlot (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUISwitchSlot_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

9.43.3.19 `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

9.43.3.20 `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

9.43.3.21 `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

9.43.3.22 `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

9.43.3.23 `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

9.43.3.24 `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

9.43.3.25 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

9.43.3.26 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

9.43.3.27 int unpack_uim_SLQSUIAuthenticate (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIAuthenticate_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

9.43.3.28 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

9.43.3.29 int unpack_uim_SLQSUIEventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIEvent-
Register_t * *pOutput*)

UIM Status Change callback enable unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.43.3.30 `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

9.43.3.31 `int unpack_uim_SLQSUIGetFileAttributes (uint8_t * pResp, uint16_t respLen,
unpack_uim_SLQSUIGetFileAttributes_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

9.43.3.32 `int unpack_uim_SLQSUIGetSlotsStatus (uint8_t * pResp, uint16_t respLen,
unpack_uim_SLQSUIGetSlotsStatus_t * pOutput)`

get slot status unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.43.3.33 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

9.43.3.34 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

9.43.3.35 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

9.43.3.36 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

9.43.3.37 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

9.43.3.38 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

9.43.3.39 `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

9.43.3.40 `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

9.43.3.41 `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

9.43.3.42 int unpack_uim_SLQSUIMSetStatusChangeCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t * *pOutput*)

UIM Status Change indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

use pack_uim_SLQSUIMEventRegister to subscribe

9.43.3.43 int unpack_uim_SLQSUIMSwitchSlot (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMSwitchSlot_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

9.43.3.44 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

9.43.3.45 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

9.43.3.46 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_DATACIRCUITSYNC](#) = 0x10,
[LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC](#) = 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_SLQSOriinateUSSD](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_voice_SLQSOriinateUSSD_t](#) *reqArg)
- [int unpack_voice_SLQSOriinateUSSD](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_voice_SLQSOriinateUSSD_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	<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

9.44.6.11 `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	<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

9.44.6.12 `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	<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

9.44.6.25 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	<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

9.44.6.26 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_SLQSOriinateUSSD_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_SLQSVoiceSend-
Flash_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 [qmiWDSDataBearerTechnology](#)
- struct [unpack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [unpack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [pack_wds_SetDefaultProfileNum_t](#)
- struct [pack_wds_GetDefaultProfileNum_t](#)
- struct [unpack_wds_GetDefaultProfileNum_t](#)
- struct [wdsDhcpv4ProfileId](#)
- struct [wdsDhcpv4HwConfig](#)
- struct [wdsDhcpv4Option](#)
- struct [wdsDhcpv4OptionList](#)
- struct [pack_wds_SLQSSGetDHCPv4ClientConfig_t](#)
- struct [unpack_wds_SLQSSGetDHCPv4ClientConfig_t](#)
- struct [pack_wds_GetPacketStatistics_t](#)
- struct [unpack_wds_GetPacketStatistics_t](#)
- struct [unpack_wds_GetByteTotals_t](#)
- struct [unpack_wds_SLQSGetCurrentChannelRate_t](#)
- struct [unpack_wds_SLQSSGetLoopback_t](#)
- struct [pack_wds_SLQSSSetLoopback_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](#)

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_PCOID_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](#)

Enumerations

- enum [liteQmiDataBearerMasks](#) {
[QMI_LITE_WDS_CURRENT_CALL_DB_MASK](#) = 0x01,
[QMI_LITE_WDS_LAST_CALL_DB_MASK](#) = 0x02 }

Functions

- int [pack_wds_SLQSSStartDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSStartDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSSStartDataSession](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSStartDataSession_t](#) *pOutput)
- int [unpack_wds_SLQSSetPacketSrvStatusCallback](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetPacketSrvStatusCallback_t](#) *pOutput)
- int [pack_wds_SLQSSStopDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSStopDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSSStopDataSession](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSStopDataSession_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)

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_SLQSResetPacketStatics_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_SLQSStopDataSession_t`

9.45.5.15 `typedef unpack_result_t unpack_wds_SLQSWdsGoActive_t`

9.45.5.16 `typedef unpack_result_t unpack_wds_SLQSWdsGoDormant_t`

9.45.5.17 `typedef unpack_result_t unpack_wds_SLQSWdsSetEventReport_t`

9.45.5.18 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"> See LibpackProfile3GPP
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> See LibpackProfile3GPP2

9.45.5.19 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"> See LibpackProfile3GPP
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> See LibpackProfile3GPP2

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	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReqParam	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	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	reqParam	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_SLQSStartDataSession (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSStartDataSession_t * reqArg)`

Start data session

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>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_SLQSStopDataSession (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSStopDataSession_t * reqArg)`

stop data session pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>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_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

9.45.7.48 `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

9.45.7.49 `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

9.45.7.50 `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

9.45.7.51 `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

9.45.7.52 `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

9.45.7.53 `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

9.45.7.54 `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

9.45.7.55 `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

9.45.7.56 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

9.45.7.57 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

9.45.7.58 `int unpack_wds_GetDefaultProfileNum (uint8_t * pResp, uint16_t respLen, unpack_wds_GetDefaultProfileNum_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

9.45.7.59 `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

9.45.7.60 `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

9.45.7.61 `int unpack_wds_GetLastMobileIPError (uint8_t * pResp, uint16_t respLen, unpack_wds_GetLastMobileIP-Error_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

9.45.7.62 `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

9.45.7.63 `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

9.45.7.64 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

9.45.7.65 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

9.45.7.66 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

9.45.7.67 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

9.45.7.68 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

9.45.7.69 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

9.45.7.70 `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

9.45.7.71 `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

9.45.7.72 `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

9.45.7.73 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

9.45.7.74 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

9.45.7.75 int unpack_wds_SetMobileIPParameters (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SetMobileIP-Parameters_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

9.45.7.76 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

9.45.7.77 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

9.45.7.78 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

9.45.7.79 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

9.45.7.80 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

9.45.7.81 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

9.45.7.82 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

9.45.7.83 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

9.45.7.84 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

9.45.7.85 int unpack_wds_SLQSGetDUNCallInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetDUNCallInfo_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

9.45.7.86 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

9.45.7.87 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

9.45.7.88 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

9.45.7.89 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

9.45.7.90 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

9.45.7.91 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

9.45.7.92 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

9.45.7.93 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

9.45.7.94 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

9.45.7.95 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

9.45.7.96 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

9.45.7.97 `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

9.45.7.98 `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

9.45.7.99 `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

9.45.7.100 `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

9.45.7.101 `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

9.45.7.102 `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

9.45.7.103 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

9.45.7.104 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

9.45.7.105 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, 997](#)
- [_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, 986](#)
- [AAASState](#)
 - [unpack_wds_GetMobileIPProfile_t, 986](#)
- [ACT_CODE_MAX_SIZE](#)
 - [dms.h, 1128](#)
- [ALS](#)
 - [voice_getAllCallInformation, 1065](#)
- [AMSSString](#)
 - [unpack_dms_GetFirmwareRevision_t, 629](#)
 - [unpack_dms_GetFirmwareRevisions_t, 631](#)
- [AMTlv](#)
 - [unpack_ims_SLQSVolPCfgCallback_ind_t, 702](#)
- [AOATlv](#)
 - [unpack_ims_SLQSVolPCfgCallback_ind_t, 702](#)
- [AOPTlv](#)
 - [NASQmiCbkNasSystemSelPrefInd, 338](#)
- [APNName](#)
 - [unpack_wds_SLQSGetRuntimeSettings_t, 1007](#)
- [AWMTlv](#)
 - [unpack_ims_SLQSVolPCfgCallback_ind_t, 702](#)
- [AWOATlv](#)
 - [unpack_ims_SLQSVolPCfgCallback_ind_t, 702](#)
- [accelTemp](#)
 - [pack_loc_SLQSLOCInjectSensorData_t, 389](#)
- [acceleroData](#)
 - [pack_loc_SLQSLOCInjectSensorData_t, 389](#)
- [acceleroTimeSrc](#)
 - [pack_loc_SLQSLOCInjectSensorData_t, 389](#)
- [AccessMac](#)
 - [nas_protocolSubtypeElement, 272](#)
- [accolc](#)
 - [pack_nas_SetACCOLC_t, 395](#)
- [accuracy](#)
 - [pack_pds_SetPDSDefaults_t, 435](#)
- [ackIndicator](#)
 - [sMSTransferRouteMTMessageInfo, 568](#)
- [acqOrdeLen](#)
 - [nas_acqOrderPref, 168](#)
 - [nas_AcqOrderPrefTlv, 168](#)
- [AcqOrderLen](#)
 - [NASAcqOrderPrefTlv, 322](#)
- [AcqOrderPref](#)
 - [NASAcqOrderPrefTlv, 322](#)
- [acroamsetting](#)
 - [pack_wds_SetAutoconnect_t, 515](#)
- [acsetting](#)
 - [pack_wds_SetAutoconnect_t, 515](#)
- [actCode](#)
 - [pack_dms_ActivateAutomatic_t, 351](#)
- [ActSetCnt](#)
 - [nas_NetworkStat1x, 250](#)
- [ActSetPilotPN](#)
 - [nas_ActPilotPNElement, 169](#)
- [ActSetPilotPNStrength](#)
 - [nas_ActPilotPNElement, 169](#)
- [action](#)
 - [pack_nas_InitiateDomainAttach_t, 394](#)
- [activated_ind](#)
 - [sms_qaQmi3GPP2BroadcastCfgInfo, 557](#)
 - [sms_qaQmi3GPPBroadcastCfgInfo, 557](#)
- [activationState](#)
 - [pack_dms_SetActivationStatusCallback_t, 352](#)
- [activationStatus](#)
 - [dms_ActivationStatusTlv, 56](#)
- [ActivationStatusTlv](#)
 - [unpack_dms_SetEventReport_ind_t, 647](#)
- [activeBand](#)
 - [nas_RfBandInfoExtFormatElements, 281](#)
- [activeBandClass](#)
 - [nas_RFBandInfoElements, 280](#)
 - [nas_RFInfoTlv, 286](#)
 - [RFBandInfoElements, 544](#)
- [activeChannel](#)

- nas_RFBandInfoElements, 280
- nas_RfBandInfoExtFormatElements, 281
- nas_RFInfoTlv, 286
- RFBandInfoElements, 544
- activeInd
 - sms_messageWaitingInfoContent, 555
- ActiveStatus
 - voice_CLIPResp, 1055
 - voice_CLIRResp, 1056
 - voice_CNAPResp, 1057
 - voice_COLPResp, 1057
 - voice_COLRResp, 1058
- ActiveTechPref
 - unpack_nas_GetNetworkPreference_t, 750
- ActiveTimer
 - unpack_dms_PSMCfgChange_ind_t, 644
- activeTimer
 - dms_PSMActiveTimerTlv, 67
- ActiveTimerInd
 - dms_PSMActiveTimerIndTlv, 66
- addr
 - unpack_qos_IPv4Addr_t, 835
 - unpack_qos_IPv6Addr_t, 836
- address
 - loc_IPv4Info, 157
 - loc_IPv6Info, 158
 - loc_urlAddr, 165
 - unpack_wds_GetMobileIPProfile_t, 986
- addressSize
 - unpack_sms_GetSMSCAddress_t, 860
- aid
 - uim_refreshevent, 596
 - uim_sessionInformation, 599
 - uim_UIMSessionInformation, 603
- aidLength
 - appStats, 41
 - uim_appStatus, 579
 - uim_refreshevent, 597
 - uim_sessionInformation, 599
 - uim_UIMSessionInformation, 603
- aidVal
 - appStats, 41
 - uim_appStatus, 579
- aidingIndicatorMask
 - loc_sensorDataUsage, 163
- airTimerValue
 - voice_airTimer, 1033
- alertPitch
 - voice_signalInfo, 1077
- alertingPattern
 - voice_arrAlertingPattern, 1036
- AlertingType
 - voice_arrAlertingType, 1037
- alertmsg
 - unpack_omaDmConfigTlv_t, 817
- alertmsglength
 - unpack_omaDmConfigTlv_t, 817
- allCallsAlphaIDInfoArr
 - voice_arrAlphaID, 1037
- AllCallsUUSInfo
 - voice_arrUUSInfo, 1043
- alphaDcs
 - voice_alphaIDInfo, 1035
- AlphaID
 - cat_AIPhaIdentifierTlv, 49
- AlphaIDInfo
 - voice_allCallsAlphaIDInfo, 1034
- alphaIDLen
 - unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 877
- AlphaIDLength
 - cat_AIPhaIdentifierTlv, 49
- alphaLen
 - voice_alphaIDInfo, 1035
- alphaText
 - voice_alphaIDInfo, 1035
- altSrcInfo_t, 38
 - coverage, 39
 - linkage, 39
 - source, 39
- Altitude
 - unpack_pds_SLQSGetGPSSStateInfo_t, 833
- altitudeAssumed
 - unpack_loc_GnssSvInfo_Ind_t, 725
- altitudeSrcInfo
 - pack_loc_SLQSLOCInjectPosition_t, 387
- altitudeWrtEllipsoid
 - pack_loc_SLQSLOCInjectPosition_t, 387
- altitudeWrtMeanSeaLevel
 - pack_loc_SLQSLOCInjectPosition_t, 387
- ambr_dl
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
- ambr_dl_ext
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
- ambr_dl_ext2
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
- ambr_ul
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
- ambr_ul_ext
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
- ambr_ul_ext2
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
- amrMode
 - ims_AMRModeInfo, 84
- amrOctAlgn
 - ims_AMROctAlgnInfo, 85
- amrWBEnable
 - ims_EnabAMRWBInfo, 87
- amrWBMode
 - ims_AMRWBModeInfo, 86

- amrWBOctAlgn
 - ims_AMRWBOctAlgnInfo, 86
- amssSize
 - unpack_dms_GetFirmwareRevision_t, 629
 - unpack_dms_GetFirmwareRevisions_t, 631
- apdoxypages.c, 1105
- apnId
 - pack_qos_SLQSQosSwiReadApnExtraParams_t, 445
 - pack_qos_SLQSQosSwiReadDataStats_t, 445
 - unpack_qos_SLQSQosSwiReadApnExtraParams_t, 841
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- apnName
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1022
- apnname
 - unpack_wds_GetDefaultProfile_t, 978
 - unpack_wds_GetDefaultProfileV2_t, 982
- apnsize
 - unpack_wds_GetDefaultProfile_t, 978
 - unpack_wds_GetDefaultProfileV2_t, 982
- appNameLength
 - loc_LocApplicationInfo, 159
- appProviderLength
 - loc_LocApplicationInfo, 159
- appState
 - appStats, 41
 - uim_appStatus, 579
- 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, 549
 - uim_slotInfo, 602
- appType
 - appStats, 41
 - uim_appStatus, 579
- appVersionLength
 - loc_LocApplicationInfo, 159
- appVersionValid
 - loc_LocApplicationInfo, 159
- Application
 - unpack_nas_GetCDMANetworkParameters_t, 745
- appversion_str
 - unpack_dms_GetFirmwareInfo_t, 628
- arfcn
 - nas_GERANInfo, 202
 - nas_gsmCellInfo, 204
- arrCallInfomation
 - unpack_voice_allCallStatusCallback_ind_t, 928
- arrfileInfo
 - uim_refreshevent, 597
 - uim_registerRefresh, 597
- atrValue
 - uim_physlotInfo, 594
- atrValueLen
 - uim_physlotInfo, 594
- audio.h, 1105
 - pack_audio_SLQSGetAudioPathConfig, 1106
 - pack_audio_SLQSGetAudioProfile, 1106
 - pack_audio_SLQSGetAudioVolTLBConfig, 1107
 - pack_audio_SLQSSetAudioPathConfig, 1107
 - pack_audio_SLQSSetAudioProfile, 1108
 - pack_audio_SLQSSetAudioVolTLBConfig, 1108
 - unpack_audio_SLQSGetAudioPathConfig, 1109
 - unpack_audio_SLQSGetAudioProfile, 1109
 - unpack_audio_SLQSGetAudioVolTLBConfig, 1109
 - unpack_audio_SLQSSetAudioPathConfig, 1110
 - unpack_audio_SLQSSetAudioProfile, 1110
 - unpack_audio_SLQSSetAudioVolTLBConfig, 1110
- audio_RXAGCList, 42
 - pRXAIG, 42
 - pRXComprSlope, 43
 - pRXComprThres, 43
 - pRXExpSlope, 43
 - pRXExpThres, 43
 - pRXStaticGain, 43
- audio_RXAVCList, 43
 - pAVRXAVCHadroom, 43
 - pAVRXAVCSens, 43
- audio_RXPCMIRFiltr, 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_TXPCMIRFiltr, 46
 - pFlag, 47
 - pStage0Val, 47
 - pStage1Val, 47
 - pStage2Val, 47
 - pStage3Val, 47
 - pStage4Val, 47

- pStageCnt, [47](#)
- auth
 - unpack_wds_GetDefaultProfile_t, [978](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- authData
 - pack_uim_SLQSUIAuthenticate_t, [483](#)
- AuthProt
 - nas_protocolSubtypeElement, [272](#)
- Authentication
 - unpack_wds_SLQSGetRuntimeSettings_t, [1007](#)
- authentication
 - pack_wds_SetDefaultProfile_t, [517](#)
- AutoConnect
 - pack_swiavms_SLQSAVMSSetSettings_t, [465](#)
 - pack_swiavms_SLQSAVMSSetSettings_v2_t, [466](#)
 - unpack_swiavms_SLQSAVMSGetSettings_t, [885](#)
 - unpack_swiavms_SLQSAVMSGetSettings_v2_t, [888](#)
- AutoReboot
 - pack_swiavms_SLQSAVMSSetSettings_t, [465](#)
 - unpack_swiavms_SLQSAVMSGetSettings_t, [885](#)
- Autosdm
 - unpack_swioma_SLQSOMADMGetSettings_t, [908](#)
- avgPeriod
 - nas_LTESigRptCfg, [238](#)
 - nas_LTESigRptConfig, [239](#)
- azimuth
 - loc_satelliteInfo, [162](#)
- bAPNLength
 - PackSwiAvmsSetSettingsAPNInfo, [538](#)
 - PackSwiAVMSSettingsAPNInfo, [539](#)
- bAuto
 - pack_pds_SetPortAutomaticTracking_t, [436](#)
 - pack_pds_SetServiceAutomaticTracking_t, [436](#)
- bBinaryType
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
- bEnable
 - pack_dms_UIMSetPINProtection_t, [367](#)
 - pack_nas_SLQSSetSignalStrengthsCallback_t, [416](#)
- bEnabled
 - pack_pds_SetXTRAAutomaticDownload_t, [437](#)
- bForceDownload
 - pack_fms_SetImagesPreference_t, [370](#)
- bICCID
 - slot_t, [547](#)
- bICCIDLength
 - slot_t, [547](#)
- bLogicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, [489](#)
 - slot_t, [547](#)
- bNotification
 - UnpackSwiAvmsEventReportNotification, [1029](#)
- bNumberOfPhySlots
 - unpack_uim_SetUimSlotStatusChangeCallback_-ind_t, [917](#)
- BOOL
 - SwiDataTypes.h, [1433](#)
- BPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- bPWDLength
 - PackSwiAvmsSetSettingsAPNInfo, [538](#)
 - PackSwiAVMSSettingsAPNInfo, [539](#)
- bPackageID
 - UnpackSwiAvmsEventReportPackageID, [1030](#)
- bRegStatus
 - UnpackSwiAvmsEventReportRegStatus, [1030](#)
- bResetStatistics
 - rmTrasferStaticsReq, [545](#)
- bSerity
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
- bState
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
 - UnpackSwiAvmsEventReportConfig, [1026](#)
- bType
 - UnpackSwiAvmsEventReportDataSessionStatus, [1027](#)
 - UnpackSwiAvmsEventReportSessionType, [1031](#)
- bUnameLength
 - PackSwiAvmsSetSettingsAPNInfo, [538](#)
 - PackSwiAVMSSettingsAPNInfo, [540](#)
- bUserInputRequest
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
 - UnpackSwiAvmsEventReportConfig, [1026](#)
 - UnpackSwiAvmsEventReportConnectionRequest, [1027](#)
- BYT_STAT_STAT_MASK
 - wds.h, [1525](#)
- BYTE
 - SwiDataTypes.h, [1433](#)
- band
 - nas_LTEInfo, [224](#)
 - NASPhyCaAggScellArray, [332](#)
- band1900
 - nas_gsmCellInfo, [204](#)
- band_pref
 - NASBandPreferenceTlv, [322](#)
- BandCapability
 - unpack_dms_GetBandCapability_t, [614](#)
- bandCapability
 - unpack_dms_SLQSGetBandCapability_t, [661](#)
 - unpack_dms_SLQSGetBandCapabilityExt_t, [665](#)
- BandPref
 - nas_BandPrefTlv, [173](#)
- bandPref
 - pack_nas_SLQSSetBandPreference_t, [415](#)
- bandwidth
 - nas_LTEInfo, [224](#)
 - nas_RfBandwidthInfoElements, [283](#)
- baseId
 - nas_CDMAInfo, [176](#)
 - nas_CDMA SysInfo, [180](#)

- baseLat
 - nas_CDMAInfo, 176
 - nas_CDMASysInfo, 180
- baseLong
 - nas_CDMAInfo, 176
 - nas_CDMASysInfo, 180
- BasestationID
 - unpack_nas_SLQSGetServingSystem_t, 772
- BasestationLatitude
 - unpack_nas_SLQSGetServingSystem_t, 772
- BasestationLongitude
 - unpack_nas_SLQSGetServingSystem_t, 772
- batchPerSec
 - loc_accelAcceptReady, 148
 - loc_accelTempAcceptReady, 149
 - loc_gyroAcceptReady, 155
 - loc_gyroTempAcceptReady, 155
- BearerID
 - unpack_qos_QosFlowInfo_t, 838
- bearerID
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1011
- bearerId
 - unpack_QosFlowStat_t, 856
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1022
- bits_129_192
 - nas_BandPrefInfoTlv, 171
 - nas_lteBandPrefExt, 218
 - nas_nr5gBandPref, 254
- bits_193_256
 - nas_BandPrefInfoTlv, 171
 - nas_lteBandPrefExt, 218
 - nas_nr5gBandPref, 254
- bits_1_64
 - nas_BandPrefInfoTlv, 171
 - nas_lteBandPrefExt, 218
 - nas_nr5gBandPref, 254
- bits_65_128
 - nas_BandPrefInfoTlv, 171
 - nas_lteBandPrefExt, 218
 - nas_nr5gBandPref, 254
- bootSize
 - unpack_dms_GetFirmwareRevisions_t, 631
- BootString
 - unpack_dms_GetFirmwareRevisions_t, 631
- bootversion_str
 - unpack_dms_GetFirmwareInfo_t, 628
- Broadcast
 - unpack_nas_GetCDMANetworkParameters_t, 745
- broadcastActivate
 - pack_sms_SLQSSetSmsBroadcastActivation_t, 454
- broadcastConfig
 - sms_qaQmi3GPPBroadcastCfgInfo, 557
- bsInfoValid
 - nas_CDMASysInfo, 180
- bsPRev
 - nas_CDMASysInfo, 180
- bsPRevValid
 - nas_CDMASysInfo, 180
- bsic
 - nas_GERANInfo, 202
- bsicId
 - nas_gsmCellInfo, 204
- bucketSz
 - unpack_qos_tokenBucket_t, 855
- buildID
 - FMSImageIdElement, 80
 - image_info_t, 84
- buildIDLen
 - image_info_t, 84
- buildIDLength
 - FMSImageIdElement, 80
- buildId
 - FMSImageElement, 79
- buildIdLength
 - FMSImageElement, 79
- BurstDTMFInfo
 - pack_voice_SLQSVoiceBurstDTMF_t, 495
- ByteLoopbackMode
 - unpack_wds_SLQSSetLoopback_t, 1019
- ByteLoopbackMultiplier
 - unpack_wds_SLQSSetLoopback_t, 1019
- CCETlv
 - unpack_cat_SetCatEventCallback_ind_t, 610
- CDMA_P_Rev
 - unpack_nas_SLQSGetServingSystem_t, 772
- CDMABroadcastConfig
 - sms_qaQmi3GPP2BroadcastCfgInfo, 557
- CDMAECIOThreshListLen
 - nas_CDMAECIOThresh, 175
- CDMARSSIThreshListLen
 - nas_CDMARSSIThresh, 177
- CDMASSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 796
- CDMASystemInfoExt
 - unpack_nas_SLQSGetServingSystem_t, 772
- CHAR
 - SwiDataTypes.h, 1433
- CK_MAX_SIZE
 - dms.h, 1128
- CQIValueCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, 810
- CQIValueCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, 810
- CRTlv
 - unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 996
- CSDomain
 - unpack_nas_GetServingNetwork_t, 753
- CSTlv
 - unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 996
- CUGIndex
 - voice_CUGInfo, 1060

- CallBarStatus
 - unpack_nas_SLQSGetServingSystem_t, 772
- callDuration
 - unpack_wds_GetSessionDuration_t, 991
 - unpack_wds_GetSessionDurationV2_t, 992
- CallEndReason
 - wds_LastMdmCallEndRsnTlv, 1094
- callEndReason
 - unpack_wds_SLQSGetDUNCallInfo_t, 1003
 - voice_arrCallEndReason, 1039
- CallFWExtInfo
 - voice_getCallFWExtInfo, 1067
- CallFWInfo
 - voice_getCallFWInfo, 1067
- callID
 - pack_voice_SLQSVoiceGetCallInfo_t, 499
 - pack_voice_SLQSVoiceStopContDTMF_t, 510
 - unpack_voice_OTASPStatusCallback_ind_t, 931
 - unpack_voice_SLQSVoiceStopContDTMF_t, 963
 - unpack_voice_SUPSNotificationCallback_ind_t, 968
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
 - unpack_voice_voicePrivacyChangeCallback_ind_t, 972
 - voice_allCallsAlphaIDInfo, 1034
 - voice_allCallsDiagInfo, 1034
 - voice_allCallsUUSInfo, 1035
 - voice_arrAlertingPattern, 1036
 - voice_arrAlertingType, 1037
 - voice_arrCallEndReason, 1039
 - voice_arrSvcOption, 1042
 - voice_callInfo, 1052
 - voice_DTMFInfo, 1062
 - voice_getAllCallRmtPtyName, 1066
 - voice_getAllCallRmtPtyNum, 1066
 - voice_peerNumberInfo, 1070
- callNumber
 - pack_voice_SLQSVoiceDialCall_t, 496
- callState
 - voice_callInfo, 1052
- callType
 - voice_callInfo, 1052
- CalledPartyNum
 - voice_arrCalledPartyNum, 1038
- callerID
 - voice_callerIDInfo, 1047
 - voice_connectNumInfo, 1059
- callerIDLen
 - voice_callerIDInfo, 1047
 - voice_connectNumInfo, 1059
- callerName
 - voice_remotePartyName, 1075
- Callinfo
 - voice_getAllCallInformation, 1065
- campedCiotLteOpMode
 - nas_LteCiotOpModeTlv, 220
- cardProtocol
 - uim_physlotInfo, 594
- cardState
 - slotInf, 549
 - uim_slotInfo, 602
- carrier
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 671
- carrier_str
 - unpack_dms_GetFirmwareInfo_t, 628
- CarrierImage_t, 47
 - m_FwBuildId, 48
 - m_FwImageld, 48
 - m_PriBuildId, 48
 - m_PriImageld, 49
 - m_nCarrierId, 48
 - m_nFolderId, 48
 - m_nStorage, 48
- carrierStr
 - _litefw_FirmwareFileInfo, 36
- cat.h, 1111
 - pack_cat_CATSendEnvelopeCommand, 1112
 - pack_cat_CATSendTerminalResponse, 1112
 - pack_cat_SetCATEventCallback, 1112
 - unpack_cat_CATSendEnvelopeCommand, 1113
 - unpack_cat_CATSendEnvelopeCommand_t, 1112
 - unpack_cat_CATSendTerminalResponse, 1113
 - unpack_cat_CATSendTerminalResponse_t, 1112
 - unpack_cat_SetCATEventCallback, 1114
 - unpack_cat_SetCatEventCallback_ind, 1114
- 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
 - CatAlphaIdtfr, 50
 - CatEndPS, 50
 - CatEvIDDData, 50
 - CatEventLst, 50
 - CatRefresh, 50
- CatAlphaIdtfr
 - cat_currentCatEvent, 50
- CatEndPS
 - cat_currentCatEvent, 50
- CatEvIDDData

- cat_currentCatEvent, [50](#)
- CatEvent
 - cat_commonEventTlv, [50](#)
- CatEventLst
 - cat_currentCatEvent, [50](#)
- CatRefresh
 - cat_currentCatEvent, [50](#)
- causeCode
 - unpack_sms_SLQSWmsAsyncRawSendCallBack-
_ind_t, [877](#)
- ccsSupported
 - nas_CDMASysInfo, [180](#)
- ccsSupportedValid
 - nas_CDMASysInfo, [180](#)
- cdmaSSInfo, [52](#)
 - ecio, [53](#)
 - rsi, [53](#)
- cdmaSysIdValid
 - nas_CDMASysInfo, [180](#)
- cell_resel_priority
 - nas_infoInterFreq, [216](#)
- cellBroadcastCap
 - nas_AddSysInfo, [170](#)
- CellID
 - unpack_nas_SLQSGetservingSystem_t, [772](#)
- cellID
 - nas_GERANInfo, [202](#)
 - nas_QmisNasSlqsNasPCICellInfo, [276](#)
 - nas_UMTSInfo, [312](#)
- cellId
 - nas_GSMSysInfo, [208](#)
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [258](#)
 - nas_WCDMASysInfo, [320](#)
- cellIdValid
 - nas_gsmCellInfo, [204](#)
 - nas_GSMSysInfo, [208](#)
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [258](#)
 - nas_WCDMASysInfo, [321](#)
- cellInterFreqParams
 - nas_infoInterFreq, [216](#)
- cellsTDD
 - nas_umtsLTENbrCell, [314](#)
- CellParams
 - nas_LTEInfoIntrafreq, [227](#)
- cellReselPriority
 - nas_lteGsmCellInfo, [222](#)
 - nas_LTEInfoIntrafreq, [227](#)
 - nas_lteWcdmaCellInfo, [244](#)
- cells_len
 - nas_infoInterFreq, [216](#)
 - nas_lteGsmCellInfo, [222](#)
- cellsLen
 - nas_LTEInfoIntrafreq, [227](#)
 - nas_lteWcdmaCellInfo, [244](#)
- CfgValue
 - pack_swidms_SLQSSwiDmsSetUsbComp_t, [470](#)
- swidms_ifaceCfgTlv, [571](#)
- chaddr
 - wds_DHCPv4HWConfig, [1088](#)
 - wdsDhcpv4HwConfig, [1102](#)
- chaddrLen
 - wds_DHCPv4HWConfig, [1088](#)
 - wdsDhcpv4HwConfig, [1102](#)
- changePIN
 - pack_uim_ChangePin_t, [480](#)
- ChannelRate
 - wds_ChannelRateTlv, [1081](#)
- channelRate
 - unpack_wds_SLQSGetDUNCallInfo_t, [1003](#)
- Chipset
 - nas_DeviceConfigDetail, [193](#)
- CiotAOPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- CiotAcqOrderLen
 - NASCiotAcqOrderPrefTlv, [323](#)
- ciotAcqOrderLen
 - nas_ciotAcqOrderPref, [183](#)
 - nas_CiotAcqOrderPrefTlv, [184](#)
- CiotAcqOrderPref
 - NASCiotAcqOrderPrefTlv, [323](#)
- CiotLteOpModePref
 - NASCiotLteOpModePrefTlv, [323](#)
- ciotLteOpModePref
 - nas_CiotLteOpModePrefTlv, [184](#)
- CiotOpMPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- ckLen
 - uim_depersonalizationInformation, [585](#)
- ckVal
 - uim_depersonalizationInformation, [585](#)
- clear
 - pack_dms_SLQSSwiGetCrashInfo_t, [360](#)
- cmdID
 - pack_cat_CATSendEnvelopeCommand_t, [349](#)
- codingScheme
 - nas_PLMNNetworkNameData, [270](#)
 - voice_remotePartyName, [1075](#)
- common.h
 - eAUDIO, [1118](#)
 - eCAT, [1118](#)
 - eCTL, [1117](#)
 - eDMS, [1117](#)
 - eIMS, [1118](#)
 - eIMSA, [1118](#)
 - eIND, [1118](#)
 - eLOC, [1118](#)
 - eLOG_DEBUG, [1117](#)
 - eLOG_FATAL, [1117](#)
 - eLOG_INFO, [1117](#)
 - eLOG_WARN, [1117](#)
 - eNAS, [1117](#)
 - ePDS, [1118](#)
 - eQOS, [1118](#)
 - eREQ, [1118](#)

- eRMS, [1118](#)
- eRSP, [1118](#)
- eSAR, [1118](#)
- eSMS, [1118](#)
- eSWIAUDIO, [1118](#)
- eSWIDMS, [1118](#)
- eSWILOC, [1118](#)
- eSWIM2MCMD, [1118](#)
- eSWIM2MCMD_AVC2, [1118](#)
- eSWIOMA, [1118](#)
- eSWIOMAEXT, [1118](#)
- eTIMEOUT_10_S, [1118](#)
- eTIMEOUT_20_S, [1118](#)
- eTIMEOUT_2_S, [1118](#)
- eTIMEOUT_300_S, [1118](#)
- eTIMEOUT_30_S, [1118](#)
- eTIMEOUT_5_S, [1118](#)
- eTIMEOUT_60_S, [1118](#)
- eTIMEOUT_8_S, [1118](#)
- eTIMEOUT_DEFAULT, [1118](#)
- eTMD, [1118](#)
- eUIM, [1118](#)
- eVOICE, [1118](#)
- eWDS, [1117](#)
- common.h, [1114](#)
 - eLOG_LEVEL, [1117](#)
 - eQMI_SVC, [1117](#)
 - eTimeout, [1118](#)
 - fill_pack_ctx, [1119](#)
 - fill_sdu_hdr, [1119](#)
 - get_version, [1119](#)
 - glog, [1120](#)
 - gloglvl, [1120](#)
 - helper_get_error_code, [1119](#)
 - helper_get_error_reason, [1119](#)
 - helper_get_req_str, [1119](#)
 - helper_get_resp_ctx, [1119](#)
 - helper_get_xid, [1119](#)
 - helper_isBootLoader_DebugEnabled, [1119](#)
 - helper_set_log_func, [1120](#)
 - helper_set_log_lvl, [1120](#)
 - liteqmi_GetVersion, [1120](#)
 - liteqmi_helper_decode7bitAsciiEncString, [1120](#)
 - liteqmi_log, [1120](#)
 - logger, [1117](#)
 - MINREQBKLEN, [1116](#)
 - MSGID_AND_LEN, [1116](#)
 - MSGID_DONT_CARE, [1116](#)
 - msgtype, [1118](#)
 - SDU_HDR_LEN, [1117](#)
 - UNUSEDPARAM, [1117](#)
 - unpack_result_code_only, [1120](#)
- commonInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, [804](#)
- ConcSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, [773](#)
- confidence
 - pack_loc_SLQSLOCSetCradleMountConfig_t, [391](#)
- conn_status
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [1011](#)
- ConnectedPartyNum
 - voice_arrConnectPartyNum, [1040](#)
- connectionStatus, [53](#)
 - MDMCallDuration, [53](#)
 - MDMConnStatus, [53](#)
 - unpack_wds_GetSessionState_t, [992](#)
 - unpack_wds_SLQSGetDUNCallInfo_t, [1003](#)
- content
 - uim_authenticateResult, [580](#)
 - uim_readResult, [595](#)
- contentLen
 - uim_authenticateResult, [580](#)
 - uim_readResult, [595](#)
- context
 - uim_authenticationData, [581](#)
- contextId
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [535](#)
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [1022](#)
- contextType
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [535](#)
- ControlMac
 - nas_protocolSubtypeElement, [272](#)
- count
 - swidms_SwiDmsGetHWWatchdog, [573](#)
- count0
 - nas_RankIndicatorTlv, [278](#)
- count1
 - nas_RankIndicatorTlv, [278](#)
- countryInitials
 - nas_PLMNNetworkNameData, [270](#)
- coverage
 - altSrcInfo_t, [39](#)
- cphy_ca_dl_bandwidth
 - NASPhyCaAggScellArray, [332](#)
- cphy_scell_info_list_len
 - NASPhyCaAggScellArray, [332](#)
- cpich_ecno
 - nas_wcdmaCellInfo, [316](#)
- cpich_rscp
 - nas_wcdmaCellInfo, [316](#)
- cradleMountConfigStatus
 - unpack_loc_CradleMountCallback_Ind_t, [717](#)
- crashAction
 - pack_dms_SetCrashAction_t, [353](#)
- crashData
 - crashInformation, [55](#)
- crashId
 - crashInformation, [55](#)
- crashInfo
 - crashInfoParams, [54](#)
- crashInfoParam
 - unpack_dms_SLQSSwiGetCrashInfo_t, [669](#)

- 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, 296
 - NASServingSystemInfo, 341
- csBarStatus
 - nas_CallBarringSysInfo, 173
 - nas_callBarStatus, 174
- cscfPortName
 - ims_CSCFPortNameInfo, 87
- csgld
 - nas_Csgld, 187
- cur_carr_name
 - unpack_dms_GetFirmwareInfo_t, 628
- cur_carr_rev
 - unpack_dms_GetFirmwareInfo_t, 628
- curDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, 1000
- CurIndex
 - dms_devMaxCfgListCaps, 59
- curProfile
 - pack_wds_SLQSMModifyProfile_t, 527
 - UnPackGetProfileSettingOut, 1023
 - UnPackGetProfileSettingOutV2, 1023
- CurSubsCapsLen
 - dms_devCurSubsCaps, 58
- CurrChanRxRate
 - dunchannelRate, 77
 - wds_channelRate, 1081
- CurrChanTxRate
 - dunchannelRate, 77
 - wds_channelRate, 1081
- currDBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1016
- currNWInfo
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1016
- currNetworkInfo, 55
 - NetworkType, 56
 - RATMask, 56
 - SOMask, 56
- unpack_wds_SLQSGetCurrDataSystemStat_t, 998
- current_channel_rx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 1000
- current_channel_tx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 1000
- CurrentCfgType
 - swidms_intfaceCfgTlv, 571
- currentChannelRXRate
 - unpack_wds_GetConnectionRate_t, 975
- currentChannelTXRate
 - unpack_wds_GetConnectionRate_t, 975
- currentDataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, 530
- currentNetwork
 - qmiWSDDataBearerTechnology, 542
- CurrentPLMN
 - unpack_nas_SLQSGetServingSystem_t, 773
- CurrentmitigationLvl
 - unpack_tmd_SLQSTmdGetMitigationLvl_t, 911
- cust_attr
 - DMScustSettingInfo, 75
- cust_id
 - DMScustSettingInfo, 75
 - DMSgetCustomInput, 77
 - pack_dms_GetCustFeaturesV2_t, 351
 - pack_dms_SetCustFeaturesV2_t, 356
- cust_value
 - DMScustSettingInfo, 75
 - pack_dms_SetCustFeaturesV2_t, 356
- custSetting
 - DMScustSettingList, 75
- CustomSCP
 - unpack_nas_GetCDMANetworkParameters_t, 745
- CwtMute
 - unpack_swiaudio_SLQSGetM2MAudioProfile_t, 879
 - unpack_swiaudio_SLQSGetM2MAVMute_t, 881
- cycleLength
 - nas_EdrxCycleLength, 196
- DBTTlv
 - unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 996
- dBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1016
- dBTechExtRatValue
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- dBTechExtSoMask
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- dBTechnology
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- dBtechExtAvail

- unpack_wds_SLQSSetWdsEventCallback_ind_t, 1016
- dBtechnologyExt
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- DHCPRelayEnabled
 - pack_dms_SetCustFeature_t, 355
 - unpack_dms_GetCustFeature_t, 617
- DHCPv4LeaseOptTlv
 - unpack_wds_DHCPv4ClientLease_ind_t, 973
- DHCPv4LeaseStateTlv
 - unpack_wds_DHCPv4ClientLease_ind_t, 973
- dLatitude
 - unpack_pds_SetEventReport_Ind_t, 827
- dLongitude
 - unpack_pds_SetEventReport_Ind_t, 827
- DMS_IMGDETAILS_LEN
 - dms.h, 1128
- DMS_PM_FACTORY
 - dms.h, 1129
- DMS_PM_LOW
 - dms.h, 1129
- DMS_PM_OFFLINE
 - dms.h, 1129
- DMS_PM_ONLINE
 - dms.h, 1129
- DMS_PM_RESET
 - dms.h, 1129
- DMS_PM_SHUT_DOWN
 - dms.h, 1129
- DMS_VALID_FSN_LEN
 - dms.h, 1130
- 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, 194
- DRCValue
 - nas_DRCCParams, 194
- DSTlv
 - unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 996
- DTMFEvent
 - voice_DTMFInfo, 1062
- DTMFInformation
 - unpack_voice_DTMFEventCallback_ind_t, 929
- DTMFInterdigitInterval
 - voice_DTMFLengths, 1063
- DTMFPulseWidth
 - voice_DTMFLengths, 1063
- DTMFDigit
 - pack_voice_SLQSVoiceStartContDTMF_t, 510
- DTMInd
 - unpack_nas_SLQSGetServingSystem_t, 773
- Data
 - cat_EventIDDDataTlv, 51
- data
 - sMSCAddressInfo, 563
 - sMSEtwSMessageInfo, 564
 - sMSTransferRouteMTMessageInfo, 568
 - uim_authenticationData, 581
- data_buf
 - NASOTAMessageTlv, 329
- data_len
 - NASOTAMessageTlv, 330
- dataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, 530
- dataBearerMask
 - unpack_wds_SLQSGetDataBearerTechnology_t, 1000
- DataBearerTech
 - wds_DataBearTechTlv, 1084
- dataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, 1003
- dataBearerTechExt
 - pack_wds_SLQSSetWdsEventCallback_t, 530
- dataCapabilities
 - nas_dataSrvCapabilities, 189
- dataCapabilitiesLen
 - nas_dataSrvCapabilities, 189
- DataCaps
 - unpack_nas_GetServingNetwork_t, 753
 - unpack_nas_GetServingNetworkCapabilities_t, 755
- dataCaps
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 758
- DataCapsLen
 - unpack_nas_GetServingNetwork_t, 753
 - unpack_nas_GetServingNetworkCapabilities_t, 755
- dataCapsSize
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 758
- dataLen
 - pack_cat_CATSendEnvelopeCommand_t, 349
 - pack_cat_CATSendTerminalResponse_t, 350
 - uim_authenticationData, 581
- DataLength
 - cat_EventIDDDataTlv, 51
- DataRate
 - unpack_qos_swiQosFlow_t, 853

- dataRateMax
 - unpack_qos_dataRate_t, [835](#)
- dataServiceCaCapability
 - unpack_dms_GetDeviceCapabilities_t, [620](#)
- DataServiceCapability
 - dms_devCaps, [57](#)
 - unpack_dms_GetDeviceCap_t, [619](#)
- DataSrvCapabilities
 - unpack_nas_SLQSSetServingSystem_t, [773](#)
- dataSysStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [1016](#)
- dataSystemStatus
 - pack_wds_SLQSSetWdsEventCallback_t, [530](#)
- Date
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [902](#)
- DateLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [902](#)
- day
 - nas_timeInfo, [309](#)
 - nas_UniversalTime, [315](#)
- dayLtSavingAdj
 - nas_timeInfo, [309](#)
- dayOfWeek
 - nas_timeInfo, [309](#)
 - nas_UniversalTime, [315](#)
- daylightSavings
 - nas_qaQmi3Gpp2TimeZone, [273](#)
- dedicatedBand
 - nas_RfDedicatedBandInfoElements, [285](#)
- defaultPDNEnabled
 - unpack_wds_SLQSSet3GPPConfigItem_t, [997](#)
- DefaultRoamInd
 - unpack_nas_SLQSSetServingSystem_t, [773](#)
- delayClass
 - LibPackGPRSRequestedQoS, [105](#)
 - wds_GPRSQoS, [1092](#)
- deliveryErrSDU
 - LibPackUMTSQoS, [146](#)
 - wds_UMTSMInQoS, [1101](#)
- depersonalisationInfo
 - pack_uim_SLQSUIDepersonalization_t, [484](#)
- description
 - unpack_omaDmFotaTlv_t, [819](#)
- descriptionlength
 - unpack_omaDmFotaTlv_t, [819](#)
- Desription
 - nas_QmiNas3GppNetworkInfo, [274](#)
- destPortRangeEnd
 - LibPackTFTIDParams, [143](#)
- destPortRangeStart
 - LibPackTFTIDParams, [143](#)
- DetailedSvcInfo
 - unpack_nas_SLQSSetServingSystem_t, [773](#)
- DevCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- DevCfgListLen
 - dms_devMaxCfgListCaps, [59](#)
- DevCrashState
 - unpack_dms_GetCrashAction_t, [614](#)
- Device
 - pack_swiaudio_SLQSSetM2MAudioAVCFG_t, [459](#)
- DeviceError
 - qmTlvResult, [542](#)
- deviceId
 - unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, [911](#)
- deviceIdLen
 - unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, [911](#)
- DeviceResult
 - qmTlvResult, [542](#)
- DiagInfo
 - voice_allCallsDiagInfo, [1034](#)
 - voice_arrDiagInfo, [1040](#)
- diagInfoLen
 - voice_diagInfo, [1061](#)
- diagnosticInfo
 - voice_diagInfo, [1061](#)
- digitBuff
 - voice_DTMFInfo, [1062](#)
- digitCnt
 - voice_burstDTMFInfo, [1044](#)
 - voice_DTMFInfo, [1062](#)
- dirNum
 - nas_dirNum, [193](#)
- dirNumLen
 - nas_dirNum, [193](#)
- direction
 - voice_callInfo, [1052](#)
- DisableIMSI
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- dispType
 - voice_extDispRecInfo, [1064](#)
- displayCondition
 - nas_serviceProviderName, [294](#)
- dl_bw_value
 - nas_PhyCaAggPcellInfo, [262](#)
 - nas_PhyCaAggScellIDBw, [262](#)
 - nas_PhyCaAggScellInfo, [266](#)
 - NASPhyCaAggPcellInfo, [330](#)
 - NASPhyCaAggScellIDBw, [332](#)
 - NASPhyCaAggScellInfo, [334](#)
- dms.h, [1120](#)
 - ACT_CODE_MAX_SIZE, [1128](#)
 - CK_MAX_SIZE, [1128](#)
 - DMS_IMGDETAILS_LEN, [1128](#)
 - DMS_PM_FACTORY, [1129](#)
 - DMS_PM_LOW, [1129](#)
 - DMS_PM_OFFLINE, [1129](#)
 - DMS_PM_ONLINE, [1129](#)
 - DMS_PM_RESET, [1129](#)
 - DMS_PM_SHUT_DOWN, [1129](#)

- DMS_VALID_FSN_LEN, 1130
- ERI_DATA_MAX_SIZE, 1130
- MAX_BUILD_ID_LEN, 1130
- MEID_MAX_SIZE, 1130
- pack_dms_ActivateAutomatic, 1130
- pack_dms_GetActivationState, 1130
- pack_dms_GetBandCapability, 1131
- pack_dms_GetCrashAction, 1131
- pack_dms_GetCustFeature, 1131
- pack_dms_GetCustFeaturesV2, 1132
- pack_dms_GetDeviceCap, 1132
- pack_dms_GetDeviceCapabilities, 1133
- pack_dms_GetDeviceCapabilitiesV2, 1133
- pack_dms_GetDeviceHardwareRev, 1133
- pack_dms_GetDeviceMfr, 1134
- pack_dms_GetDeviceSerialNumbers, 1134
- pack_dms_GetFSN, 1135
- pack_dms_GetFirmwareInfo, 1134
- pack_dms_GetFirmwareRevision, 1135
- pack_dms_GetFirmwareRevisions, 1135
- pack_dms_GetHardwareRevision, 1136
- pack_dms_GetIMSI, 1136
- pack_dms_GetManufacturer, 1136
- pack_dms_GetModelID, 1137
- pack_dms_GetNetworkTime, 1137
- pack_dms_GetNetworkTimeV2, 1137
- pack_dms_GetOfflineReason, 1138
- pack_dms_GetPRLVersion, 1138
- pack_dms_GetPower, 1138
- pack_dms_GetSerialNumbers, 1139
- pack_dms_GetUSBComp, 1139
- pack_dms_GetVoiceNumber, 1139
- pack_dms_ResetToFactoryDefaults, 1140
- pack_dms_SLQSDmsSwiGetPCInfo, 1144
- pack_dms_SLQSDmsSwiGetResetInfo, 1144
- pack_dms_SLQSDmsSwiGetUimSelection, 1144
- pack_dms_SLQSDmsSwiIndicationRegister, 1145
- pack_dms_SLQSGetBandCapability, 1145
- pack_dms_SLQSGetERIFile, 1145
- pack_dms_SLQSGetPowerSaveModeConfig, 1146
- pack_dms_SLQSSetPowerSaveModeConfig, 1146
- pack_dms_SLQSSwiClearDyingGaspStatistics, 1146
- pack_dms_SLQSSwiGetCrashInfo, 1147
- pack_dms_SLQSSwiGetDyingGaspCfg, 1147
- pack_dms_SLQSSwiGetDyingGaspStatistics, 1147
- pack_dms_SLQSSwiGetFirmwareCurr, 1148
- pack_dms_SLQSSwiGetFwUpdateStatus, 1148
- pack_dms_SLQSSwiGetHostDevInfo, 1148
- pack_dms_SLQSSwiGetOSInfo, 1149
- pack_dms_SLQSSwiGetSerialNoExt, 1149
- pack_dms_SLQSSwiSetDyingGaspCfg, 1149
- pack_dms_SLQSSwiSetHostDevInfo, 1150
- pack_dms_SLQSSwiSetOSInfo, 1150
- pack_dms_SLQSUIMGetState, 1151
- pack_dms_SetActivationStatusCallback, 1140
- pack_dms_SetCrashAction, 1141
- pack_dms_SetCustFeature, 1141
- pack_dms_SetCustFeaturesV2, 1141
- pack_dms_SetEventReport, 1142
- pack_dms_SetFirmwarePreference, 1142
- pack_dms_SetIndicationRegister, 1142
- pack_dms_SetPower, 1143
- pack_dms_SetUSBComp, 1143
- pack_dms_SwiSetEventReport, 1151
- pack_dms_SwiUimSelect, 1151
- pack_dms_UIMChangePIN, 1152
- pack_dms_UIMGetControlKeyStatus, 1152
- pack_dms_UIMGetICCID, 1152
- pack_dms_UIMGetPINStatus, 1153
- pack_dms_UIMSetControlKeyProtection, 1153
- pack_dms_UIMSetPINProtection, 1154
- pack_dms_UIMUnblockControlKey, 1154
- pack_dms_UIMUnblockPIN, 1154
- pack_dms_UIMVerifyPIN, 1155
- pack_dms_ValidateSPC, 1155
- SPC_SIZE, 1130
- UNIQUE_ID_LEN, 1130
- unpack_dms_ActivateAutomatic, 1155
- unpack_dms_GetActivationState, 1156
- unpack_dms_GetBandCapability, 1156
- unpack_dms_GetCrashAction, 1157
- unpack_dms_GetCustFeature, 1157
- unpack_dms_GetCustFeaturesV2, 1157
- unpack_dms_GetDeviceCap, 1158
- unpack_dms_GetDeviceCapabilities, 1158
- unpack_dms_GetDeviceCapabilitiesV2, 1158
- unpack_dms_GetDeviceHardwareRev, 1159
- unpack_dms_GetDeviceMfr, 1159
- unpack_dms_GetDeviceSerialNumbers, 1159
- unpack_dms_GetFSN, 1161
- unpack_dms_GetFirmwareInfo, 1160
- unpack_dms_GetFirmwareRevision, 1160
- unpack_dms_GetFirmwareRevisions, 1160
- unpack_dms_GetHardwareRevision, 1161
- unpack_dms_GetIMSI, 1161
- unpack_dms_GetManufacturer, 1162
- unpack_dms_GetModelID, 1162
- unpack_dms_GetNetworkTime, 1163
- unpack_dms_GetNetworkTimeV2, 1163
- unpack_dms_GetOfflineReason, 1163
- unpack_dms_GetPRLVersion, 1164
- unpack_dms_GetPower, 1164
- unpack_dms_GetSerialNumbers, 1164
- unpack_dms_GetUSBComp, 1165
- unpack_dms_GetVoiceNumber, 1165
- unpack_dms_PSMCfgChange_ind, 1165
- unpack_dms_ResetToFactoryDefaults, 1166
- unpack_dms_SLQSDmsSwiGetPCInfo, 1170
- unpack_dms_SLQSDmsSwiGetResetInfo, 1170
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 1170
- unpack_dms_SLQSDmsSwiGetUimSelection, 1171

- unpack_dms_SLQSDmsSwiIndicationRegister, 1171
- unpack_dms_SLQSSGetBandCapability, 1171
- unpack_dms_SLQSSGetBandCapabilityExt, 1172
- unpack_dms_SLQSSGetERIFile, 1172
- unpack_dms_SLQSSGetPowerSaveModeConfig, 1173
- unpack_dms_SLQSSSetPowerSaveModeConfig, 1173
- unpack_dms_SLQSSwiClearDyingGaspStatistics, 1173
- unpack_dms_SLQSSwiGetCrashInfo, 1174
- unpack_dms_SLQSSwiGetDyingGaspCfg, 1174
- unpack_dms_SLQSSwiGetDyingGaspStatistics, 1174
- unpack_dms_SLQSSwiGetFirmwareCurr, 1175
- unpack_dms_SLQSSwiGetFwUpdateStatus, 1175
- unpack_dms_SLQSSwiGetHostDevInfo, 1175
- unpack_dms_SLQSSwiGetOSInfo, 1176
- unpack_dms_SLQSSwiGetSerialNoExt, 1176
- unpack_dms_SLQSSwiSetDyingGaspCfg, 1176
- unpack_dms_SLQSSwiSetHostDevInfo, 1177
- unpack_dms_SLQSSwiSetOSInfo, 1177
- unpack_dms_SLQSUIMGetState, 1177
- unpack_dms_SetActivationStatusCallback, 1166
- unpack_dms_SetCrashAction, 1166
- unpack_dms_SetCustFeature, 1167
- unpack_dms_SetCustFeaturesV2, 1167
- unpack_dms_SetEventReport, 1167
- unpack_dms_SetEventReport_ind, 1168
- unpack_dms_SetFirmwarePreference, 1168
- unpack_dms_SetIndicationRegister, 1168
- unpack_dms_SetPower, 1169
- unpack_dms_SetUSBComp, 1169
- unpack_dms_SwiEventReportCallBack_ind, 1178
- unpack_dms_SwiSetEventReport, 1178
- unpack_dms_SwiUimSelect, 1178
- unpack_dms_UIMChangePIN, 1179
- unpack_dms_UIMGetControlKeyStatus, 1179
- unpack_dms_UIMGetICCID, 1179
- unpack_dms_UIMGetPINStatus, 1180
- unpack_dms_UIMSetControlKeyProtection, 1180
- unpack_dms_UIMSetPINProtection, 1180
- unpack_dms_UIMUnblockControlKey, 1181
- unpack_dms_UIMUnblockPIN, 1181
- unpack_dms_UIMVerifyPIN, 1182
- unpack_dms_ValidateSPC, 1182
- 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
 - CurIndex, 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, [1091](#)
- domainLen
 - wds_Domain, [1090](#)
- DomainList
 - unpack_wds_SLQSSetRuntimeSettings_t, [1007](#)
- domainName
 - wds_Domain, [1090](#)
- DormancyStat
 - wds_DormStatTlv, [1091](#)
- dormancyStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [1017](#)
- dormancyState
 - unpack_wds_GetDormancyState_t, [983](#)
- dormancyStatus
 - pack_wds_SLQSSetWdsEventCallback_t, [530](#)
 - unpack_wds_SLQSSetDUNCallInfo_t, [1003](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [1017](#)
- downLink
 - voice_NSSAudioCtrl, [1069](#)
- dtmSupp
 - nas_GSMSysInfo, [208](#)
- dtmSuppValid
 - nas_GSMSysInfo, [208](#)
- dunchannelRate, [77](#)
 - CurrChanRxRate, [77](#)
 - CurrChanTxRate, [77](#)
 - MaxChanRxRate, [77](#)
 - MaxChanTxRate, [77](#)
- Duration
 - pack_nas_SetNetworkPreference_t, [398](#)
 - unpack_nas_GetNetworkPreference_t, [750](#)
- durationDueToOOS
 - dms_PSMDurationDueToOOSTlv, [67](#)
- durationThreshold
 - dms_PSMDurationThresholdTlv, [68](#)
- eAUDIO
 - common.h, [1118](#)
- eCAT
 - common.h, [1118](#)
- eCTL
 - common.h, [1117](#)
- eDMS
 - common.h, [1117](#)
- eFILE_TYPE_CAR_PRI
 - lite-fw.h, [1209](#)
- eFILE_TYPE_COMPO_PRI
 - lite-fw.h, [1209](#)
- eFILE_TYPE_NONE
 - lite-fw.h, [1209](#)
- eFILE_TYPE_OEM_PRI
 - lite-fw.h, [1209](#)
- eFIREHOSE_ERR_SECBOOT_INVALID_CERT_CHA-
IN
 - lite-fw.h, [1210](#)
- eFW_TYPE_CWE
 - lite-fw.h, [1209](#)
- eFW_TYPE_CWE_NVU
 - lite-fw.h, [1210](#)
- eFW_TYPE_INVALID
 - lite-fw.h, [1209](#)
- eFW_TYPE_MBN
 - lite-fw.h, [1209](#)
- eFW_TYPE_MBN_GOBI
 - lite-fw.h, [1209](#)
- eFW_TYPE_NVU
 - lite-fw.h, [1209](#)
- eFW_TYPE_SPK
 - lite-fw.h, [1209](#)
- eIMAGE_TYPE_ANY
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_APPL
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_APPS
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_BOOT
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_FILE
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_INVALID
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_KEYS
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_MAX
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_MIN

- lite-fw.h, [1210](#)
- eIMAGE_TYPE_MODM
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_SPKG
 - lite-fw.h, [1210](#)
- eIMAGE_TYPE_USER
 - lite-fw.h, [1210](#)
- eIMS
 - common.h, [1118](#)
- eIMSA
 - common.h, [1118](#)
- eIND
 - common.h, [1118](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_1
 - nas.h, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_10
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_11
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_12
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_125
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_126
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_127
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_13
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_14
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_17
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_18
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_19
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_2
 - nas.h, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_20
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_21
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_23
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_24
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_25
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_250
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_26
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_27
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_28
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_29
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_3
 - nas.h, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_30
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_31
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_32
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_33
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_34
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_35
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_36
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_37
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_38
 - nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_39
 - nas.h, [1254](#)

- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_4
nas.h, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_40
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_41
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_42
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_43
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_46
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_47
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_48
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_5
nas.h, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_6
nas.h, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_66
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_7
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_71
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_8
nas.h, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_9
nas.h, [1254](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CELL_STATE_CONFIGURED_ACTIVATED
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CELL_STATE_CONFIGURED_DEACTIVATED
nas.h, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_CELL_STATE_DECONFIGURED
nas.h, [1255](#)
- eLOC
common.h, [1118](#)
- eLOG_DEBUG
common.h, [1117](#)
- eLOG_FATAL
common.h, [1117](#)
- eLOG_INFO
common.h, [1117](#)
- eLOG_WARN
common.h, [1117](#)
- eModel_9X15
lite-fw.h, [1210](#)
- eModel_9X30
lite-fw.h, [1210](#)
- eModel_9x06
lite-fw.h, [1210](#)
- eModel_9x07
lite-fw.h, [1210](#)
- eModel_Unknown
lite-fw.h, [1210](#)
- eModel_WP9X15
lite-fw.h, [1210](#)
- eNAS
common.h, [1117](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
nas.h, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
nas.h, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
nas.h, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
nas.h, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
nas.h, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
nas.h, [1255](#)
- eNAS_LTE_CPHY_CELL_STATE_CONFIGURED_ACTIVATED_LITE
nas.h, [1255](#)
- eNAS_LTE_CPHY_CELL_STATE_CONFIGURED_DEACTIVATED_LITE
nas.h, [1255](#)
- eNAS_LTE_CPHY_CELL_STATE_DECONFIGURED_LITE
nas.h, [1255](#)
- ePDS
common.h, [1118](#)
- eQCWWAN_ERR_API_MUTEX_TIMEOUT
qmerrno.h, [1368](#)

- eQCWWAN_ERR_BUFFER_SZ
qmerrno.h, [1367](#)
- eQCWWAN_ERR_CANCEL_OP
qmerrno.h, [1368](#)
- eQCWWAN_ERR_DRIVER
qmerrno.h, [1368](#)
- eQCWWAN_ERR_ENUM_BEGIN
qmerrno.h, [1367](#)
- eQCWWAN_ERR_ENUM_END
qmerrno.h, [1368](#)
- eQCWWAN_ERR_FILE_COPY
qmerrno.h, [1367](#)
- eQCWWAN_ERR_FILE_OPEN
qmerrno.h, [1367](#)
- eQCWWAN_ERR_GENERAL
qmerrno.h, [1367](#)
- eQCWWAN_ERR_INTERNAL
qmerrno.h, [1367](#)
- eQCWWAN_ERR_INVALID_ARG
qmerrno.h, [1367](#)
- eQCWWAN_ERR_INVALID_DEVID
qmerrno.h, [1367](#)
- eQCWWAN_ERR_INVALID_FILE
qmerrno.h, [1367](#)
- eQCWWAN_ERR_INVALID_QMI_RSP
qmerrno.h, [1367](#)
- eQCWWAN_ERR_INVALID_XID
qmerrno.h, [1368](#)
- eQCWWAN_ERR_MALFORMED_QMI_RSP
qmerrno.h, [1367](#)
- eQCWWAN_ERR_MEMORY
qmerrno.h, [1367](#)
- eQCWWAN_ERR_MULTIPLE_DEVICES
qmerrno.h, [1368](#)
- eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
qmerrno.h, [1368](#)
- eQCWWAN_ERR_NO_CANCELABLE_OP
qmerrno.h, [1368](#)
- eQCWWAN_ERR_NO_CONNECTION
qmerrno.h, [1367](#)
- eQCWWAN_ERR_NO_DEVICE
qmerrno.h, [1367](#)
- eQCWWAN_ERR_NO_SIGNAL
qmerrno.h, [1368](#)
- eQCWWAN_ERR_NONE
qmerrno.h, [1367](#)
- eQCWWAN_ERR_NULL_TLV
qmerrno.h, [1371](#)
- eQCWWAN_ERR_OFFLINE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_PDU_GENERATION
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_ABORTED
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_ACCESS_DENIED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_ACK_NOT_SENT
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_ARG_TOO_LONG
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_CALL_FAILED
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_CARD_BUSY_RSP
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_CAT_END
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_CAT_START
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_CAUSE_CODE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_CONNECT
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_DEVICE_IN_USE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_DEVICE_NOT_READY
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_DISABLED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_ENCODING
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_EVENT_REG_FAILED
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_FDN_RESTRICT
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_FLOW_SUSPENDED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_GENERAL
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_IFACE
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INCORRECT_PIN
qmerrno.h, [1368](#)

- eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_INJECT_TIMEOUT
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCE-
S
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTI-
ON
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTIO-
N
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, [1371](#)
- eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_MAX
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_I-
N_USE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_-
USE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAIL-
URE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTE-
D
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPOR-
TED
qmerrno.h, [1368](#)

- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_REQ
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_RSP
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, [1367](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, [1370](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, [1368](#)
- eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, [1369](#)
- eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, [1371](#)
- eQCWWAN_ERR_RESET
qmerrno.h, [1368](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_END
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_START
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, [1370](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIDCS_END
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIDCS_START
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIIM_END
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, [1371](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, [1371](#)

eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD-
 _MODE
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDE-
 X
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISM-
 ATCH
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT-
 _ACTIVE_IMAGE
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_N-
 OT_SIGNED
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_SECBOOT_INVALID_-
 CERT_CHAIN
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_TOO_MANY_FILES
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_INVALID_PATH
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_OPENING_DIR
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_OPENING_FILE
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWIIM_START
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWISM_END
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_F-
 OUND
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPT-
 ED
 qmerrno.h, [1371](#)
 eQCWWAN_ERR_SWISMS_START
 qmerrno.h, [1371](#)
 eQDL_HW_FLOW_DISABLE
 lite-fw.h, [1211](#)
 eQDL_HW_FLOW_ENABLE
 lite-fw.h, [1211](#)
 eQDL_HW_FLOW_INIT
 lite-fw.h, [1211](#)
 eQDL_HW_FLOW_Unknown
 lite-fw.h, [1211](#)
 eQDL_MODE_INIT
 lite-fw.h, [1211](#)
 eQDL_MODE_TTYUSB
 lite-fw.h, [1211](#)
 eQDL_MODE_UART0
 lite-fw.h, [1211](#)
 eQDL_MODE_Unknown
 lite-fw.h, [1211](#)
 eQMI_LOC_SESS_STATUS_FAILURE
 loc.h, [1223](#)
 eQMI_LOC_SESS_STATUS_IN_PROGRESS
 loc.h, [1223](#)
 eQMI_LOC_SESS_STATUS_SUCCESS
 loc.h, [1223](#)
 eQMI_LOC_SESS_STATUS_TIMEOUT
 loc.h, [1223](#)
 eQOS
 common.h, [1118](#)
 eREQ
 common.h, [1118](#)
 eRMS
 common.h, [1118](#)
 eRSP
 common.h, [1118](#)
 eSAR
 common.h, [1118](#)
 eSDP_FWDWL_ERR_END
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_FAIL
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_FW_UPGRADE
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_FW_VERSION_FAIL
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_GENERAL
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_INVALID_DEV
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_INVALID_PATH
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_PATH_TOO_LONG
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_PRI_FAIL
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_SDK
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_SDP_TIMEOUT
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_SET_CBK
 lite-fw.h, [1210](#)
 eSDP_FWDWL_ERR_TIMEOUT
 lite-fw.h, [1210](#)
 eSDP_FWDWL_SUCCESS
 lite-fw.h, [1210](#)

- eSMS
 - common.h, [1118](#)
- eSWIAUDIO
 - common.h, [1118](#)
- eSWIDMS
 - common.h, [1118](#)
- eSWILOC
 - common.h, [1118](#)
- eSWIM2MCMD
 - common.h, [1118](#)
- eSWIM2MCMD_AVC2
 - common.h, [1118](#)
- eSWIOMA
 - common.h, [1118](#)
- eSWIOMAEXT
 - common.h, [1118](#)
- eTIMEOUT_10_S
 - common.h, [1118](#)
- eTIMEOUT_20_S
 - common.h, [1118](#)
- eTIMEOUT_2_S
 - common.h, [1118](#)
- eTIMEOUT_300_S
 - common.h, [1118](#)
- eTIMEOUT_30_S
 - common.h, [1118](#)
- eTIMEOUT_5_S
 - common.h, [1118](#)
- eTIMEOUT_60_S
 - common.h, [1118](#)
- eTIMEOUT_8_S
 - common.h, [1118](#)
- eTIMEOUT_DEFAULT
 - common.h, [1118](#)
- eTMD
 - common.h, [1118](#)
- eUIM
 - common.h, [1118](#)
- eVOICE
 - common.h, [1118](#)
- eWDS
 - common.h, [1117](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID-
_IDENT_FOR_PROFILE
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT-
_DEFINED
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_
_PROFILES
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFIL-
_E_FAMILY
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_F-
LAG_SET
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_N-
OT_SET
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_END
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMI-
LY
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_HNDL
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_IDENT
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_OP
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_PROFILE_NUM
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_PROFILE_TYPE
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_SUBS_ID
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_IN-
VALID
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NO-
T_INITED
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL
 - qmerrno.h, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END
 - qmerrno.h, [1372](#)
- EAWTlv
 - unpack_ims_SLQSVolPCfgCallBack_ind_t, [703](#)
- ECIOThresListLen
 - nas_ECIOThresh, [195](#)
- ECTCallState
 - voice_ECTNum, [1064](#)
- eLOG_LEVEL
 - common.h, [1117](#)
- EMTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- eQCWWANError
 - qmerrno.h, [1367](#)
- eQMI_SVC
 - common.h, [1117](#)
- ERI_DATA_MAX_SIZE
 - dms.h, [1130](#)
- ESATlv
 - unpack_ims_SLQSVolPCfgCallBack_ind_t, [703](#)
- ESAWTlv

- unpack_ims_SLQSVolPCfgCallback_ind_t, 703
- ESNString
 - unpack_dms_GetDeviceSerialNumbers_t, 627
- ETWSPLMNInfo
 - eTWSPLMNInfoTlv, 78
- eTWSPLMNInfoTlv, 78
 - ETWSPLMNInfo, 78
 - TlvPresent, 78
- ETWSPLMNTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 863
- ETWSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 863
- eTimeout
 - common.h, 1118
- eValid
 - LibPackTFTIDParams, 143
- EarMute
 - pack_audio_SLQSSetAudioProfile_t, 347
 - pack_swiaudio_SLQSSetM2MAVMute_t, 462
 - unpack_audio_SLQSGetAudioProfile_t, 608
 - unpack_swiaudio_SLQSGetM2MAudioProfile_t, 879
 - unpack_swiaudio_SLQSGetM2MAVMute_t, 881
- earfcn
 - nas_infoInterFreq, 216
 - nas_LTEInfoIntraFreq, 227
 - nas_ltePCI, 235
 - nas_umtsLTENbrCell, 314
- earfcn0
 - nas_lteEARFCN, 221
- earfcn1
 - nas_lteEARFCN, 221
- earlyWakeupTime
 - dms_PSMEarlyWakeupTimeTlv, 68
- ecio
 - cdmaSSInfo, 53
 - hdrSSInfo, 83
 - nas_ecioListElement, 194
 - nas_rxInfo, 291
 - nas_UMTSInfo, 312
 - tdscdmaSigInfoExt, 574
- ecioDelta
 - nas_SLQSSignalStrengthsIndReq, 298
- ecioInfo
 - nas_SLQSSignalStrengthsInformation, 300
- ecioList
 - unpack_nas_SLQSGetSignalStrength_t, 775
- ecioListLen
 - unpack_nas_SLQSGetSignalStrength_t, 775
- ecioThresholdList
 - nas_SLQSSignalStrengthsIndReq, 298
- ecioThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, 298
- edrxCiotLteModeTlv
 - unpack_nas_SLQSNasEdrxChangeInfoCallback_ind_t, 788
- edrxCycleLengthTlv
 - unpack_nas_SLQSNasEdrxChangeInfoCallback_ind_t, 788
- edrxEnableTypeTlv
 - unpack_nas_SLQSNasEdrxChangeInfoCallback_ind_t, 788
- edrxEnabled
 - nas_EdrxEnableType, 197
- edrxPagingTimeWindowTlv
 - unpack_nas_SLQSNasEdrxChangeInfoCallback_ind_t, 788
- edrxPtw
 - nas_EdrxPagingTimeWindow, 197
- edrxRatType
 - nas_EdrxRatType, 198
- edrxRatTypeTlv
 - unpack_nas_SLQSNasEdrxChangeInfoCallback_ind_t, 788
- egprsSupp
 - nas_GSMSSysInfo, 208
- egprsSuppValid
 - nas_GSMSSysInfo, 208
- ehrpMTUSize
 - swidms_ehrpdMTUSizeTlv, 569
- elevation
 - loc_satelliteInfo, 162
- EmerMode
 - nas_EmerModeTlv, 199
 - NASEmergencyModeTlv, 324
- emmConnState
 - nas_LTEInfo, 224
- emmState
 - nas_LTEInfo, 224
- emmSubState
 - nas_LTEInfo, 224
- Enable
 - pack_swiaudio_SLQSSetM2MAudioLPBK_t, 459
- enable
 - pack_pds_SetPDSSState_t, 435
 - pack_qos_SLQSSetQosEventCallback_t, 446
 - pack_swidms_SLQSSwiDmsSetHWWatchdog_t, 468
 - swidms_SwiDmsGetHWWatchdog, 573
- EnableState
 - unpack_dms_PSMCfgChange_ind_t, 644
- enableState
 - dms_PSMEnableStateTlv, 69
- EnableStateInd
 - dms_PSMEnableStateIndTlv, 69
- enabled
 - pack_rms_SetSMSWake_t, 446
 - unpack_rms_GetSMSWake_t, 857
 - unpack_wds_GetMobileIPProfile_t, 986
- EnabledStatus
 - unpack_pds_SetPdsState_ind_t, 828
- EncryptProt
 - nas_protocolSubtypeElement, 272
- EncryptedPIN1
 - pack_uim_ChangePin_t, 480

- pack_uim_SetPinProtection_t, 482
- pack_uim_UnblockPin_t, 490
- EndPointType
 - qos_BindDataPortPeripheralEndPointID_t, 543
- EndProactiveSession
 - cat_EndProactiveSessionTlv, 51
- EngineState
 - unpack_pds_SLQSGetGPSStateInfo_t, 833
- engineState
 - unpack_loc_EngineState_Ind_t, 720
- eqmiCbKSetStatus
 - sms.h, 1390
- eriData
 - eriDataparams, 78
- eriDataLen
 - eriDataparams, 78
- eriDataparams, 77
 - eriData, 78
 - eriDataLen, 78
- eriFile
 - unpack_dms_SLQSGetERIFile_t, 666
- error
 - unpack_wds_GetLastMobileIPError_t, 983
- errorClass
 - unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 877
- ErrorCode
 - imsa_IMSRegStatusErrorCodeInfo, 98
- ErrorCodeData
 - imsa_RatHandoverStatusInfo, 100
- ErrorCodeLen
 - imsa_RatHandoverStatusInfo, 100
- errorMask
 - unpack_cat_SetCATEventCallback_t, 611
- errorRate
 - nas_errorRateListElement, 200
- errorRateInfo
 - nas_SLQSSignalStrengthsInformation, 300
- errorRateList
 - unpack_nas_SLQSGetSignalStrength_t, 775
- errorRateListLen
 - unpack_nas_SLQSGetSignalStrength_t, 775
- errorState
 - slotInf, 549
 - uim_slotInfo, 602
- esn
 - unpack_dms_GetSerialNumbers_t, 640
- esnSize
 - unpack_dms_GetDeviceSerialNumbers_t, 626
- EspSpi
 - unpack_qos_swiQosFilter_t, 849
- EtwsMessageInfo
 - sMSEtwsMessageTlv, 565
- event
 - dms_UimStatusTlv, 73
 - unpack_qos_SLQSSetQosPriEventCallback_ind_-t, 845
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 847
- event_Index
 - unpack_cat_SetCatEventCallback_ind_t, 610
- EventID
 - cat_commonEventTlv, 50
- EventLength
 - cat_commonEventTlv, 50
- eventMask
 - pack_cat_SetCATEventCallback_t, 350
 - pack_uim_SLQSUIEventRegister_t, 484
 - unpack_uim_SLQSUIEventRegister_t, 919
- eventRegister
 - pack_loc_EventRegister_t, 381
- eventType
 - unpack_swioma_SLQSOMADMAAlertCallback_ind_-t, 900
- evrcCapability
 - voice_prefVoiceSO, 1073
- executingImage
 - FMSImageIDEntries, 81
- exponent
 - unpack_qos_pktErrRate_t, 837
- extBit
 - voice_calledPartySubAdd, 1046
- extDispInfo
 - voice_extDispRecInfo, 1064
- extDispInfoLen
 - voice_extDispRecInfo, 1064
- ExtErrorCode
 - PackCreateProfileOut, 536
- extPowerState
 - pack_loc_SetExtPowerState_t, 381
- extendedErrorCode
 - unpack_wds_SLQSDelateProfile_t, 994
- FIRMWARE_BCHDATESIZE
 - lite-fw.h, 1206
- FLOAT
 - SwiDataTypes.h, 1433
- 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_swioama_SLQSOMADMSetSettings_t, [476](#)
 - pack_swioama_SLQSOMADMSetSettingsExt_t, [477](#)
 - unpack_swioama_SLQSOMADMGetSettings_t, [908](#)
- FOTAdownload
 - pack_swioama_SLQSOMADMSetSettings_t, [476](#)
 - pack_swioama_SLQSOMADMSetSettingsExt_t, [477](#)
 - unpack_swioama_SLQSOMADMGetSettings_t, [908](#)
- FUMOEnable
 - pack_swioama_SLQSOMADMSetSettingsExt_t, [477](#)
- facility
 - pack_dms_UIMGetControlKeyStatus_t, [365](#)
 - pack_dms_UIMSetControlKeyProtection_t, [366](#)
 - pack_dms_UIMUnblockControlKey_t, [367](#)
- facilityCk
 - pack_dms_UIMSetControlKeyProtection_t, [366](#)
 - pack_dms_UIMUnblockControlKey_t, [367](#)
- facilityState
 - pack_dms_UIMSetControlKeyProtection_t, [366](#)
 - unpack_dms_UIMGetControlKeyStatus_t, [681](#)
- FailErrCode
 - unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, [707](#)
- failureCount
 - FMSImageIdElement, [80](#)
- family
 - pack_wds_GetDefaultProfileNum_t, [511](#)
 - pack_wds_SetDefaultProfileNum_t, [518](#)
- feature
 - uim_depersonalizationInformation, [585](#)
 - uim_personalizationStatus, [593](#)
- fileID
 - uim_fileAttributes, [589](#)
 - uim_fileInfo, [590](#)
- fileIndex
 - pack_uim_ReadTransparent_t, [481](#)
 - pack_uim_SLQSUIMGetFileAttributes_t, [485](#)
- fileSize
 - uim_fileAttributes, [589](#)
- fileType
 - uim_fileAttributes, [589](#)
- fill_pack_ctx
 - common.h, [1119](#)
- fill_sdu_hdr
 - common.h, [1119](#)
- filterId
 - LibPackTFTIDParams, [143](#)
- fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
 - unpack_swiloc_SwiLocGetAutoStart_t, [899](#)
- fix_rate_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, [899](#)
- fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
 - unpack_swiloc_SwiLocGetAutoStart_t, [899](#)
- fix_type_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, [899](#)
- fixAccuracy
 - pack_pds_StartPDSTrackingSessionExt_t, [443](#)
- fixCount
 - pack_pds_StartPDSTrackingSessionExt_t, [443](#)
- fixInterval
 - pack_pds_StartPDSTrackingSessionExt_t, [443](#)
- fixTimeout
 - pack_pds_StartPDSTrackingSessionExt_t, [443](#)
- flags
 - sensorData_t, [546](#)
- flowLabel
 - LibPackTFTIDParams, [143](#)
- fms.h, [1182](#)
 - GetValidFwPriCombinations, [1184](#)
 - pack_fms_GetImagesPreference, [1184](#)
 - pack_fms_GetStoredImages, [1184](#)
 - pack_fms_SetImagesPreference, [1185](#)
 - unpack_fms_GetImagesPreference, [1185](#)
 - unpack_fms_GetStoredImages, [1185](#)
 - unpack_fms_SetImagesPreference, [1185](#)
- Forbidden
 - nas_QmiNas3GppNetworkInfo, [274](#)
- forbiddenNwInstLen
 - nas_ForbiddenNetworks3GPP, [201](#)
- forceFlag
 - pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, [438](#)
- ForceRev0
 - unpack_nas_GetCDMANetworkParameters_t, [745](#)
- format
 - sMSTransferRouteMTMessageInfo, [568](#)
- ForwardMac
 - nas_protocolSubtypeElement, [272](#)
- fqdnAddr
 - wds_PCSCFFQDNAddress, [1095](#)
- fqdnLen
 - wds_PCSCFFQDNAddress, [1095](#)
- freeSlots
 - sms_maxStorageSizeResp, [555](#)
- freq
 - nas_PhyCaAggPcellInfo, [262](#)
 - nas_PhyCaAggScellIndType, [263](#)
 - nas_PhyCaAggScellInfo, [266](#)
 - nas_QmisNasSlqsNasPCICellInfo, [276](#)
 - NASPhyCaAggPcellInfo, [330](#)
 - NASPhyCaAggScellArray, [332](#)
 - NASPhyCaAggScellIndType, [333](#)
 - NASPhyCaAggScellInfo, [334](#)
- freqsLen
 - nas_LTEInfoInterfreq, [225](#)
 - nas_LTEInfoNeighboringGSM, [228](#)
 - nas_LTEInfoNeighboringWCDMA, [229](#)

- fromServiceId
 - sms_BroadcastConfig, [550](#)
- fullPath
 - _litefw_FirmwareFileInfo, [36](#)
- fumoResultCode
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- fumoState
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- function
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
 - unpack_swiloc_SwiLocGetAutoStart_t, [899](#)
- function_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, [899](#)
- FwAutoCheck
 - unpack_swima_SLQSOMADMGetSettings_t, [908](#)
- FwAutoSDM
 - unpack_swiaavms_SLQSAVMSGetSettings_t, [885](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [888](#)
- FwAvailability
 - unpack_swima_SLQSOMADMStartSession_t, [909](#)
- FwPromptUpdate
 - unpack_swiaavms_SLQSAVMSGetSettings_t, [886](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [888](#)
- FwPromptdownload
 - unpack_swiaavms_SLQSAVMSGetSettings_t, [885](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [888](#)
- fwloadsize
 - unpack_omaDmFotaTlv_t, [819](#)
- fwloadComplete
 - unpack_omaDmFotaTlv_t, [819](#)
- fwvers
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [671](#)
- gDIBitRate
 - LibPackQosClassID, [142](#)
- GPRSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, [1007](#)
- GPSLPM
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- GPSSel
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- GSMRSSIthreshListLen
 - nas_GSMRSSIthresh, [205](#)
- GSMSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [796](#)
- gUIBitRate
 - LibPackQosClassID, [142](#)
- GWAOPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- GWAcqOrderPref
 - nas_GWAcqOrderPrefTlv, [209](#)
- NASGWAcqOrderPrefTlv, [325](#)
- GWAddressV4
 - unpack_wds_SLQSGetRuntimeSettings_t, [1007](#)
- gcdumpString
 - crashInformation, [55](#)
- gcdumpStrlen
 - crashInformation, [55](#)
- Generator
 - pack_audio_SLQSGetAudioProfile_t, [343](#)
 - pack_audio_SLQSGetAudioVoITLBConfig_t, [344](#)
 - pack_audio_SLQSSetAudioProfile_t, [347](#)
 - pack_audio_SLQSSetAudioVoITLBConfig_t, [348](#)
 - pack_swiaudio_SLQSGetM2MAudioVolume_t, [457](#)
 - pack_swiaudio_SLQSSetM2MAudioVolume_t, [461](#)
 - unpack_swiaudio_SLQSGetM2MAudioProfile_t, [879](#)
- geoSysIdx
 - nas_AddCDMASysInfo, [170](#)
 - nas_AddSysInfo, [170](#)
- geranArfcn
 - nas_geranInstInfo, [203](#)
- geranBsicBcc
 - nas_geranInstInfo, [203](#)
- geranBsicNcc
 - nas_geranInstInfo, [203](#)
- geranInst
 - nas_UMTSInfo, [312](#)
- GeranInstInfo
 - nas_UMTSInfo, [312](#)
- geranRssi
 - nas_geranInstInfo, [203](#)
- get_version
 - common.h, [1119](#)
- getAllCallInfo
 - voice_arrCallInfo, [1039](#)
- GetAllCallRmtPtyName
 - voice_arrRemotePartyName, [1041](#)
- GetCustomFeatureV2
 - unpack_dms_GetCustFeaturesV2_t, [618](#)
- GetValidFwPriCombinations
 - fms.h, [1184](#)
- glo_almanac_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- glo_ephemeris_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- glo_health_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- glo_visible_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- GlobalCellID
 - nas_QmisNasSlqsNasPCICellInfo, [276](#)
- globalCellId
 - nas_LTEInfoIntrafreq, [227](#)
- glog
 - common.h, [1120](#)
- gloglvl
 - common.h, [1120](#)
- gnssSvId

- loc_satelliteInfo, [162](#)
- gnssSvUsedList
 - loc_svUsedforFix, [165](#)
- gnssSvUsedList_len
 - loc_svUsedforFix, [165](#)
- Gpp2TimeZone
 - unpack_nas_SLQSGetServingSystem_t, [773](#)
- GppNetworkDSTAdjustment
 - unpack_nas_SLQSGetServingSystem_t, [773](#)
- GppTimeZone
 - unpack_nas_SLQSGetServingSystem_t, [773](#)
- gps_almanac_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- gps_ephemeris_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- gps_health_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- gps_visible_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- GpsEnable
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- gpsTimeOfWeekMs
 - loc_gpsTime, [154](#)
- gpsWeek
 - loc_gpsTime, [154](#)
- grntDownlinkBitrate
 - LibPackUMTSQoS, [146](#)
 - wds_UMTSMInQoS, [1101](#)
- grntUplinkBitrate
 - LibPackUMTSQoS, [146](#)
 - wds_UMTSMInQoS, [1102](#)
- gsmAmrStat
 - voice_curAMRConfig, [1061](#)
- GsmCellInfo
 - nas_lteGsmCellInfo, [222](#)
- gsmUmtsDI
 - pack_nas_SLQSNasSwiIndicationRegister_t, [413](#)
- gsmUmtsUI
 - pack_nas_SLQSNasSwiIndicationRegister_t, [413](#)
- guaranteedRate
 - unpack_qos_dataRate_t, [835](#)
- gwAddressV6
 - wds_IPV6GWAddressInfo, [1094](#)
- gwV6PrefixLen
 - wds_IPV6GWAddressInfo, [1094](#)
- gyroData
 - pack_loc_SLQSLOCInjectSensorData_t, [389](#)
- gyroTemp
 - pack_loc_SLQSLOCInjectSensorData_t, [389](#)
- gyroTimeSrc
 - pack_loc_SLQSLOCInjectSensorData_t, [390](#)
- HASPI
 - unpack_wds_GetMobileIPProfile_t, [986](#)
- HASState
 - unpack_wds_GetMobileIPProfile_t, [986](#)
- HDOP
 - loc_precisionDilution, [160](#)
- HDRECIOTreshListLen
 - nas_HDRECIOTresh, [210](#)
- HDRIOTreshListLen
 - nas_HDRIOTresh, [210](#)
- HDRRSSITreshListLen
 - nas_HDRRSSITresh, [211](#)
- HDRSINRThresListLen
 - nas_HDRSINRThresh, [212](#)
- HDRSINRThreshListLen
 - nas_HDRSINRThreshold, [213](#)
- HDRSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [796](#)
- HWVersion
 - nas_DeviceConfigDetail, [193](#)
- HardwareControlledMode
 - unpack_dms_GetPower_t, [638](#)
- has_LpmFlag
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [652](#)
- has_PdpStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, [378](#)
- has_PersistentLpm
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [652](#)
- has_PositionDataNMEA
 - unpack_pds_SetEventReport_Ind_t, [827](#)
- has_PowerOffMode
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [652](#)
- has_RatHandoverStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, [378](#)
- has_RegStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, [378](#)
- has_ServiceStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, [378](#)
- has_SessionStatus
 - unpack_pds_SetEventReport_Ind_t, [827](#)
- has_Wdisable
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [652](#)
- has_accelTemp
 - pack_loc_SLQSLOCInjectSensorData_t, [390](#)
- has_acceleroTimeSrc
 - pack_loc_SLQSLOCInjectSensorData_t, [390](#)
- has_accleroData
 - pack_loc_SLQSLOCInjectSensorData_t, [390](#)
- has_altitudeSrcInfo
 - pack_loc_SLQSLOCInjectPosition_t, [387](#)
- has_altitudeWrtEllipsoid
 - pack_loc_SLQSLOCInjectPosition_t, [387](#)
- has_altitudeWrtMeanSeaLevel
 - pack_loc_SLQSLOCInjectPosition_t, [387](#)
- has_confidence
 - pack_loc_SLQSLOCSetCradleMountConfig_t, [391](#)
- has_dLatitude
 - unpack_pds_SetEventReport_Ind_t, [827](#)
- has_dLongitude
 - unpack_pds_SetEventReport_Ind_t, [827](#)
- has_gyroData
 - pack_loc_SLQSLOCInjectSensorData_t, [390](#)
- has_gyroTemp
 - pack_loc_SLQSLOCInjectSensorData_t, [390](#)

- has_gyroTimeSrc
 - pack_loc_SLQSLOCInjectSensorData_t, 390
- has_horConfidence
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_horReliability
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_horUncCircular
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_latitude
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_longitude
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_opaqueId
 - pack_loc_SLQSLOCInjectSensorData_t, 390
- has_posSrc
 - unpack_pds_SetEventReport_Ind_t, 827
- has_positionSrc
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_rawHorConfidence
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_rawHorUncCircular
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_serverAddrTypeMask
 - pack_loc_SLQSLOCGetServer_t, 383
- has_timestampAge
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_timestampUtc
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_vertConfidence
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_vertReliability
 - pack_loc_SLQSLOCInjectPosition_t, 387
- has_vertUnc
 - pack_loc_SLQSLOCInjectPosition_t, 388
- hdrActiveProt
 - nas_HDRSysInfo, 215
- hdrActiveProtValid
 - nas_HDRSysInfo, 215
- hdrHybrid
 - nas_detailSvcInfo, 192
- HdrPersonality
 - unpack_nas_SLQSGetServingSystem_t, 773
- hdrPersonality
 - nas_HDRSysInfo, 215
 - NASServingSystemInfo, 341
- hdrPersonalityValid
 - nas_HDRSysInfo, 215
- hdrSSInfo, 82
 - ecio, 83
 - io, 83
 - rsi, 83
 - sinr, 83
- hdrSrvStatus
 - nas_detailSvcInfo, 192
- headerType
 - _litew_FirmwareFileInfo, 36
- healthStatus
 - loc_satelliteInfo, 162
- helper_get_error_code
 - common.h, 1119
- helper_get_error_reason
 - common.h, 1119
- helper_get_req_str
 - common.h, 1119
- helper_get_resp_ctx
 - common.h, 1119
- helper_get_xid
 - common.h, 1119
- helper_isBootLoader_DebugEnabled
 - common.h, 1119
- helper_set_log_func
 - common.h, 1120
- helper_set_log_lvl
 - common.h, 1120
- hfaMaxRetry
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext_t, 906
- hfaRetryIndex
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext_t, 906
- hfaRetryInterval
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext_t, 906
- hfaStatus
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext_t, 906
- homeOrigVoiceSO
 - voice_prefVoiceSO, 1073
- homePageVoiceSO
 - voice_prefVoiceSO, 1073
- horConfidence
 - pack_loc_SLQSLOCInjectPosition_t, 388
- horReliability
 - pack_loc_SLQSLOCInjectPosition_t, 388
- horUncCircular
 - pack_loc_SLQSLOCInjectPosition_t, 388
- HorizontalUncertainty
 - unpack_pds_SLQSGetGPSSStateInfo_t, 833
- hostID
 - pack_dms_SLQSSwiSetHostDevInfo_t, 362
 - unpack_dms_SLQSSwiGetHostDevInfo_t, 674
- hotSwap
 - uim_hotSwapStatus, 592
- hotSwapLength
 - uim_hotSwapStatus, 592
- hour
 - nas_timeInfo, 309
 - nas_UniversalTime, 315
- hrpdMTUSize
 - swidms_hrpdmTUSizeTlv, 570
- hsCallStatus
 - nas_WCDMASysInfo, 321
- hsCallStatusValid
 - nas_WCDMASysInfo, 321
- hsInd
 - nas_WCDMASysInfo, 321

- hsIndValid
 - nas_WCDMASysInfo, [321](#)
- hwType
 - wds_DHCPv4HWConfig, [1088](#)
 - wdsDhcpv4HwConfig, [1102](#)
- hwVer
 - unpack_dms_GetHardwareRevision_t, [632](#)
- IDTlv
 - unpack_ims_SLQSUerCfgCallBack_ind_t, [701](#)
- iFaceTab
 - swiaudio_PCMparams, [569](#)
- iFaceTabLen
 - swiaudio_PCMparams, [569](#)
- iLTEbandValue
 - nas_PhyCaAggPcellInfo, [262](#)
 - nas_PhyCaAggScellInfo, [266](#)
 - NASPhyCaAggPcellInfo, [330](#)
 - NASPhyCaAggScellInfo, [335](#)
- IMCNflag
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- IMEIString
 - unpack_dms_GetDeviceSerialNumbers_t, [627](#)
- IMG_MASK_CLEAR
 - lite-fw.h, [1206](#)
- IMG_MASK_GENERIC
 - lite-fw.h, [1206](#)
- IMG_MASK_MDM
 - lite-fw.h, [1206](#)
- IMG_MASK_PRI
 - lite-fw.h, [1206](#)
- IMSInfo
 - sMSOnIMSTlv, [567](#)
- IMSRegistration
 - unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, [708](#)
- IMSRegistrationError
 - unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, [708](#)
- IMSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [863](#)
- INT32
 - SwiDataTypes.h, [1433](#)
- INT8
 - SwiDataTypes.h, [1433](#)
- IOThresListLen
 - nas_IOTresh, [217](#)
- IPAddress
 - nas_DataStatusDetail, [190](#)
- IPAddressV6
 - ipv6AddressInfo, [104](#)
 - wds_IPV6AddressInfo, [1093](#)
- IPFamSupport
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- IPFamilyPreference
 - pack_wds_SLQSSetIPFamilyPreference_t, [529](#)
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- IPSECSPI
 - LibPackTFTIDParams, [143](#)
- IPv6AddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- IPv6GWAddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- IPv6PrefixLen
 - ipv6AddressInfo, [105](#)
 - wds_IPV6AddressInfo, [1093](#)
- IPv4
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- IPv4Addr
 - loc_IPv4Config, [156](#)
 - wds_IPv4AdTlv, [1093](#)
- IPv4AddrTlv
 - unpack_wds_DHCPv4ClientLease_ind_t, [973](#)
- IPv4DstAddr
 - unpack_qos_swqosFilter_t, [849](#)
- IPv4Port
 - loc_IPv4Config, [156](#)
- IPv4SrcAddr
 - unpack_qos_swqosFilter_t, [849](#)
- IPv4Tos
 - unpack_qos_swqosFilter_t, [849](#)
- IPv6Addr
 - loc_IPv6Config, [157](#)
- IPv6DstAddr
 - unpack_qos_swqosFilter_t, [849](#)
- IPv6Label
 - unpack_qos_swqosFilter_t, [849](#)
- IPv6Port
 - loc_IPv6Config, [157](#)
- IPv6SrcAddr
 - unpack_qos_swqosFilter_t, [849](#)
- IPv6TrafCls
 - unpack_qos_swqosFilter_t, [849](#)
- Id
 - unpack_qos_swqosFilter_t, [849](#)
- id
 - loc_BdsSV, [149](#)
 - loc_SV, [164](#)
 - nas_CSGID, [187](#)
 - pack_dms_UIMChangePIN_t, [364](#)
 - pack_dms_UIMSetPINProtection_t, [367](#)
 - pack_dms_UIMUnblockPIN_t, [368](#)
 - pack_dms_UIMVerifyPIN_t, [368](#)
 - unpack_qos_QosFlowInfoState_t, [839](#)
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, [847](#)
- id_length
 - DMScustSettingInfo, [75](#)
- IdleState
 - nas_protocolSubtypeElement, [272](#)
- IfaceID
 - qos_BindDataPortPeripheralEndPointID_t, [543](#)
- 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, 687
- ImageListSize
 - unpack_fms_GetImagesPreference_t, 686
- imageListSize
 - pack_fms_SetImagesPreference_t, 370
- imageMask
 - _litefw_FirmwareFileInfo, 36
- imageType
 - _litefw_FirmwareFileInfo, 36
 - FMSImageElement, 79
 - FMSImageIDEntries, 81
 - image_info_t, 84
- ImageTypes
 - unpack_fms_SetImagesPreference_t, 687
- ImageTypesSize
 - unpack_fms_SetImagesPreference_t, 687
- imagelistSize
 - unpack_fms_GetStoredImages_t, 687
- imei_no
 - unpack_dms_GetSerialNumbers_t, 640
- imeiSize
 - unpack_dms_GetDeviceSerialNumbers_t, 627
- imeiSvnSize
 - unpack_dms_GetDeviceSerialNumbers_t, 627
- ImeiSvnString
 - unpack_dms_GetDeviceSerialNumbers_t, 627
- imeisv_svn
 - unpack_dms_GetSerialNumbers_t, 640
- imgType
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 673
- ims.h, 1186
 - MAX_NAME_LEN, 1188
 - pack_ims_SLQSSetIMSSMSConfig, 1188
 - pack_ims_SLQSSetIMSUserConfig, 1188
 - pack_ims_SLQSSetIMSVoIPConfig, 1189
 - pack_ims_SLQSSetRegMgrConfig, 1189
 - pack_ims_SLQSSetSIPConfig, 1189
 - pack_ims_SLQSImsConfigIndicationRegister, 1190
 - pack_ims_SLQSSetIMSSMSConfig, 1190
 - pack_ims_SLQSSetIMSUserConfig, 1191
 - pack_ims_SLQSSetIMSVoIPConfig, 1191
 - pack_ims_SLQSSetRegMgrConfig, 1192
 - pack_ims_SLQSSetSIPConfig, 1192
 - unpack_ims_SLQSSetIMSSMSConfig, 1192
 - unpack_ims_SLQSSetIMSUserConfig, 1193
 - unpack_ims_SLQSSetIMSVoIPConfig, 1193
 - unpack_ims_SLQSSetRegMgrConfig, 1194
 - unpack_ims_SLQSSetSIPConfig, 1194
 - unpack_ims_SLQSImsConfigIndicationRegister, 1194
 - unpack_ims_SLQSImsConfigIndicationRegister_t, 1188
 - unpack_ims_SLQSSetRegMgrCfgCallBack_ind, 1195
 - unpack_ims_SLQSSetSIPCfgCallBack_ind, 1197
 - unpack_ims_SLQSSetMSCfgCallBack_ind, 1197
 - unpack_ims_SLQSSetIMSSMSConfig, 1195
 - unpack_ims_SLQSSetIMSUserConfig, 1195
 - unpack_ims_SLQSSetIMSVoIPConfig, 1196
 - unpack_ims_SLQSSetRegMgrConfig, 1196
 - unpack_ims_SLQSSetSIPConfig, 1196
 - unpack_ims_SLQSSetUserCfgCallBack_ind, 1197
 - unpack_ims_SLQSVoIPCfgCallBack_ind, 1198
- 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
- ImsFailErrCode
 - imsa_IMSFailErrCodeTlv, 97
- ImsRegErrCode
 - unpack_imsa_SLQSGetIMSARegStatus_t, 704
- imsRegState
 - nas_CommInfo, 186
- ImsRegStatus
 - imsa_NewIMSRegStatusInfo, 99
 - unpack_imsa_SLQSGetIMSARegStatus_t, 704
- ImsRegistered
 - imsa_IMSRegStatusInfo, 98
- imsTestMode
 - ims_IMSTestModelInfo, 89
- imsa.h, 1198
 - MAX_ERROR_CODE_LEN, 1199
 - pack_imsa_SLQSGetIMSARegStatus, 1199
 - pack_imsa_SLQSGetIMSAServiceStatus, 1200
 - pack_imsa_SLQSRegisterIMSAlndication, 1200
 - unpack_imsa_SLQSGetIMSARegStatus, 1201
 - unpack_imsa_SLQSGetIMSAServiceStatus, 1201
 - unpack_imsa_SLQSImsaPdpStatusCallBack_ind, 1202
 - unpack_imsa_SLQSImsaRatStatusCallBack_ind, 1202
 - unpack_imsa_SLQSImsaRegStatusCallBack_ind, 1202
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind, 1203
 - unpack_imsa_SLQSRegisterIMSAlndication, 1203
 - unpack_imsa_SLQSRegisterIMSAlndication_t, 1199
- imsa_IMSFailErrCodeTlv, 97
 - ImsFailErrCode, 97
 - TlvPresent, 97
- imsa_IMSRegStatusErrorInfo, 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, 632
- imsi_11_12

- nas_CDMASysInfoExt, [181](#)
- imsiM1112
 - nas_minBasedIMSI, [245](#)
- imsiMS1
 - nas_minBasedIMSI, [245](#)
- imsiMS2
 - nas_minBasedIMSI, [245](#)
- imsiT1112
 - nas_trueIMSI, [310](#)
- imsiTS1
 - nas_trueIMSI, [310](#)
- imsiTS2
 - nas_trueIMSI, [310](#)
- imsiTaddrNum
 - nas_trueIMSI, [310](#)
- InUse
 - nas_QmiNas3GppNetworkInfo, [274](#)
- InactTmr
 - ims_RTPRTCPInactTmrDurInfo, [92](#)
- includes_pcs_digit
 - nas_QmisNasPcsDigit, [276](#)
- index
 - pack_wds_GetMobileIPProfile_t, [513](#)
 - pack_wds_SetDefaultProfileNum_t, [518](#)
 - pack_wds_SetMobileIPProfile_t, [521](#)
 - unpack_qos_swiQosFilter_t, [849](#)
 - unpack_qos_swiQosFlow_t, [853](#)
 - unpack_wds_GetDefaultProfileNum_t, [979](#)
- index1xPri
 - uim_cardStatus, [583](#)
- index1xSec
 - uim_cardStatus, [583](#)
- indexGwPri
 - uim_cardStatus, [583](#)
- indexGwSec
 - uim_cardStatus, [583](#)
- Info
 - unpack_nas_SLQSNasSwiOTAMessageCallback-_ind_t, [805](#)
 - unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind_t, [806](#)
- InfoInterfreq
 - nas_LTEInfoInterfreq, [225](#)
- injectEnable
 - loc_accelAcceptReady, [148](#)
 - loc_accelTempAcceptReady, [149](#)
 - loc_gyroAcceptReady, [155](#)
 - loc_gyroTempAcceptReady, [155](#)
- injectSensorDataStatus
 - unpack_loc_InjectSensorDataCallback_Ind_t, [727](#)
- injectTimeSyncStatus
 - unpack_loc_InjectTimeSyncDataCallback_Ind_t, [728](#)
- insNmrCellInfo
 - nas_GERANInfo, [202](#)
- instancesSize
 - unpack_nas_GetRFInfo_t, [751](#)
- interval
 - pack_pds_SetPDSDDefaults_t, [435](#)
 - pack_pds_SetXTRAAutomaticDownload_t, [437](#)
 - pack_wds_SLQSSetWdsEventCallback_t, [530](#)
- intf
 - dms_UimStatusTlv, [73](#)
- Io
 - unpack_nas_SLQSSetSignalStrength_t, [775](#)
- io
 - hdrSSInfo, [83](#)
 - nas_SLQSSignalStrengthsInformation, [300](#)
- ioDelta
 - nas_SLQSSignalStrengthsIndReq, [299](#)
- lono_valid
 - unpack_pds_SLQSSetGPSSStateInfo_t, [833](#)
- ipAddress
 - pack_wds_SetDefaultProfile_t, [517](#)
- ipFamily
 - unpack_wds_SLQSSetPacketSrvStatusCallback_-_t, [1011](#)
- ipVersion
 - LibPackTFTIDParams, [144](#)
- ipaddr
 - unpack_wds_GetDefaultProfile_t, [978](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- ipaddrv6
 - unpack_wds_GetDefaultProfile_t, [979](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- ipv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- ipv4GWAddress
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- ipv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- ipv6AddressInfo, [104](#)
 - IPAddressV6, [104](#)
 - IPv6PrefixLen, [105](#)
- ipv6GWAddress
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- is856SysId
 - nas_HDRSysInfo, [215](#)
- is856SysIdValid
 - nas_HDRSysInfo, [215](#)
- is_DataRate_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_EspSpi_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_IPv4DstAddr_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_IPv4SrcAddr_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_IPv4Tos_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_IPv6DstAddr_Available
 - unpack_qos_swiQosFilter_t, [849](#)

- is_IPv6Label_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_IPv6SrcAddr_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_IPv6TrafCls_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_Id_Available
 - unpack_qos_swiQosFilter_t, [849](#)
- is_Jitter_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_Latency_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_LteBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, [661](#)
 - unpack_dms_SLQSGetBandCapabilityExt_t, [665](#)
- is_LteQci_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_MaxAllowedPktSz_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_MinPolicedPktSz_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_NxtHdrProto_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_PktErrRate_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_Precedence_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_ProfileId3GPP2_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_RxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, [838](#)
- is_TCPDstPort_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_TCPSrcPort_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_TdsBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, [661](#)
 - unpack_dms_SLQSGetBandCapabilityExt_t, [665](#)
- is_TokenBucket_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_TrafficClass_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_TranDstPort_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_TranSrcPort_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_TxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, [838](#)
- is_UDPDstPort_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_UDPSrcPort_Available
 - unpack_qos_swiQosFilter_t, [850](#)
- is_val_3GPP2Pri_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_val_3GPPImCn_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_val_3GPPResResidualBER_Available
 - unpack_qos_swiQosFlow_t, [853](#)
- is_val_3GPPSigInd_Available
 - unpack_qos_swiQosFlow_t, [854](#)
- is_val_3GPPTraHdlPri_Available
 - unpack_qos_swiQosFlow_t, [854](#)
- isEmpty
 - voice_getAllCallInformation, [1065](#)
- isInTraffic
 - nas_txInfo, [310](#)
- isModByCC
 - voice_SUPSInfo, [1078](#)
- isNewFlow
 - unpack_qos_QosFlowInfoState_t, [839](#)
- isPrefDataPath
 - nas_GSMSrvStatusInfo, [206](#)
 - nas_NR5GSerStatTlv, [255](#)
 - nas_SrvStatusInfo, [302](#)
- isRadioTuned
 - nas_rxInfo, [291](#)
 - nas_RxSigInfo, [292](#)
- isSysForbidden
 - nas_detailSvcInfo, [192](#)
 - nas_sysInfoCommon, [304](#)
- isSysForbiddenValid
 - nas_sysInfoCommon, [304](#)
- isSysPriMatch
 - nas_CDMASysInfo, [180](#)
 - nas_HDRSysInfo, [215](#)
- isSysPriMatchValid
 - nas_CDMASysInfo, [180](#)
 - nas_HDRSysInfo, [215](#)
- IsVoiceEnabled
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- iseUICC
 - uim_physlotInfo, [594](#)
- Item
 - pack_audio_SLQSGetAudioPathConfig_t, [343](#)
 - pack_audio_SLQSGetAudioVolTLBConfig_t, [344](#)
 - pack_audio_SLQSSetAudioVolTLBConfig_t, [348](#)
- Jitter
 - unpack_qos_swiQosFlow_t, [854](#)
- jtagAccessAllowed
 - unpack_swidms_SLQSSwiDmsGetSecureInfo_t, [894](#)
- KeyExchange
 - nas_protocolSubtypeElement, [272](#)
- LITE_SYS_SRV_DOMAIN_CAMPED
 - nas.h, [1253](#)
- LITE_SYS_SRV_DOMAIN_CS_ONLY
 - nas.h, [1253](#)
- LITE_SYS_SRV_DOMAIN_CS_PS
 - nas.h, [1253](#)
- LITE_SYS_SRV_DOMAIN_NO_SRV
 - nas.h, [1253](#)
- LITE_SYS_SRV_DOMAIN_PS_ONLY
 - nas.h, [1253](#)

- LITE_VOICE_SUPS_SRV_CLASS_DATA
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_DATA_CIRCUITS_YNC
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_DATA_CIRCUITS_YNC
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_FAX
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_NONE
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_PADACCESS
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_SMS
 - voice.h, [1492](#)
- LITE_VOICE_SUPS_SRV_CLASS_VOICE
 - voice.h, [1492](#)
- LITEQMI_QMI_CBK_PARAM_NOCHANGE
 - sms.h, [1390](#)
- LITEQMI_QMI_CBK_PARAM_RESET
 - sms.h, [1390](#)
- LITEQMI_QMI_CBK_PARAM_SET
 - sms.h, [1390](#)
- LBPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- LITE_MAX_PCOID_LIST
 - wds.h, [1525](#)
- LMCERTlv
 - unpack_wds_SLQSDUNCallInfoCallBack_ind_t, [996](#)
- LOCEVENTMASKGNSSSVINFO
 - loc.h, [1221](#)
- LOCEVENTMASKNMEA
 - loc.h, [1222](#)
- LOCEVENTMASKWIFIREQ
 - loc.h, [1223](#)
- LPCSTR
 - SwiDataTypes.h, [1433](#)
- LTEAttachProfile
 - unpack_wds_SLQSGet3GPPConfigItem_t, [997](#)
- LTEAttachProfileList
 - unpack_wds_SLQSGet3GPPConfigItem_t, [998](#)
- LTEAttachProfileListLen
 - pack_wds_SLQSSet3GPPConfigItem_t, [528](#)
 - unpack_wds_SLQSGet3GPPConfigItem_t, [998](#)
- LTEBandPref
 - nas_LTEBandPrefTlv, [219](#)
 - NASLTEBandPreferenceTlv, [326](#)
- LTECphyCAInfo
 - unpack_nas_SlqsGetLTECphyCAInfo_t, [764](#)
- LTERSRRPThreshListLen
 - nas_LTERSRRPThresh, [236](#)
- LTERSRRQThreshListLen
 - nas_LTERSRRQThresh, [237](#)
- LTERSSIThreshListLen
 - nas_LTERSSIThresh, [237](#)
- LTESNRThreshListLen
 - nas_LTESNRThresh, [240](#)
- LTESNRThreshListLen
 - nas_LTESNRThreshold, [241](#)
- LTESSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [796](#)
- LTEWCDMACellInfo
 - nas_LTEInfoNeighboringWCDMA, [229](#)
- Lac
 - unpack_nas_SLQSGetServingSystem_t, [773](#)
- lac
 - nas_GERANInfo, [202](#)
 - nas_GSMsSysInfo, [208](#)
 - nas_LTEsSysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [258](#)
 - nas_UMTSInfo, [312](#)
 - nas_WCDMAsSysInfo, [321](#)
- lac1
 - nas_OperatorPLMNData, [260](#)
- lac2
 - nas_OperatorPLMNData, [260](#)
- lacValid
 - nas_GSMsSysInfo, [208](#)
 - nas_LTEsSysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [258](#)
 - nas_WCDMAsSysInfo, [321](#)
- language
 - sms_CDMABroadcastConfig, [551](#)
- lastCallDataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, [1003](#)
- lastCallDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, [1001](#)
- lastCallRXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, [1003](#)
- lastCallTXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, [1003](#)
- LastErrCode
 - nas_DataStatusDetail, [190](#)
- Latency
 - unpack_qos_swiQosFlow_t, [854](#)
- Latitude
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- latitude
 - pack_loc_SLQSLOCInjectPosition_t, [388](#)
- leapSeconds
 - nas_qaQmi3Gpp2TimeZone, [273](#)
- leaseState
 - wds_DHCPLeaseStateTlv, [1086](#)
- len
 - loc_BdsSVInfo, [150](#)
 - loc_SVInfo, [164](#)
 - unpack_nas_GetSignalStrengths_t, [755](#)
- length
 - sMSCAddressInfo, [563](#)
 - sMSEtwSMessageInfo, [564](#)
 - sMSTransferRouteMTMessageInfo, [568](#)

- uim_readTransparentInfo, 595
- Level
 - pack_swiaudio_SLQSSetM2MAudioVolume_t, 461
 - unpack_swiaudio_SLQSGetM2MAudioVolume_t, 880
- 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
 - pIPv6AddrPref, 134
 - pImCnFlag, 134
 - pPDNInactivTimeout, 134
 - pPDPTtype, 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, [1207](#)
- libSDP_CalculateImageMask
 - lite-fw.h, [1207](#)
- libSDP_CheckValidFirmwareInfo
 - lite-fw.h, [1207](#)
- libSDP_DownloadFW
 - lite-fw.h, [1207](#)
- libSDP_ExtractFirmwareParametersByPath
 - lite-fw.h, [1207](#)
- libSDP_FirmwareInfo
 - lite-fw.h, [1207](#)
- libSDP_GetModelFamily
 - lite-fw.h, [1207](#)
- libSDP_GetVersion
 - lite-fw.h, [1207](#)
- libSDP_getFileType
 - lite-fw.h, [1207](#)
- 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, [112](#)
 - 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](#)
 - plpcpAckTimeout, [118](#)
 - plpcpCreqRetryCount, [118](#)
 - plsPcscfAddressNedded, [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](#)
 - pGPRSMinimumQoS, [127](#)
 - pGPRSRequestedQos, [127](#)
 - pIPv6DelegFlag, [128](#)
 - pIPv4AddrPref, [128](#)
 - pIPv6AddPref, [128](#)

- pIWLANtoLTEHandoverFlag, 128
- pImCnFlag, 128
- pLTetoIWLANHandoverFlag, 128
- pLteRoamPDPTYPE, 128
- pMaxPDN, 128
- pMaxPDNTimer, 128
- pMcc, 128
- pMnc, 128
- pMsisdnFlag, 128
- pOperatorPCOID, 128
- pOverridePDPTYPE, 128
- pPCOIDList, 128
- pPDNDisconnectWaitTimer, 128
- pPDNInactivTimeout, 128
- pPDNThrottleTimer, 128
- pPDNWaitTimer, 128
- pPDPtype, 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
- pUmtsRoamPDPTYPE, 129
- pUsername, 129
- pUsernameSize, 129
- libsdp_SetReadBlockSize
 - lite-fw.h, 1207
- libsdp_set_log_func
 - lite-fw.h, 1207
- libsdplogger
 - lite-fw.h, 1207
- lineValue
 - pack_voice_SLQSVoiceALSSelectLine_t, 493
- linkage
 - altSrcInfo_t, 39
- list_type
 - DMScustSettingList, 75
 - DMSgetCustomInput, 77
 - pack_dms_GetCustFeaturesV2_t, 351
- listEntries
 - FMSPrefImageList, 82
- listSize
 - FMSImageList, 81
 - FMSPrefImageList, 82
- lite-fw.h
 - eFILE_TYPE_CAR_PRI, 1209
 - eFILE_TYPE_COMPO_PRI, 1209
 - eFILE_TYPE_NONE, 1209
 - eFILE_TYPE_OEM_PRI, 1209
 - eFIREHOSE_ERR_SECBOOT_INVALID_CERT_-CHAIN, 1210
 - eFW_TYPE_CWE, 1209
 - eFW_TYPE_CWE_NVU, 1210
 - eFW_TYPE_INVALID, 1209
 - eFW_TYPE_MBN, 1209
 - eFW_TYPE_MBN_GOBI, 1209
 - eFW_TYPE_NVU, 1209
 - eFW_TYPE_SPK, 1209
 - eIMAGE_TYPE_ANY, 1210
 - eIMAGE_TYPE_APPL, 1210
 - eIMAGE_TYPE_APPS, 1210
 - eIMAGE_TYPE_BOOT, 1210
 - eIMAGE_TYPE_FILE, 1210
 - eIMAGE_TYPE_INVALID, 1210
 - eIMAGE_TYPE_KEYS, 1210
 - eIMAGE_TYPE_MAX, 1210
 - eIMAGE_TYPE_MIN, 1210
 - eIMAGE_TYPE_MODM, 1210
 - eIMAGE_TYPE_SPKG, 1210
 - eIMAGE_TYPE_USER, 1210
 - eModel_9X15, 1210
 - eModel_9X30, 1210
 - eModel_9x06, 1210
 - eModel_9x07, 1210
 - eModel_Unknown, 1210
 - eModel_WP9X15, 1210
 - eQDL_HW_FLOW_DISABLE, 1211
 - eQDL_HW_FLOW_ENABLE, 1211
 - eQDL_HW_FLOW_INIT, 1211
 - eQDL_HW_FLOW_Unknown, 1211
 - eQDL_MODE_INIT, 1211
 - eQDL_MODE_TTYUSB, 1211
 - eQDL_MODE_UART0, 1211
 - eQDL_MODE_Unknown, 1211
 - eSDP_FWDWL_ERR_END, 1210
 - eSDP_FWDWL_ERR_FAIL, 1210
 - eSDP_FWDWL_ERR_FW_UPGRADE, 1210
 - eSDP_FWDWL_ERR_FW_VERSION_FAIL, 1210
 - eSDP_FWDWL_ERR_GENERAL, 1210
 - eSDP_FWDWL_ERR_INVALID_DEV, 1210
 - eSDP_FWDWL_ERR_INVALID_PATH, 1210
 - eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED, 1210
 - eSDP_FWDWL_ERR_PATH_TOO_LONG, 1210
 - eSDP_FWDWL_ERR_PRI_FAIL, 1210

- eSDP_FWDWL_ERR_SDK, [1210](#)
- eSDP_FWDWL_ERR_SDP_TIMEOUT, [1210](#)
- eSDP_FWDWL_ERR_SET_CBK, [1210](#)
- eSDP_FWDWL_ERR_TIMEOUT, [1210](#)
- eSDP_FWDWL_SUCCESS, [1210](#)
- lite-fw.h, [1203](#)
 - FIRMWARE_BCHDATESIZE, [1206](#)
 - IMG_MASK_CLEAR, [1206](#)
 - IMG_MASK_GENERIC, [1206](#)
 - IMG_MASK_MDM, [1206](#)
 - IMG_MASK_PRI, [1206](#)
 - libSDP_BuildImagesPreferenceRequest, [1207](#)
 - libSDP_CalculateImageMask, [1207](#)
 - libSDP_CheckValidFirmwareInfo, [1207](#)
 - libSDP_DownloadFW, [1207](#)
 - libSDP_ExtractFirmwareParametersByPath, [1207](#)
 - libSDP_FirmwareInfo, [1207](#)
 - libSDP_GetModelFamily, [1207](#)
 - libSDP_GetVersion, [1207](#)
 - libSDP_getFileType, [1207](#)
 - libsdp_SetReadBlockSize, [1207](#)
 - libsdp_set_log_func, [1207](#)
 - libsdplogger, [1207](#)
 - litefw_BuildImagesPreferenceRequest, [1211](#)
 - litefw_CalculateImageMask, [1211](#)
 - litefw_CheckValidFirmwareInfo, [1212](#)
 - litefw_DownloadFW, [1212](#)
 - litefw_ExtractFirmwareParametersByPath, [1212](#)
 - litefw_ExtractFirmwarePartNoByPath, [1213](#)
 - litefw_FirmwareFileInfo, [1207](#)
 - litefw_FirmwareInfo, [1208](#)
 - litefw_FirmwarePartNo, [1209](#)
 - litefw_Fw_Type, [1209](#)
 - litefw_GetModelFamily, [1214](#)
 - litefw_GetQTLDownloadMode, [1214](#)
 - litefw_GetQTLHWFlowControl, [1214](#)
 - litefw_GetVersion, [1214](#)
 - litefw_Models, [1210](#)
 - litefw_QDL_FLOW_CONTROLS, [1210](#)
 - litefw_QDL_MODES, [1211](#)
 - litefw_SLQSGetFirmwareFileInfo, [1216](#)
 - litefw_SetQTLDownloadMode, [1215](#)
 - litefw_SetQTLHWFlowControl, [1215](#)
 - litefw_SetReadBlockSize, [1215](#)
 - litefw_fileimgtype, [1209](#)
 - litefw_fwdwl_error_codes, [1210](#)
 - litefw_getFileType, [1213](#)
 - litefw_imagetype, [1210](#)
 - litefw_logsenable, [1214](#)
 - litefw_set_log_func, [1215](#)
 - litefw_switch_9x07_to_downloadmode, [1216](#)
 - litefw_switch_to_BootHoldMode, [1216](#)
 - litefwlogger, [1209](#)
- liteQmiDataBearerMasks
 - wds.h, [1527](#)
- liteServiceClassInformation
 - voice.h, [1491](#)
- litefw_BuildImagesPreferenceRequest
 - lite-fw.h, [1211](#)
- litefw_CalculateImageMask
 - lite-fw.h, [1211](#)
- litefw_CheckValidFirmwareInfo
 - lite-fw.h, [1212](#)
- litefw_DownloadFW
 - lite-fw.h, [1212](#)
- litefw_ExtractFirmwareParametersByPath
 - lite-fw.h, [1212](#)
- litefw_ExtractFirmwarePartNoByPath
 - lite-fw.h, [1213](#)
- litefw_FirmwareFileInfo
 - lite-fw.h, [1207](#)
- litefw_FirmwareInfo
 - lite-fw.h, [1208](#)
- litefw_FirmwarePartNo
 - lite-fw.h, [1209](#)
- litefw_Fw_Type
 - lite-fw.h, [1209](#)
- litefw_GetModelFamily
 - lite-fw.h, [1214](#)
- litefw_GetQTLDownloadMode
 - lite-fw.h, [1214](#)
- litefw_GetQTLHWFlowControl
 - lite-fw.h, [1214](#)
- litefw_GetVersion
 - lite-fw.h, [1214](#)
- litefw_Models
 - lite-fw.h, [1210](#)
- litefw_QDL_FLOW_CONTROLS
 - lite-fw.h, [1210](#)
- litefw_QDL_MODES
 - lite-fw.h, [1211](#)
- litefw_SLQSGetFirmwareFileInfo
 - lite-fw.h, [1216](#)
- litefw_SetQTLDownloadMode
 - lite-fw.h, [1215](#)
- litefw_SetQTLHWFlowControl
 - lite-fw.h, [1215](#)
- litefw_SetReadBlockSize
 - lite-fw.h, [1215](#)
- litefw_fileimgtype
 - lite-fw.h, [1209](#)
- litefw_fwdwl_error_codes
 - lite-fw.h, [1210](#)
- litefw_getFileType
 - lite-fw.h, [1213](#)
- litefw_imagetype
 - lite-fw.h, [1210](#)
- litefw_logsenable
 - lite-fw.h, [1214](#)
- litefw_set_log_func
 - lite-fw.h, [1215](#)
- litefw_switch_9x07_to_downloadmode
 - lite-fw.h, [1216](#)
- litefw_switch_to_BootHoldMode
 - lite-fw.h, [1216](#)
- litefwlogger

- lite-fw.h, [1209](#)
- liteqmi_GetVersion
 - common.h, [1120](#)
- liteqmi_helper_decode7bitAsciiEncString
 - common.h, [1120](#)
- liteqmi_log
 - common.h, [1120](#)
- loc.h
 - eQMI_LOC_SESS_STATUS_FAILURE, [1223](#)
 - eQMI_LOC_SESS_STATUS_IN_PROGRESS, [1223](#)
 - eQMI_LOC_SESS_STATUS_SUCCESS, [1223](#)
 - eQMI_LOC_SESS_STATUS_TIMEOUT, [1223](#)
- loc.h, [1217](#)
 - LOCEVENTMASKNMEA, [1222](#)
 - LOCEVENTMASKWIFIREQ, [1223](#)
 - MAX_TEMP_DATA_LEN, [1223](#)
 - pack_loc_DeleteAssistData, [1224](#)
 - pack_loc_EventRegister, [1224](#)
 - pack_loc_SLQSLOCGetBestAvailPos, [1225](#)
 - pack_loc_SLQSLOCGetOpMode, [1225](#)
 - pack_loc_SLQSLOCGetServer, [1226](#)
 - pack_loc_SLQSLOCInjectPosition, [1226](#)
 - pack_loc_SLQSLOCInjectSensorData, [1227](#)
 - pack_loc_SLQSLOCInjectUTCTime, [1227](#)
 - pack_loc_SLQSLOCSetCradleMountConfig, [1227](#)
 - pack_loc_SLQSLOCSetServer, [1228](#)
 - pack_loc_SetExtPowerState, [1224](#)
 - pack_loc_SetOperationMode, [1225](#)
 - pack_loc_Start, [1228](#)
 - pack_loc_Stop, [1229](#)
 - unpack_loc_BestAvailPos_Ind, [1229](#)
 - unpack_loc_CradleMountCallback_Ind, [1229](#)
 - unpack_loc_DeleteAssistData, [1230](#)
 - unpack_loc_DeleteAssistData_Ind, [1230](#)
 - unpack_loc_DeleteAssistData_t, [1223](#)
 - unpack_loc_EngineState_Ind, [1230](#)
 - unpack_loc_EventNMEA_Ind, [1231](#)
 - unpack_loc_EventRegister, [1231](#)
 - unpack_loc_EventTimeSyncCallback_Ind, [1231](#)
 - unpack_loc_GetOpMode_Ind, [1232](#)
 - unpack_loc_GetServer_Ind, [1232](#)
 - unpack_loc_GnssSvInfo_Ind, [1232](#)
 - unpack_loc_InjectPositionCallback_Ind, [1233](#)
 - unpack_loc_InjectSensorDataCallback_Ind, [1233](#)
 - unpack_loc_InjectTimeSyncDataCallback_Ind, [1233](#)
 - unpack_loc_InjectUTCTimeCallback_Ind, [1234](#)
 - unpack_loc_PositionRpt_Ind, [1234](#)
 - unpack_loc_SLQSLOCGetBestAvailPos, [1236](#)
 - unpack_loc_SLQSLOCGetOpMode, [1237](#)
 - unpack_loc_SLQSLOCGetServer, [1237](#)
 - unpack_loc_SLQSLOCGetServer_t, [1223](#)
 - unpack_loc_SLQSLOCInjectPosition, [1237](#)
 - unpack_loc_SLQSLOCInjectPosition_t, [1223](#)
 - unpack_loc_SLQSLOCInjectSensorData, [1238](#)
 - unpack_loc_SLQSLOCInjectSensorData_t, [1223](#)
 - unpack_loc_SLQSLOCInjectUTCTime, [1238](#)
 - unpack_loc_SLQSLOCInjectUTCTime_t, [1223](#)
 - unpack_loc_SLQSLOCSetCradleMountConfig, [1238](#)
 - unpack_loc_SLQSLOCSetCradleMountConfig_t, [1223](#)
 - unpack_loc_SLQSLOCSetServer, [1239](#)
 - unpack_loc_SensorStreamingCallback_Ind, [1234](#)
 - unpack_loc_SetExtPowerConfig_Ind, [1235](#)
 - unpack_loc_SetExtPowerState, [1235](#)
 - unpack_loc_SetOperationMode, [1235](#)
 - unpack_loc_SetOperationMode_Ind, [1236](#)
 - unpack_loc_SetServer_Ind, [1236](#)
 - unpack_loc_Start, [1239](#)
 - unpack_loc_Stop, [1239](#)
- loc_BdsSV, [149](#)
 - id, [149](#)
 - mask, [149](#)
- loc_BdsSVInfo, [149](#)
 - len, [150](#)
 - pSV, [150](#)
- loc_CellDb, [150](#)
 - mask, [150](#)
- loc_ClkInfo, [150](#)
 - mask, [151](#)
- loc_GnssData, [152](#)
 - mask, [153](#)
- loc_IPv4Config, [156](#)
 - IPv4Addr, [156](#)
 - IPv4Port, [156](#)
- loc_IPv4Info, [156](#)
 - address, [157](#)
 - port, [157](#)
 - TlvPresent, [157](#)
- loc_IPv6Config, [157](#)
 - IPv6Addr, [157](#)
 - IPv6Port, [157](#)
- loc_IPv6Info, [157](#)
 - address, [158](#)
 - port, [158](#)
 - TlvPresent, [158](#)
- loc_LocApplicationInfo, [158](#)
 - appNameLength, [159](#)
 - appProviderLength, [159](#)
 - appVersionLength, [159](#)
 - appVersionValid, [159](#)
 - pAppName, [159](#)
 - pAppProvider, [159](#)
 - pAppVersion, [159](#)
- loc_SV, [163](#)
 - id, [164](#)
 - mask, [164](#)
 - system, [164](#)
- loc_SVInfo, [164](#)
 - len, [164](#)
 - pSV, [164](#)
- loc_URLAddrInfo, [165](#)
 - urlAddr, [166](#)
- loc_accelAcceptReady, [147](#)

- batchPerSec, [148](#)
- injectEnable, [148](#)
- samplesPerBatch, [148](#)
- loc_accelTempAcceptReady, [148](#)
 - batchPerSec, [149](#)
 - injectEnable, [149](#)
 - samplesPerBatch, [149](#)
- loc_gpsTime, [153](#)
 - gpsTimeOfWeekMs, [154](#)
 - gpsWeek, [154](#)
- loc_gyroAcceptReady, [154](#)
 - batchPerSec, [155](#)
 - injectEnable, [155](#)
 - samplesPerBatch, [155](#)
- loc_gyroTempAcceptReady, [155](#)
 - batchPerSec, [155](#)
 - injectEnable, [155](#)
 - samplesPerBatch, [155](#)
- loc_precisionDilution, [159](#)
 - HDOP, [160](#)
 - PDOP, [160](#)
 - VDOP, [160](#)
- loc_satelliteInfo, [160](#)
 - azimuth, [162](#)
 - elevation, [162](#)
 - gnssSvId, [162](#)
 - healthStatus, [162](#)
 - snr, [162](#)
 - svInfoMask, [162](#)
 - svListLen, [162](#)
 - svStatus, [162](#)
 - system, [162](#)
 - validMask, [162](#)
- loc_sensorDataUsage, [162](#)
 - aidingIndicatorMask, [163](#)
 - usageMask, [163](#)
- loc_svUsedforFix, [164](#)
 - gnssSvUsedList, [165](#)
 - gnssSvUsedList_len, [165](#)
- loc_urlAddr, [165](#)
 - address, [165](#)
 - TlvPresent, [165](#)
- localTimeOffset
 - nas_qaQmi3Gpp2TimeZone, [273](#)
- logString
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, [673](#)
- logger
 - common.h, [1117](#)
- longName
 - nas_PLMNNetworkNameData, [270](#)
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- longNameCI
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- longNameEn
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- longNameLen
 - nas_PLMNNetworkNameData, [270](#)
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- longNameSB
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- longNameSpareBits
 - nas_PLMNNetworkNameData, [270](#)
- Longitude
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- longitude
 - pack_loc_SLQSLOCInjectPosition_t, [388](#)
- loopbackMode
 - pack_wds_SLQSSSetLoopback_t, [531](#)
- loopbackMultiplier
 - pack_wds_SLQSSSetLoopback_t, [531](#)
- LpmFlag
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [653](#)
- LteBandCapability
 - unpack_dms_SLQSGetBandCapability_t, [661](#)
 - unpack_dms_SLQSGetBandCapabilityExt_t, [665](#)
- LteBands
 - dms_LteBandsSupport, [65](#)
- LteBandsSupport
 - unpack_dms_SLQSGetBandCapabilityExt_t, [665](#)
- LteEmmDI
 - pack_nas_SLQSNasSwiIndicationRegister_t, [413](#)
- LteEmmUI
 - pack_nas_SLQSNasSwiIndicationRegister_t, [413](#)
- LteEsmDI
 - pack_nas_SLQSNasSwiIndicationRegister_t, [413](#)
- LteEsmUI
 - pack_nas_SLQSNasSwiIndicationRegister_t, [413](#)
- LteGsmCellInfo
 - nas_LTEInfoNeighboringGSM, [228](#)
- LteM1BandPref
 - NASLteM1BandPrefTlv, [326](#)
- LteM1BandPref
 - nas_LteM1BandPrefTlv, [230](#)
- LteNB1BandPref
 - NASLteNB1BandPrefTlv, [327](#)
- LteNb1BandPref
 - nas_LteNb1BandPrefTlv, [232](#)
- LteOpMode
 - nas_EdrxCiotLteMode, [196](#)
 - nas_LteOpMode, [234](#)
 - nas_LteOpModeTlv, [235](#)
- LteOpModeLen
 - nas_LteOpModeTlv, [235](#)
- LteQci
 - unpack_qos_swiQosFlow_t, [854](#)
- LteRsrpDelta
 - nas_SLQSSignalStrengthsIndReq, [299](#)
- LteRsrpinfo
 - nas_SLQSSignalStrengthsInformation, [300](#)
- LteSSInfo, [166](#)
 - rsrp, [166](#)
 - rsrq, [167](#)
 - rsqi, [167](#)
 - snr, [167](#)
- LteSnrDelta
 - nas_SLQSSignalStrengthsIndReq, [299](#)

- lteSnrinfo
 - nas_SLQSSignalStrengthsInformation, [300](#)
- ltersrp
 - unpack_nas_SLQSGetSignalStrength_t, [776](#)
- ltesnr
 - unpack_nas_SLQSGetSignalStrength_t, [776](#)
- lvi
 - unpack_tmd_SLQSTmdMitigationLvIRptCallback_ind_t, [911](#)
- M1BandPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- m_FwBuildId
 - CarrierImage_t, [48](#)
- m_FwImageld
 - CarrierImage_t, [48](#)
- m_PriBuildId
 - CarrierImage_t, [48](#)
- m_Prilmageld
 - CarrierImage_t, [49](#)
- m_nCarrierId
 - CarrierImage_t, [48](#)
- m_nFolderId
 - CarrierImage_t, [48](#)
- m_nStorage
 - CarrierImage_t, [48](#)
- MACIndex
 - nas_NetworkStatEVDO, [252](#)
- MAX_ATR_LENGTH
 - uim.h, [1468](#)
- MAX_BUILD_ID_LEN
 - dms.h, [1130](#)
- MAX_ERROR_CODE_LEN
 - imsa.h, [1199](#)
- MAX_ICCID_LENGTH
 - uim.h, [1468](#)
- MAX_MSE_TWS_MSG
 - sms.h, [1389](#)
- MAX_NAME_LEN
 - ims.h, [1188](#)
- MAX_NO_OF_SLOTS
 - uim.h, [1468](#)
- MAX_PHY_SLOTS_INFO
 - uim.h, [1468](#)
- MAX_SENSOR_DATA_LEN
 - loc.h, [1223](#)
- MAX_SLOTS_STATUS
 - uim.h, [1468](#)
- MAX_SMS_LIST_SIZE
 - sms.h, [1389](#)
- MAX_TEMP_DATA_LEN
 - loc.h, [1223](#)
- MAXVOICEUSSDLENGTH
 - voice.h, [1491](#)
- MCC
 - nas_CDMA SysInfo, [180](#)
 - nas_CDMA SysInfoExt, [181](#)
 - nas_currentPLMN, [188](#)
 - nas_ForbiddenNetworks3GPP, [201](#)
 - nas_GSM SysInfo, [208](#)
 - nas_IteOpModeTlv, [235](#)
 - nas_LTE SysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [258](#)
 - nas_QmiNas3GppNetworkInfo, [274](#)
 - nas_QmiNas3GppNetworkRAT, [275](#)
 - nas_QmisNasPcsDigit, [276](#)
 - nas_WCDMA SysInfo, [321](#)
 - unpack_nas_GetServingNetwork_t, [753](#)
- MDMCallDuration
 - connectionStatus, [53](#)
- MDMConnStatus
 - connectionStatus, [53](#)
 - wds_ConnStatusTlv, [1082](#)
- MEID_MAX_SIZE
 - dms.h, [1130](#)
- MEIDString
 - unpack_dms_GetDeviceSerialNumbers_t, [627](#)
- MIN
 - unpack_dms_GetVoiceNumber_t, [643](#)
- MINREQBKLEN
 - common.h, [1116](#)
- MMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [863](#)
- MNC
 - LibPackProfileMnc, [141](#)
 - nas_CDMA SysInfo, [180](#)
 - nas_currentPLMN, [188](#)
 - nas_ForbiddenNetworks3GPP, [201](#)
 - nas_GSM SysInfo, [208](#)
 - nas_IteOpModeTlv, [235](#)
 - nas_LTE SysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [258](#)
 - nas_QmiNas3GppNetworkInfo, [274](#)
 - nas_QmiNas3GppNetworkRAT, [275](#)
 - nas_QmisNasPcsDigit, [276](#)
 - nas_WCDMA SysInfo, [321](#)
 - unpack_nas_GetServingNetwork_t, [753](#)
- MPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- MSETlv
 - unpack_ims_SLQSVolPCfgCallBack_ind_t, [703](#)
- MSGID_AND_LEN
 - common.h, [1116](#)
- MSGID_DONT_CARE
 - common.h, [1116](#)
- MTMessageInfo
 - newMTMessageTlv, [342](#)
- MTUSize
 - pack_swidms_SLQSSwiDmsSetMTU_t, [469](#)
- MTUSize3gpp
 - swidms_mtuSize3gppTlv, [572](#)
- manString
 - pack_dms_SLQSSwiSetHostDevInfo_t, [362](#)
 - unpack_dms_SLQSSwiGetHostDevInfo_t, [674](#)
- manufacturer
 - unpack_dms_GetManufacturer_t, [633](#)
- Mask

- pack_wds_SLQSGetDUNCallInfo_t, 524
- mask
 - loc_BdsSV, 149
 - loc_CellDb, 150
 - loc_ClkInfo, 151
 - loc_GnssData, 153
 - loc_SV, 164
 - unpack_qos_IPv6TrafCls_t, 836
 - unpack_qos_Tos_t, 855
- max
 - PackSwiAVMSSettingsPeriodsInfo, 541
- max_channel_rx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 1000
- max_channel_tx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 1000
- max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, 472
 - unpack_swiloc_SwiLocGetAutoStart_t, 899
- max_dist_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 899
- max_time
 - pack_swiloc_SwiLocSetAutoStart_t, 472
 - unpack_swiloc_SwiLocGetAutoStart_t, 899
- max_time_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 899
- MaxActive
 - dms_devMaxCfgListCaps, 59
 - dms_devMultiSimVoiceDataCaps, 61
 - dms_devSubsCfgList, 62
- MaxAllowedPktSz
 - unpack_qos_swilQosFlow_t, 854
- MaxChanRxRate
 - dunchannelRate, 77
- MaxChanTxRate
 - dunchannelRate, 77
- maxChannelRXRate
 - unpack_wds_GetConnectionRate_t, 975
- maxChannelTXRate
 - unpack_wds_GetConnectionRate_t, 975
- maxDIBitRate
 - LibPackQosClassID, 142
- maxDownlinkBitrate
 - LibPackUMTSQoS, 146
 - wds_UMTSMInQoS, 1102
- maxImages
 - FMSImageIDEntries, 81
- maxMitigationLevel
 - tmd_mitigationDevList, 576
- MaxRXChannelRate
 - dms_devCaps, 57
 - unpack_dms_GetDeviceCap_t, 619
- maxRxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, 621
- maxSDUSize
 - LibPackUMTSQoS, 146
 - wds_UMTSMInQoS, 1102
- maxStorageSize
 - sms_maxStorageSizeResp, 555
- 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, 619
- maxTxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, 621
- maxUIBitRate
 - LibPackQosClassID, 142
- maxUplinkBitrate
 - LibPackUMTSQoS, 146
 - wds_UMTSMInQoS, 1102
- mcc
 - nas_CSGID, 187
 - nas_MNRInfo, 246
 - nas_netSelectionPref, 248
 - nas_OperatorPLMNData, 260
 - nas_PlmnID, 268
 - pack_nas_SLQSGetPLMNName_t, 400
 - unpack_nas_GetHomeNetwork_t, 749
- mccM
 - nas_minBasedIMSI, 245
- mccT
 - nas_trueIMSI, 310
- mdmCallDurationActive
 - unpack_wds_SLQSGetDUNCallInfo_t, 1004
- meanThroughputClass
 - LibPackGPRSRequestedQoS, 105
 - wds_GPRSQoS, 1092
- meid
 - unpack_dms_GetSerialNumbers_t, 640
- meidSize
 - unpack_dms_GetDeviceSerialNumbers_t, 627
- meidString
 - unpack_dms_SLQSSwiGetSerialNoExt_t, 676
- memoryDumpAllowed
 - unpack_swidms_SLQSSwiDmsGetSecureInfo_t, 894
- message
 - unpack_sms_SLQSGetSMS_t, 866
- message_type
 - NASOTAMessageTlv, 330
- messageClass
 - sms_routeEntry, 558
- messageFailureCode
 - unpack_sms_SendSMS_t, 861
- messageFormat
 - pack_sms_SaveSMS_t, 448
 - pack_sms_SendSMS_t, 448
 - sms_sendAsynccsmsParams, 560

- unpack_sms_SLQSGetSMS_t, 866
- messageID
 - unpack_sms_SendSMS_t, 861
 - unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 877
- messageIndex
 - pack_sms_SLQSGetSMS_t, 451
 - pack_sms_SLQSMModifySMSStatus_t, 453
 - qmiSmsMessageList, 541
 - sMSMTMessageInfo, 566
- messageList
 - unpack_sms_SLQSGetSMSList_t, 868
- messageListSize
 - unpack_sms_SLQSGetSMSList_t, 868
- messageMode
 - sMSMessageModelInfo, 566
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 878
- MessageModelInfo
 - messageModeTlv, 167
- messageModeTlv, 167
 - MessageModelInfo, 167
 - TlvPresent, 167
- messageSize
 - pack_sms_SaveSMS_t, 448
 - pack_sms_SendSMS_t, 448
 - sms_sendAsynsmsParams, 560
 - unpack_sms_SLQSGetSMS_t, 866
- messageTag
 - pack_sms_SLQSMModifySMSStatus_t, 453
 - qmiSmsMessageList, 541
 - unpack_sms_SLQSGetSMS_t, 866
- messageType
 - sms_routeEntry, 558
- MicMute
 - pack_audio_SLQSSetAudioProfile_t, 347
 - pack_swiaudio_SLQSSetM2MAVMute_t, 462
 - unpack_audio_SLQSGetAudioProfile_t, 609
 - unpack_swiaudio_SLQSGetM2MAudioProfile_t, 879
 - unpack_swiaudio_SLQSGetM2MAVMute_t, 881
- min
 - PackSwiAVMSSettingsPeriodsInfo, 541
- MinPolicedPktSz
 - unpack_qos_swiQosFlow_t, 854
- minSessExp
 - ims_MinSessExpInfo, 90
- minSize
 - unpack_dms_GetVoiceNumber_t, 643
- minute
 - nas_timeInfo, 309
 - nas_UniversalTime, 315
- mipMode
 - unpack_wds_GetMobileIP_t, 984
- mipStatus
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- mipstatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- mitigationDevID
 - pack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 478
 - pack_tmd_SLQSTmdGetMitigationLvl_t, 479
 - pack_tmd_SLQSTmdRegNotMitigationLvl_t, 479
- mitigationDevIDLen
 - pack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 478
 - pack_tmd_SLQSTmdGetMitigationLvl_t, 479
 - pack_tmd_SLQSTmdRegNotMitigationLvl_t, 479
- mitigationDevId
 - tmd_mitigationDevList, 576
- mitigationDevIdLen
 - tmd_mitigationDevList, 576
- MitigationDevList
 - unpack_tmd_SLQSTmdGetMitigationDevList_t, 910
- MitigationDevListLen
 - unpack_tmd_SLQSTmdGetMitigationDevList_t, 910
- mnc
 - nas_CSGID, 187
 - nas_MNRInfo, 246
 - nas_netSelectionPref, 248
 - nas_OperatorPLMNData, 260
 - nas_PlmnID, 268
 - pack_nas_SLQSGetPLMNName_t, 400
 - unpack_nas_GetHomeNetwork_t, 749
- mncPcsDigits
 - nas_CSGID, 187
- mobileCountryCode
 - sMSEtwsPlmnInfo, 565
- mobileIP
 - pack_wds_SLQSSetWdsEventCallback_t, 530
- mobileNetworkCode
 - sMSEtwsPlmnInfo, 565
- mode
 - pack_dms_SetEventReport_t, 356
 - pack_dms_SetPower_t, 358
 - pack_loc_SetOperationMode_t, 382
 - pack_sms_SLQSGetSmsBroadcastConfig_t, 451
 - pack_sms_SLQSSetSmsBroadcastActivation_t, 454
 - pack_sms_SLQSSetSmsBroadcastConfig_t, 455
 - pack_wds_SetMobileIP_t, 518
 - uim_refreshevent, 597
 - voice_callInfo, 1052
- ModePref
 - nas_ModePrefTlv, 246
 - NASModePreferenceTlv, 328
- modelIdStr
 - _litfw_FirmwareFileInfo, 36
- modelString
 - pack_dms_SLQSSwiSetHostDevInfo_t, 362
 - unpack_dms_SLQSSwiGetHostDevInfo_t, 674
- modelid

- unpack_dms_GetModelID_t, 634
- modelid_str
 - unpack_dms_GetFirmwareInfo_t, 628
- modemMode
 - nas_CommInfo, 186
- modemindex
 - pack_fms_SetImagesPreference_t, 370
- month
 - nas_timeInfo, 309
 - nas_UniversalTime, 315
- msgCount
 - sms_messageWaitingInfoContent, 555
- msgDelFailureCause
 - unpack_sms_SLQSWmsAsyncRawSendCallBack-
_ind_t, 877
- msgDelFailureType
 - unpack_sms_SLQSWmsAsyncRawSendCallBack-
_ind_t, 877
- msgProtocol
 - sms_msgProtocolResp, 556
- msgType
 - sms_messageWaitingInfoContent, 555
- msgWaitInfo
 - sms_getMsgWaitingInfo, 552
 - unpack_sms_SLQSWmsMessageWaitingCall-
Back_ind_t, 878
- msgid
 - pack_qmi_t, 444
 - unpack_qmi_t, 834
- msgtype
 - common.h, 1118
- Mtu
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- MultDisc
 - nas_protocolSubtypeElement, 272
- multiplier
 - unpack_qos_pktErrRate_t, 837
- MuxID
 - qos_BindDataPortMuxID_t, 543
- NAI
 - unpack_wds_GetMobileIPProfile_t, 986
- NAS_NAM_NAME_LENGTH
 - nas.h, 1252
- NAS_PLMN_LENGTH
 - nas.h, 1252
- NASAcqOrderPrefTlv, 322
 - AcqOrderLen, 322
 - AcqOrderPref, 322
 - TlvPresent, 322
- NASBandPreferenceTlv, 322
 - band_pref, 322
 - TlvPresent, 322
- NASCiotAcqOrderPrefTlv, 323
 - CiotAcqOrderLen, 323
 - CiotAcqOrderPref, 323
 - TlvPresent, 323
- NASCiotLteOpModePrefTlv, 323
 - CiotLteOpModePref, 323
- TlvPresent, 323
- NASEmergencyModeTlv, 323
 - EmerMode, 324
 - TlvPresent, 324
- NASGWAcqOrderPrefTlv, 325
 - GWAcqOrderPref, 325
 - TlvPresent, 325
- NASLTEBandPreferenceTlv, 325
 - LTEBandPref, 326
 - TlvPresent, 326
- NASLteM1BandPrefTlv, 326
 - LteM1BandPref, 326
 - TlvPresent, 326
- NASLteNB1BandPrefTlv, 327
 - LteNB1BandPref, 327
 - TlvPresent, 327
- NASLteNasReleaseInfoTlv, 326
 - nas_major, 327
 - nas_minor, 327
 - nas_release, 327
 - TlvPresent, 327
- NASModePreferenceTlv, 327
 - ModePref, 328
 - TlvPresent, 328
- NASNetSelPreferenceTlv, 328
 - NetSelPref, 328
 - TlvPresent, 328
- NASNr5gBandPrefTlv, 328
 - Nr5gBandPrefbits_1_64, 329
 - Nr5gBandPrefbits_129_192, 329
 - Nr5gBandPrefbits_193_256, 329
 - Nr5gBandPrefbits_65_128, 329
 - TlvPresent, 329
- NASOTAMessageTlv, 329
 - data_buf, 329
 - data_len, 330
 - message_type, 330
 - TlvPresent, 330
- NASPRLPreferenceTlv, 335
 - PRLPref, 335
 - TlvPresent, 335
- NASPhyCaAggPcellInfo, 330
 - dl_bw_value, 330
 - freq, 330
 - iLTEbandValue, 330
 - pci, 330
 - TlvPresent, 330
- NASPhyCaAggScellArray, 331
 - band, 332
 - cphy_ca_dl_bandwidth, 332
 - cphy_scell_info_list_len, 332
 - freq, 332
 - pci, 332
 - scell_idx, 332
 - scell_state, 332
 - TlvPresent, 332
- NASPhyCaAggScellDIBw, 332
 - dl_bw_value, 332

- TlvPresent, [332](#)
- NASPhyCaAggScellIndType, [333](#)
 - freq, [333](#)
 - pci, [334](#)
 - scell_state, [334](#)
 - TlvPresent, [334](#)
- NASPhyCaAggScellIndex, [332](#)
 - scell_idx, [333](#)
 - TlvPresent, [333](#)
- NASPhyCaAggScellInfo, [334](#)
 - dl_bw_value, [334](#)
 - freq, [334](#)
 - iLTEbandValue, [335](#)
 - pci, [335](#)
 - scell_state, [335](#)
 - TlvPresent, [335](#)
- NASQmiCbkNasSwiOTAMessageInd, [335](#)
 - nasRelInfoTlv, [336](#)
 - otaMsgTlv, [336](#)
 - timeTlv, [336](#)
- NASQmiCbkNasSystemSelPrefInd, [336](#)
 - AOPTlv, [338](#)
 - BPTlv, [338](#)
 - CiotAOPTlv, [338](#)
 - CiotOpMPTlv, [338](#)
 - EMTlv, [338](#)
 - GWAOPTlv, [338](#)
 - LBPTlv, [338](#)
 - M1BandPTlv, [338](#)
 - MPTlv, [338](#)
 - NB1BandPTlv, [338](#)
 - NR5gBandPTlv, [338](#)
 - NSPTlv, [338](#)
 - PRLPTlv, [338](#)
 - RPTlv, [338](#)
 - RatDMTlv, [338](#)
 - SDPTlv, [338](#)
- NASRatDisabledMaskTlv, [338](#)
 - RatDisabledMask, [339](#)
 - TlvPresent, [339](#)
- NASRoamPreferenceTlv, [339](#)
 - RoamPref, [339](#)
 - TlvPresent, [339](#)
- NASServDomainPrefTlv, [339](#)
 - SrvDomainPref, [339](#)
 - TlvPresent, [339](#)
- NASServingSystemInfo, [339](#)
 - csAttachState, [341](#)
 - hdrPersonality, [341](#)
 - psAttachState, [341](#)
 - radioInterfaceList, [341](#)
 - radioInterfaceNo, [341](#)
 - registrationState, [341](#)
 - selectedNetwork, [341](#)
- NASTimeInfoTlv, [341](#)
 - time, [341](#)
 - TlvPresent, [341](#)
- NB1BandPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- NMEADData
 - unpack_loc_EventNMEA_Ind_t, [720](#)
- NR5gBandPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- NSPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- nUsbNetNum
 - pack_swidms_SLQSSwiDmsSetUsbNetNum_t, [470](#)
- NWQoSStatus
 - unpack_qos_SLQSQosGetNetworkStatus_t, [840](#)
- NWRegStat
 - unpack_sms_SLQSNWRegInfoCallback_ind_t, [870](#)
- naiSize
 - unpack_wds_GetMobileIPProfile_t, [986](#)
- namID
 - pack_nas_SLQSNasGet3GPP2Subscription_t, [406](#)
 - voice_airTimer, [1033](#)
 - voice_prefVoiceSO, [1073](#)
 - voice_roamTimer, [1077](#)
- namName
 - nas_namName, [247](#)
- namNameLen
 - nas_namName, [247](#)
- Name
 - unpack_nas_GetServingNetwork_t, [753](#)
- name
 - unpack_nas_GetHomeNetwork_t, [749](#)
 - unpack_wds_GetDefaultProfile_t, [979](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- nameLen
 - voice_remotePartyName, [1075](#)
- namePI
 - voice_remotePartyName, [1075](#)
- nameSize
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
 - unpack_nas_GetServingNetwork_t, [754](#)
- nameString
 - pack_dms_SLQSSwiSetOSInfo_t, [362](#)
 - unpack_dms_SLQSSwiGetOSInfo_t, [675](#)
- namelength
 - unpack_omaDmFotaTlv_t, [819](#)
- namesize
 - unpack_wds_GetDefaultProfile_t, [979](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- nas.h
 - eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_1, [1253](#)
 - eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_10, [1254](#)
 - eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_11, [1254](#)
 - eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND_12, [1254](#)

- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_125, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_126, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_127, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_13, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_14, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_17, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_18, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_19, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_2, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_20, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_21, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_23, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_24, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_25, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_250, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_26, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_27, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_28, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_29, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_3, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_30, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_31, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_32, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_33, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_34, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_35, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_36, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_37, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_38, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_39, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_4, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_40, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_41, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_42, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_43, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_46, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_47, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_48, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_5, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_6, [1253](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_66, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_7, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_71, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_8, [1254](#)
- eLITEQMI_LTE_BAND_CLASS_E_UTRA_OPER-
ATING_BAND_9, [1254](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100,
[1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15,
[1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25,
[1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50,
[1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6,
[1255](#)
- eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75,
[1255](#)
- eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_C-
ONFIGURED_ACTIVATED, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_C-
ONFIGURED_DEACTIVATED, [1255](#)
- eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_D-
ECONFIGURED, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_100,
[1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, [1255](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, [1255](#)

- eNAS_LTE_CPHY_SELL_STATE_CONFIGURED_ACTIVATED_LITE, [1255](#)
- eNAS_LTE_CPHY_SELL_STATE_CONFIGURED_DEACTIVATED_LITE, [1255](#)
- eNAS_LTE_CPHY_SELL_STATE_DECONFIGURED_LITE, [1255](#)
- LITE_SYS_SRV_DOMAIN_CAMPED, [1253](#)
- LITE_SYS_SRV_DOMAIN_CS_ONLY, [1253](#)
- LITE_SYS_SRV_DOMAIN_CS_PS, [1253](#)
- LITE_SYS_SRV_DOMAIN_NO_SRV, [1253](#)
- LITE_SYS_SRV_DOMAIN_PS_ONLY, [1253](#)
- nas.h, [1240](#)
- NAS_PLMN_LENGTH, [1252](#)
- nas_LTEBandPrefExtTlv, [1252](#)
- nas_NR5gBandPrefTlv, [1252](#)
- pack_nas_GetACCOLC, [1255](#)
- pack_nas_GetANAAAAAuthenticationStatus, [1256](#)
- pack_nas_GetCDMANetworkParameters, [1256](#)
- pack_nas_GetHomeNetwork, [1256](#)
- pack_nas_GetHomeNetwork3GPP2, [1257](#)
- pack_nas_GetNetworkPreference, [1257](#)
- pack_nas_GetRFInfo, [1257](#)
- pack_nas_GetServingNetwork, [1257](#)
- pack_nas_GetServingNetworkCapabilities, [1258](#)
- pack_nas_GetSignalStrengths, [1258](#)
- pack_nas_InitiateDomainAttach, [1258](#)
- pack_nas_PerformNetworkScan, [1259](#)
- pack_nas_PerformNetworkScanPCI, [1259](#)
- pack_nas_SLQSConfigSigInfo, [1261](#)
- pack_nas_SLQSGetErrorRate, [1262](#)
- pack_nas_SLQSGetNetworkTime, [1262](#)
- pack_nas_SLQSGetOperatorNameData, [1263](#)
- pack_nas_SLQSGetPLMNName, [1263](#)
- pack_nas_SLQSGetServingSystem, [1263](#)
- pack_nas_SLQSGetSignalStrength, [1264](#)
- pack_nas_SLQSGetSysInfo, [1264](#)
- pack_nas_SLQSGetSysSelectionPref, [1264](#)
- pack_nas_SLQSGetSysSelectionPrefExt, [1265](#)
- pack_nas_SLQSInitiateNetworkRegistration, [1265](#)
- pack_nas_SLQSNASGetForbiddenNetworks, [1267](#)
- pack_nas_SLQSNASGetDRXParams, [1267](#)
- pack_nas_SLQSNASGetDRXParamsExt, [1267](#)
- pack_nas_SLQSNASSeteDRXParams, [1269](#)
- pack_nas_SLQSNASSwiGetChannelLock, [1270](#)
- pack_nas_SLQSNASSwiSetChannelLock, [1271](#)
- pack_nas_SLQSNasConfigSigInfo2, [1265](#)
- pack_nas_SLQSNasGet3GPP2Subscription, [1266](#)
- pack_nas_SLQSNasGetCellLocationInfo, [1266](#)
- pack_nas_SLQSNasGetHDRColorCode, [1268](#)
- pack_nas_SLQSNasGetRFInfo, [1268](#)
- pack_nas_SLQSNasGetSigInfo, [1268](#)
- pack_nas_SLQSNasGetTxRxInfo, [1269](#)
- pack_nas_SLQSNasIndicationRegisterExt, [1269](#)
- pack_nas_SLQSNasSwiIndicationRegister, [1270](#)
- pack_nas_SLQSNasSwiModemStatus, [1271](#)
- pack_nas_SLQSSetBandPreference, [1271](#)
- pack_nas_SLQSSetSignalStrengthsCallback, [1272](#)
- pack_nas_SLQSSetSysSelectionPref, [1272](#)
- pack_nas_SLQSSetSysSelectionPrefExt, [1272](#)
- pack_nas_SLQSSwiGetHDRPersonality, [1273](#)
- pack_nas_SLQSSwiGetHDRProtSubtype, [1273](#)
- pack_nas_SLQSSwiGetHRPDStats, [1273](#)
- pack_nas_SLQSSwiGetLteCQI, [1274](#)
- pack_nas_SLQSSwiGetLteSccRxInfo, [1274](#)
- pack_nas_SLQSSwiNetworkDebug, [1274](#)
- pack_nas_SLQSSwiPSDetach, [1275](#)
- pack_nas_SetACCOLC, [1260](#)
- pack_nas_SetCDMANetworkParameters, [1260](#)
- pack_nas_SetLURejectCallback, [1260](#)
- pack_nas_SetNetworkPreference, [1261](#)
- pack_nas_SetRFInfoCallback, [1261](#)
- pack_nas_SlqsGetLTECphyCAInfo, [1262](#)
- unpack_nas_GetACCOLC, [1275](#)
- unpack_nas_GetANAAAAAuthenticationStatus, [1275](#)
- unpack_nas_GetCDMANetworkParameters, [1276](#)
- unpack_nas_GetHomeNetwork, [1276](#)
- unpack_nas_GetHomeNetwork3GPP2, [1277](#)
- unpack_nas_GetNetworkPreference, [1277](#)
- unpack_nas_GetRFInfo, [1277](#)
- unpack_nas_GetServingNetwork, [1278](#)
- unpack_nas_GetServingNetworkCapabilities, [1278](#)
- unpack_nas_GetSignalStrengths, [1278](#)
- unpack_nas_InitiateDomainAttach, [1279](#)
- unpack_nas_InitiateDomainAttach_t, [1252](#)
- unpack_nas_PerformNetworkScan, [1279](#)
- unpack_nas_SLQSConfigSigInfo, [1283](#)
- unpack_nas_SLQSConfigSigInfo_t, [1253](#)
- unpack_nas_SLQSGetErrorRate, [1283](#)
- unpack_nas_SLQSGetNetworkTime, [1284](#)
- unpack_nas_SLQSGetOperatorNameData, [1284](#)
- unpack_nas_SLQSGetPLMNName, [1285](#)
- unpack_nas_SLQSGetServingSystem, [1285](#)
- unpack_nas_SLQSGetSignalStrength, [1285](#)
- unpack_nas_SLQSGetSysInfo, [1286](#)
- unpack_nas_SLQSGetSysSelectionPref, [1286](#)
- unpack_nas_SLQSGetSysSelectionPrefExt, [1287](#)
- unpack_nas_SLQSInitiateNetworkRegistration, [1287](#)
- unpack_nas_SLQSInitiateNetworkRegistration_t, [1253](#)
- unpack_nas_SLQSNASGetForbiddenNetworks, [1290](#)
- unpack_nas_SLQSNASGetDRXParams, [1289](#)
- unpack_nas_SLQSNASGetDRXParamsExt, [1289](#)
- unpack_nas_SLQSNASSeteDRXParams, [1293](#)
- unpack_nas_SLQSNASSeteDRXParams_t, [1253](#)
- unpack_nas_SLQSNASSwiGetChannelLock, [1293](#)
- unpack_nas_SLQSNASSwiSetChannelLock, [1295](#)
- unpack_nas_SLQSNASSwiSetChannelLock_t, [1253](#)
- unpack_nas_SLQSNasConfigSigInfo2, [1287](#)

- unpack_nas_SLQSNasConfigSigInfo2_t, 1253
- unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind, 1288
- unpack_nas_SLQSNasGet3GPP2Subscription, 1288
- unpack_nas_SLQSNasGetCellLocationInfo, 1288
- unpack_nas_SLQSNasGetHDRColorCode, 1290
- unpack_nas_SLQSNasGetRFInfo, 1290
- unpack_nas_SLQSNasGetSigInfo, 1291
- unpack_nas_SLQSNasGetTxRxInfo, 1291
- unpack_nas_SLQSNasIndicationRegisterExt, 1291
- unpack_nas_SLQSNasIndicationRegisterExt_t, 1253
- unpack_nas_SLQSNasNetworkRejectCallback_Ind, 1292
- unpack_nas_SLQSNasNetworkTimeCallBack_ind, 1292
- unpack_nas_SLQSNasSigInfoCallback_ind, 1293
- unpack_nas_SLQSNasSwiIndicationRegister, 1294
- unpack_nas_SLQSNasSwiIndicationRegister_t, 1253
- unpack_nas_SLQSNasSwiModemStatus, 1294
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind, 1294
- unpack_nas_SLQSNasSysInfoCallback_ind, 1295
- unpack_nas_SLQSNasTimerCallback_ind, 1295
- unpack_nas_SLQSSetBandPreference, 1296
- unpack_nas_SLQSSetBandPreference_t, 1253
- unpack_nas_SLQSSetSignalStrengthsCallback, 1296
- unpack_nas_SLQSSetSignalStrengthsCallback_t, 1253
- unpack_nas_SLQSSetSysSelectionPref, 1296
- unpack_nas_SLQSSetSysSelectionPref_t, 1253
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind, 1297
- unpack_nas_SLQSSetSysSelectionPrefExt, 1297
- unpack_nas_SLQSSetSysSelectionPrefExt_t, 1253
- unpack_nas_SLQSSwiGetHDRPersonality, 1297
- unpack_nas_SLQSSwiGetHDRProtSubtype, 1298
- unpack_nas_SLQSSwiGetHRPDStats, 1298
- unpack_nas_SLQSSwiGetLteCQI, 1298
- unpack_nas_SLQSSwiGetLteScsRxInfo, 1299
- unpack_nas_SLQSSwiHDRPersonalityCallback_Ind, 1299
- unpack_nas_SLQSSwiNetworkDebug, 1299
- unpack_nas_SLQSSwiPSDetach, 1300
- unpack_nas_SLQSSwiPSDetach_t, 1253
- unpack_nas_SLQSSwiRandIndicatorCallback_Ind, 1300
- unpack_nas_SetACCOLC, 1279
- unpack_nas_SetACCOLC_t, 1253
- unpack_nas_SetCDMANetworkParameters, 1280
- unpack_nas_SetCDMANetworkParameters_t, 1253
- unpack_nas_SetDataCapabilitiesCallback_ind, 1280
- unpack_nas_SetEventReportInd, 1280
- unpack_nas_SetLURejectCallback, 1281
- unpack_nas_SetLURejectCallback_t, 1253
- unpack_nas_SetNasLTECphyCaIndCallback_ind, 1281
- unpack_nas_SetNetworkPreference, 1282
- unpack_nas_SetRFInfoCallback, 1282
- unpack_nas_SetRFInfoCallback_t, 1253
- unpack_nas_SetRoamingIndicatorCallback_ind, 1282
- unpack_nas_SetServingSystemCallback_ind, 1283
- unpack_nas_SlqsGetLTECphyCAInfo, 1284
- unpack_valid_nas_GetCDMANetworkParameters, 1300
- unpack_valid_nas_SLQSGetServingSystem, 1301
- unpack_valid_nas_SLQSGetSignalStrength, 1302
- unpack_valid_nas_SLQSNasGetSigInfo, 1302
- nas_AcqOrderPrefTlv, 168
 - acqOrdeLen, 168
 - pAcqOrder, 168
 - TlvPresent, 169
- nas_ActPilotPNElement, 169
 - ActSetPilotPN, 169
 - ActSetPilotPNStrength, 169
- nas_AddCDMASysInfo, 169
 - geoSysIdx, 170
 - regPrd, 170
- nas_AddSysInfo, 170
 - cellBroadcastCap, 170
 - geoSysIdx, 170
- nas_BandPrefInfoTlv, 170
 - bits_129_192, 171
 - bits_193_256, 171
 - bits_1_64, 171
 - bits_65_128, 171
 - TlvPresent, 171
- nas_BandPrefTlv, 171
 - BandPref, 173
 - TlvPresent, 173
- nas_CDMAChannel, 174
 - priChA, 175
 - priChB, 175
 - secChA, 175
 - secChB, 175
- nas_CDMAECIOThresh, 175
 - CDMAECIOThreshListLen, 175
 - pCDMAECIOThreshList, 175
- nas_CDMAInfo, 176
 - baseId, 176
 - baseLat, 176
 - baseLong, 176
 - nid, 176
 - refpn, 177
 - sid, 177
- nas_CDMARSSIThresh, 177

- CDMARSSIThreshListLen, 177
- pCDMARSSIThreshList, 177
- nas_CDMASysInfo, 177
 - baseId, 180
 - baseLat, 180
 - baseLong, 180
 - bsInfoValid, 180
 - bsPRev, 180
 - bsPRevValid, 180
 - ccsSupported, 180
 - ccsSupportedValid, 180
 - cdmaSysIdValid, 180
 - isSysPrIMatch, 180
 - isSysPrIMatchValid, 180
 - MCC, 180
 - MNC, 180
 - networkID, 180
 - networkIdValid, 180
 - pRevInUse, 181
 - pRevInUseValid, 181
 - packetZone, 181
 - packetZoneValid, 181
 - sysInfoCDMA, 181
 - systemID, 181
- nas_CDMASysInfoExt, 181
 - imsi_11_12, 181
 - MCC, 181
- nas_CSGID, 186
 - id, 187
 - mcc, 187
 - mnc, 187
 - mncPcsDigits, 187
 - rat, 187
- nas_CallBarringSysInfo, 173
 - csBarStatus, 173
 - psBarStatus, 173
- nas_CiotAcqOrderPrefTlv, 183
 - ciotAcqOrderLen, 184
 - pCiotAcqOrder, 184
 - TlvPresent, 184
- nas_CiotLteOpModePrefTlv, 184
 - ciotLteOpModePref, 184
 - TlvPresent, 184
- nas_CommInfo, 184
 - imsRegState, 186
 - modemMode, 186
 - psState, 186
 - systemMode, 186
 - temperature, 186
- nas_CsgId, 187
 - csgId, 187
 - TlvPresent, 187
- nas_DRCParams, 193
 - DRCover, 194
 - DRCValue, 194
- nas_DataStatusDetail, 189
 - IPAddress, 190
 - LastErrCode, 190
- nas_DeviceConfigDetail, 192
 - Chipset, 193
 - HWVersion, 193
 - QLIC, 193
 - Technology, 193
- nas_ECIOThresh, 195
 - ECIOThresListLen, 195
 - pECIOThresList, 195
- nas_EdrxCiotLteMode, 195
 - lteOpMode, 196
 - TlvPresent, 196
- nas_EdrxCycleLength, 196
 - cycleLength, 196
 - TlvPresent, 196
- nas_EdrxEnableType, 196
 - edrxEnabled, 197
 - TlvPresent, 197
- nas_EdrxPagingTimeWindow, 197
 - edrxPtw, 197
 - TlvPresent, 197
- nas_EdrxRatType, 197
 - edrxRatType, 198
 - TlvPresent, 198
- nas_EmerModeTlv, 198
 - EmerMode, 199
 - TlvPresent, 199
- nas_ForbiddenNetworks3GPP, 200
 - forbiddenNwInstLen, 201
 - MCC, 201
 - MNC, 201
 - TlvPresent, 201
- nas_GERANInfo, 201
 - arfcn, 202
 - bsic, 202
 - cellID, 202
 - insNmrCellInfo, 202
 - lac, 202
 - nmrInst, 202
 - plmn, 202
 - rxLev, 202
 - timingAdvance, 202
- nas_GSMRSSIThresh, 204
 - GSMRSSIThreshListLen, 205
 - pGSMRSSIThreshList, 205
- nas_GSMSrvStatusInfo, 205
 - isPrefDataPath, 206
 - srvStatus, 206
 - trueSrvStatus, 206
- nas_GSMSysInfo, 206
 - cellId, 208
 - cellIdValid, 208
 - dtmSupp, 208
 - dtmSuppValid, 208
 - egprsSupp, 208
 - egprsSuppValid, 208
 - lac, 208
 - lacValid, 208
 - MCC, 208

- MNC, [208](#)
- networkIdValid, [208](#)
- regRejectInfoValid, [208](#)
- rejCause, [208](#)
- rejectSrvDomain, [209](#)
- sysInfoGSM, [209](#)
- nas_GWAcqOrderPrefTlv, [209](#)
 - GWAcqOrderPref, [209](#)
 - TlvPresent, [209](#)
- nas_HDRECIOTthresh, [209](#)
 - HDRECIOTthreshListLen, [210](#)
 - pHDRECIOTthreshList, [210](#)
- nas_HDRIOTthresh, [210](#)
 - HDRIOTthreshListLen, [210](#)
 - pHDRIOTthreshList, [210](#)
- nas_HDRPersonality_Ind_Data, [210](#)
 - pCurrentPersonality, [211](#)
 - pPersonalityListLength, [211](#)
 - pProtocolSubtypeElement, [211](#)
- nas_HDRRSSITthresh, [211](#)
 - HDRRSSITthreshListLen, [211](#)
 - pHRRSSITthreshList, [211](#)
- nas_HDRSINRThresh, [212](#)
 - HDRSINRThresListLen, [212](#)
 - pHRSINRThresList, [212](#)
- nas_HDRSINRThreshold, [212](#)
 - HDRSINRThreshListLen, [213](#)
 - pHRSINRThreshList, [213](#)
- nas_HDRSysInfo, [213](#)
 - hdrActiveProt, [215](#)
 - hdrActiveProtValid, [215](#)
 - hdrPersonality, [215](#)
 - hdrPersonalityValid, [215](#)
 - is856SysId, [215](#)
 - is856SysIdValid, [215](#)
 - isSysPrIMatch, [215](#)
 - isSysPrIMatchValid, [215](#)
 - sysInfoHDR, [215](#)
- nas_IOTthresh, [217](#)
 - IOTthreshListLen, [217](#)
 - pIOTthreshList, [217](#)
- nas_LTEBandPrefExtTlv
 - nas.h, [1252](#)
- nas_LTEBandPrefTlv, [218](#)
 - LTEBandPref, [219](#)
 - TlvPresent, [220](#)
- nas_LTEInfo, [222](#)
 - band, [224](#)
 - bandwidth, [224](#)
 - emmConnState, [224](#)
 - emmState, [224](#)
 - emmSubState, [224](#)
 - RXChan, [224](#)
 - TXChan, [224](#)
- nas_LTEInfoIntrafreq, [225](#)
 - freqsLen, [225](#)
 - InfoInterfreq, [225](#)
 - ueInIdle, [225](#)
- nas_LTEInfoIntrafreq, [225](#)
 - CellParams, [227](#)
 - cellReselPriority, [227](#)
 - cellsLen, [227](#)
 - earfcn, [227](#)
 - globalCellId, [227](#)
 - plmn, [227](#)
 - sIntraSearch, [227](#)
 - sNonIntraSearch, [227](#)
 - servingCellId, [227](#)
 - tac, [227](#)
 - threshServingLow, [227](#)
 - ueInIdle, [227](#)
- nas_LTEInfoNeighboringGSM, [227](#)
 - freqsLen, [228](#)
 - LteGsmCellInfo, [228](#)
 - ueInIdle, [228](#)
- nas_LTEInfoNeighboringWCDMA, [228](#)
 - freqsLen, [229](#)
 - ueInIdle, [229](#)
- nas_LTEOperationMode, [233](#)
 - pLTEOperationMode, [233](#)
 - TlvPresent, [233](#)
- nas_LTERSRPThresh, [236](#)
 - LTERSRPThreshListLen, [236](#)
 - pLTERSRPThreshList, [236](#)
- nas_LTERSRQThresh, [236](#)
 - LTERSRQThreshListLen, [237](#)
 - pLTERSRQThreshList, [237](#)
- nas_LTERSSITthresh, [237](#)
 - LTERSSITthreshListLen, [237](#)
 - pLTERSSITthreshList, [237](#)
- nas_LTESNRThresh, [240](#)
 - LTESNRThresListLen, [240](#)
 - pLTESNRThresList, [240](#)
- nas_LTESNRThreshold, [240](#)
 - LTESNRThreshListLen, [241](#)
 - pLTESNRThreshList, [241](#)
- nas_LTESigRptCfg, [238](#)
 - avgPeriod, [238](#)
 - rptRate, [238](#)
- nas_LTESigRptConfig, [238](#)
 - avgPeriod, [239](#)
 - rptRate, [239](#)
- nas_LTESysInfo, [241](#)
 - cellId, [243](#)
 - cellIdValid, [243](#)
 - lac, [243](#)
 - lacValid, [243](#)
 - MCC, [243](#)
 - MNC, [243](#)
 - networkIdValid, [243](#)
 - regRejectInfoValid, [243](#)
 - rejCause, [243](#)
 - rejectSrvDomain, [243](#)
 - sysInfoLTE, [243](#)
 - tac, [243](#)
 - tacValid, [243](#)

- nas_LteCiotOpModeTlv, 220
 - campedCiotLteOpMode, 220
 - TlvPresent, 220
- nas_LteM1BandPrefTlv, 229
 - lteM1BandPref, 230
 - TlvPresent, 231
- nas_LteNb1BandPrefTlv, 231
 - lteNb1BandPref, 232
 - TlvPresent, 233
- nas_LteOpMode, 233
 - lteOpMode, 234
 - TlvPresent, 234
- nas_MNRInfo, 245
 - mcc, 246
 - mnc, 246
 - rat, 246
- nas_ModePrefTlv, 246
 - ModePref, 246
 - TlvPresent, 246
- nas_NR5GCellStatusTlv, 254
 - nr5gCellStatus, 254
 - TlvPresent, 254
- nas_NR5GSerStatTlv, 255
 - isPrefDataPath, 255
 - srvStatus, 255
 - TlvPresent, 255
 - trueSrvStatus, 255
- nas_NR5GSystemInfoTlv, 256
 - cellId, 258
 - cellIdValid, 258
 - lac, 258
 - lacValid, 258
 - MCC, 258
 - MNC, 258
 - nwIdValid, 258
 - regRejectInfoValid, 259
 - rejCause, 259
 - rejectSrvDomain, 259
 - roamStatus, 259
 - roamStatusValid, 259
 - srvCapValid, 259
 - srvDomain, 259
 - srvDomainValid, 259
 - srvcapability, 259
 - sysForbidden, 259
 - sysForbiddenValid, 259
 - tac, 259
 - tacValid, 259
 - TlvPresent, 259
- nas_NR5gBandPrefTlv
 - nas.h, 1252
- nas_NetSelPrefTlv, 248
 - NetSelPref, 248
 - TlvPresent, 248
- nas_NetworkStat1x, 248
 - ActSetCnt, 250
 - NeighborSetCnt, 250
 - pActPilotPNElements, 250
 - pNeighborSetPilotPN, 250
 - RX_EC_IO, 250
 - RX_PWR, 250
 - SO, 250
 - State, 250
 - TX_PWR, 250
- nas_NetworkStatEVDO, 250
 - MACIndex, 252
 - PER, 252
 - pSectorID, 252
 - PilotEnergy, 252
 - RX_PWR, 252
 - SNR, 252
 - SectorIDLen, 252
 - State, 252
- nas_OperatorPLMNData, 259
 - lac1, 260
 - lac2, 260
 - mcc, 260
 - mnc, 260
 - PLMNRecID, 260
- nas_PLMNNetworkName, 268
 - numInstance, 268
 - PLMNNetName, 268
- nas_PLMNNetworkNameData, 268
 - codingScheme, 270
 - countryInitials, 270
 - longName, 270
 - longNameLen, 270
 - longNameSpareBits, 270
 - shortName, 270
 - shortNameLen, 270
 - shortNameSpareBits, 270
- nas_PRLPrefTlv, 270
 - PRLPref, 271
 - TlvPresent, 271
- nas_PhyCaAggPcellInfo, 261
 - dl_bw_value, 262
 - freq, 262
 - iLTEbandValue, 262
 - pci, 262
 - TlvPresent, 262
- nas_PhyCaAggScellDIBw, 262
 - dl_bw_value, 262
 - TlvPresent, 262
- nas_PhyCaAggScellIndType, 263
 - freq, 263
 - pci, 263
 - scell_state, 263
 - TlvPresent, 263
- nas_PhyCaAggScellIndex, 262
 - scell_idx, 263
 - TlvPresent, 263
- nas_PhyCaAggScellInfo, 264
 - dl_bw_value, 266
 - freq, 266
 - iLTEbandValue, 266
 - pci, 266

- scell_state, 266
 - TlvPresent, 266
- nas_PilotSetData, 266
 - NumPilots, 266
 - pPilotSetInfo, 266
- nas_PilotSetParams, 267
 - PilotPN, 267
 - PilotStrength, 267
 - PilotType, 267
- nas_PlmnID, 267
 - mcc, 268
 - mnc, 268
 - pcsDigit, 268
 - TlvPresent, 268
- nas_QmiNas3GppNetworkInfo, 273
 - Description, 274
 - Forbidden, 274
 - InUse, 274
 - MCC, 274
 - MNC, 274
 - Preferred, 274
 - Roaming, 274
- nas_QmiNas3GppNetworkRAT, 274
 - MCC, 275
 - MNC, 275
 - RAT, 275
- nas_QmisNasPcsDigit, 275
 - includes_pcs_digit, 276
 - MCC, 276
 - MNC, 276
- nas_QmisNasSlqsNasPCICellInfo, 276
 - cellID, 276
 - freq, 276
 - GlobalCellID, 276
 - nasQmisNasPcsDigit, 276
 - PlmnLen, 276
- nas_QmisNasSlqsNasPCIInfo, 277
 - nasQmisNasSlqsNasPCICellInfo, 277
 - PCICellInfoLen, 277
 - rsrp, 277
 - rsrpRx0, 277
 - rsrpRx1, 277
 - rsrq, 278
 - rsrqRx0, 278
 - rsrqRx1, 278
- nas_RFBandInfoElements, 279
 - activeBandClass, 280
 - activeChannel, 280
 - radioInterface, 280
- nas_RFInfoTlv, 285
 - activeBandClass, 286
 - activeChannel, 286
 - radioInterface, 286
 - radioInterfaceSize, 286
 - TlvPresent, 286
- nas_RSRPThresh, 287
 - pRSRPThresList, 288
 - RSRPThresListLen, 288
- nas_RSRQThresh, 288
 - pRSRQThresList, 289
 - RSRQThresListLen, 289
- nas_RSSIThresh, 289
 - pRSSIThresList, 290
 - RSSIThresListLen, 290
- nas_RankIndicatorTlv, 278
 - count0, 278
 - count1, 278
 - TlvPresent, 278
- nas_RatDisabledMaskTlv, 278
 - ratDisabledMask, 279
 - TlvPresent, 279
- nas_RejectReasonTlv, 279
 - rejectCause, 279
 - serviceDomain, 279
 - TlvPresent, 279
- nas_RfBandInfoExtFormat, 280
 - pInstancesSize, 281
 - pRfBandInfoExtFormatParam, 281
 - TlvPresent, 281
- nas_RfBandInfoExtFormatElements, 281
 - activeBand, 281
 - activeChannel, 281
 - radioInterface, 281
- nas_RfBandInfoList, 282
 - pInstanceSize, 282
 - pRFBandInfoParam, 282
- nas_RfBandwidthInfo, 282
 - pInstancesSize, 283
 - pRfBandwidthInfoParam, 283
 - TlvPresent, 283
- nas_RfBandwidthInfoElements, 283
 - bandwidth, 283
 - radioInterface, 283
- nas_RfDedicatedBandInfo, 284
 - pInstancesSize, 284
 - pRfDedicatedBandInfoParam, 284
 - TlvPresent, 284
- nas_RfDedicatedBandInfoElements, 284
 - dedicatedBand, 285
 - radioInterface, 285
- nas_RoamPrefTlv, 287
 - RoamPref, 287
 - TlvPresent, 287
- nas_RxSigInfo, 291
 - isRadioTuned, 292
 - rsrp, 292
 - rxChainIndex, 292
 - rxPower, 292
- nas_SLQSSignalStrengthsIndReq, 297
 - ecioDelta, 298
 - ecioThresholdList, 298
 - ecioThresholdListLen, 298
 - ioDelta, 299
 - lteRsrpDelta, 299
 - lteSnrDelta, 299
 - rsrqDelta, 299

- rxSignalStrengthDelta, 299
- sinrDelta, 299
- sinrThresholdList, 299
- sinrThresholdListLen, 299
- nas_SLQSSignalStrengthsInformation, 299
 - ecioInfo, 300
 - errorRateInfo, 300
 - io, 300
 - lteRsrpinfo, 300
 - lteSnrinfo, 300
 - rsrqInfo, 300
 - rxSignalStrengthInfo, 300
 - sinr, 300
- nas_SLQSSignalStrengthsTlv, 300
 - sSLQSSignalStrengthsInfo, 301
 - TlvPresent, 301
- nas_SccRxInfo, 293
 - numInstances, 294
 - rsrq, 294
 - sigInfo, 294
 - snr, 294
 - TlvPresent, 294
- nas_SignalStrengthTlv, 297
 - radiolInterface, 297
 - signalStrength, 297
 - TlvPresent, 297
- nas_SrvDomainPrefTlv, 301
 - SrvDomainPref, 301
 - TlvPresent, 301
- nas_SrvStatusInfo, 301
 - isPrefDataPath, 302
 - srvStatus, 302
- nas_TDSCDMAECIOThresh, 305
- nas_TDSCDMARSCPTThresh, 305
- nas_TDSCDMARSSIThresh, 306
- nas_TDSCDMASINRCONFThresh, 306
- nas_TDSCDMASINRThresh, 307
- nas_UMTSInfo, 310
 - cellID, 312
 - ecio, 312
 - geranInst, 312
 - GeranInstInfo, 312
 - lac, 312
 - plmn, 312
 - psc, 312
 - rscp, 312
 - UMTSInstInfo, 312
 - uarfcn, 312
 - umtsInst, 312
- nas_UMTSinstInfo, 312
 - umtsEcio, 313
 - umtsPsc, 313
 - umtsRscp, 313
 - umtsUarfcn, 313
- nas_UniversalTime, 314
 - day, 315
 - dayOfWeek, 315
 - hour, 315
 - minute, 315
 - month, 315
 - second, 315
 - year, 315
- nas_WCDMAECIOThresh, 316
- nas_WCDMAInfoLTENeighborCell, 316
 - umtsLTENbrCellLen, 317
 - wcdmaRRCTest, 317
- nas_WCDMARSSIThresh, 317
- nas_WCDMASysInfo, 318
 - cellId, 320
 - cellIdValid, 321
 - hsCallStatus, 321
 - hsCallStatusValid, 321
 - hsInd, 321
 - hsIndValid, 321
 - lac, 321
 - lacValid, 321
 - MCC, 321
 - MNC, 321
 - networkIdValid, 321
 - psc, 321
 - pscValid, 321
 - regRejectInfoValid, 321
 - rejCause, 321
 - rejectSrvDomain, 321
 - sysInfoWCDMA, 321
- nas_acqOrderPref, 167
 - acqOrdeLen, 168
 - pAcqOrder, 168
- nas_callBarStatus, 173
 - csBarStatus, 174
 - psBarStatus, 174
- nas_cellParams, 181
 - pci, 182
 - rsrp, 182
 - rsrq, 182
 - rsi, 182
 - srxlev, 182
- nas_ciotAcqOrderPref, 182
 - ciotAcqOrderLen, 183
 - pCiotAcqOrder, 183
- nas_currentPLMN, 187
 - MCC, 188
 - MNC, 188
 - netDescr, 188
 - netDescrLength, 188
- nas_dataSrvCapabilities, 188
 - dataCapabilities, 189
 - dataCapabilitiesLen, 189
- nas_detailSvcInfo, 191
 - hdrHybrid, 192
 - hdrSrvStatus, 192
 - isSysForbidden, 192
 - srvCapability, 192
 - srvStatus, 192
- nas_dirNum, 193
 - dirNum, 193

- dirNumLen, 193
- nas_ecioListElement, 194
 - ecio, 194
 - radiolf, 195
- nas_errorRateListElement, 199
 - errorRate, 200
 - radiolf, 200
- nas_geranInstInfo, 202
 - geranArfcn, 203
 - geranBsicBcc, 203
 - geranBsicNcc, 203
 - geranRssi, 203
- nas_gsmCellInfo, 203
 - arfcn, 204
 - band1900, 204
 - bsicId, 204
 - cellIdValid, 204
 - rssi, 204
 - srxlev, 204
- nas_homeSIDNID, 215
 - numInstances, 215
 - SidNid, 215
- nas_infoInterFreq, 215
 - cell_resel_priority, 216
 - cellInterFreqParams, 216
 - cells_len, 216
 - earfcn, 216
 - threshXHigh, 216
 - threshXLow, 217
- nas_lteBandPrefExt, 217
 - bits_129_192, 218
 - bits_193_256, 218
 - bits_1_64, 218
 - bits_65_128, 218
- nas_lteEARFCN, 220
 - earfcn0, 221
 - earfcn1, 221
 - status, 221
- nas_lteGsmCellInfo, 221
 - cellReselPriority, 222
 - cells_len, 222
 - GsmCellInfo, 222
 - nccPermitted, 222
 - threshGsmHigh, 222
 - threshGsmLow, 222
- nas_lteOpModeTlv, 234
 - lteOpMode, 235
 - lteOpModeLen, 235
 - MCC, 235
 - MNC, 235
 - TlvPresent, 235
- nas_ltePCI, 235
 - earfcn, 235
 - pci, 235
 - status, 235
- nas_lteRsrpInformation, 235
 - rsrplevel, 236
- nas_lteSnrInformation, 239
 - snrlevel, 240
- nas_lteWcdmaCellInfo, 243
 - cellReselPriority, 244
 - cellsLen, 244
 - threshXhigh, 244
 - threshXlow, 244
 - uarfcn, 244
 - WCDMACellInfo, 244
- nas_major
 - NASLteNasReleaseInfoTlv, 327
- nas_minBasedIMSI, 244
 - imsiM1112, 245
 - imsiMS1, 245
 - imsiMS2, 245
 - mccM, 245
- nas_minor
 - NASLteNasReleaseInfoTlv, 327
- nas_namName, 246
 - namName, 247
 - namNameLen, 247
- nas_netSelectionPref, 247
 - mcc, 248
 - mnc, 248
 - netReg, 248
- nas_nmrCellInfo, 252
 - nmrArfcn, 253
 - nmrBsic, 253
 - nmrCellID, 253
 - nmrLac, 253
 - nmrPlmn, 253
 - nmrRxLev, 253
- nas_nr5gBandPref, 253
 - bits_129_192, 254
 - bits_193_256, 254
 - bits_1_64, 254
 - bits_65_128, 254
- nas_operatorNameString, 259
 - PLMNName, 259
- nas_operatorPLMNList, 260
 - numInstance, 261
 - PLMNData, 261
- nas_protocolSubtypeElement, 271
 - AccessMac, 272
 - AuthProt, 272
 - ControlMac, 272
 - EncryptProt, 272
 - ForwardMac, 272
 - IdleState, 272
 - KeyExchange, 272
 - MultDisc, 272
 - PhysicalLayer, 272
 - ReverseMac, 272
 - SecProt, 272
 - VirtStream, 272
- nas_qaQmi3Gpp2TimeZone, 272
 - daylightSavings, 273
 - leapSeconds, 273
 - localTimeOffset, 273

- nas_release
 - NASLteNasReleaseInfoTlv, 327
- nas_roamIndList, 286
 - numInstances, 286
 - radiolInterface, 287
 - roamIndicator, 287
- nas_rsrqInformation, 288
 - radiolf, 288
 - rsrq, 288
- nas_rxInfo, 290
 - ecio, 291
 - isRadioTuned, 291
 - phase, 291
 - rscp, 291
 - rsrp, 291
 - rxPower, 291
- nas_rxSignalStrengthListElement, 292
 - radiolf, 293
 - rxSignalStrength, 293
- nas_servSystem, 295
 - csAttachState, 296
 - numRadiolInterfaces, 296
 - psAttachState, 296
 - radiolInterface, 296
 - regState, 296
 - selNetwork, 296
- nas_serviceProviderName, 294
 - displayCondition, 294
 - spn, 294
 - spnLength, 294
- nas_sidNid, 296
 - nid, 296
 - sid, 296
- nas_sysInfoCommon, 302
 - isSysForbidden, 304
 - isSysForbiddenValid, 304
 - roamStatus, 304
 - roamStatusValid, 304
 - srvCapability, 304
 - srvCapabilityValid, 304
 - srvDomain, 304
 - srvDomainValid, 304
- nas_timeInfo, 307
 - day, 309
 - dayLtSavingAdj, 309
 - dayOfWeek, 309
 - hour, 309
 - minute, 309
 - month, 309
 - radiolInterface, 309
 - second, 309
 - timeZone, 309
 - TlvPresent, 309
 - year, 309
- nas_trueIMSI, 309
 - imsiT112, 310
 - imsiTS1, 310
 - imsiTS2, 310
 - imsiTaddrNum, 310
 - mccT, 310
- nas_txInfo, 310
 - isInTraffic, 310
 - txPower, 310
- nas_umtsLTENbrCell, 313
 - cellsTDD, 314
 - earfcn, 314
 - pci, 314
 - rsrp, 314
 - rsrq, 314
 - srxlev, 314
- nas_wcdmaCellInfo, 315
 - cpich_ecno, 316
 - cpich_rscp, 316
 - psc, 316
 - srxlev, 316
- nas_wcdmaUARFCN, 321
 - status, 322
 - uarfcn, 322
- NasGetLTECphyCaInfo, 324
 - PhyCaAggPcellInfo, 325
 - PhyCaAggScellArray, 325
 - PhyCaAggScellIDBw, 325
 - PhyCaAggScellIndType, 325
 - PhyCaAggScellIndex, 325
 - PhyCaAggScellInfo, 325
- nasQmisNasPcsDigit
 - nas_QmisNasSlqsNasPCICellInfo, 276
- nasQmisNasSlqsNasPCICellInfo
 - nas_QmisNasSlqsNasPCIIInfo, 277
- nasRelInfoTlv
 - NASQmiCbkNasSwiOTAMessageInd, 336
- nccPermitted
 - nas_lteGsmCellInfo, 222
- NeighborSetCnt
 - nas_NetworkStat1x, 250
- netDescr
 - nas_currentPLMN, 188
- netDescrLength
 - nas_currentPLMN, 188
- netInfoLen
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- netReg
 - nas_netSelectionPref, 248
- NetSelPref
 - nas_NetSelPrefTlv, 248
 - NASNetSelPreferenceTlv, 328
- NetworkID
 - unpack_nas_SLQSGetServingSystem_t, 773
- networkID
 - nas_CDMA SysInfo, 180
- networkIdValid
 - nas_CDMA SysInfo, 180
 - nas_GSM SysInfo, 208
 - nas_LTE SysInfo, 243
 - nas_WCDMA SysInfo, 321

- networkInfo
 - voice_USSDNotificationNetworkInfo, 1079
- networkInfoLen
 - unpack_wds_SLQSGetCurrDataSystemStat_t, 998
- NetworkType
 - currNetworkInfo, 56
 - wds_currNetworkInfo, 1083
- NewIMSRegistration
 - unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 708
- NewImsRegStatus
 - unpack_imsa_SLQSGetImsaRegStatus_t, 704
- NewMMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 863
- newMTMessageTlv, 342
 - MTMessageInfo, 342
 - TlvPresent, 342
- newPINLen
 - uim_unblockUIMPIN, 604
- newPINVal
 - uim_unblockUIMPIN, 604
- newPasswd
 - pack_voice_SLQSVoiceSetCallBarringPassword_t, 505
- newPasswdAgain
 - pack_voice_SLQSVoiceSetCallBarringPassword_t, 505
- newPin
 - pack_dms_UIMUnblockPIN_t, 368
- newPwd
 - voice_newPwdData, 1069
- newPwdAgain
 - voice_newPwdData, 1069
- newValue
 - pack_dms_UIMChangePIN_t, 364
- nextHeader
 - LibPackTFTIDParams, 144
- nid
 - nas_CDMAInfo, 176
 - nas_sidNid, 296
 - unpack_nas_GetHomeNetwork_t, 749
- nmrArfcn
 - nas_nmrCellInfo, 253
- nmrBsic
 - nas_nmrCellInfo, 253
- nmrCellID
 - nas_nmrCellInfo, 253
- nmrInst
 - nas_GERANInfo, 202
- nmrLac
 - nas_nmrCellInfo, 253
- nmrPlmn
 - nas_nmrCellInfo, 253
- nmrRxLev
 - nas_nmrCellInfo, 253
- noReplyTimer
 - voice_callFWExtInfo, 1049
- voice_callFWInfo, 1050
- notifType
 - unpack_voice_SUPSNotificationCallback_ind_t, 968
- notification
 - unpack_omaDmNotificationsTlv_t, 820
- notification_Type
 - unpack_voice_USSDNotificationCallback_ind_t, 968
- notificationType
 - sMSEtwsMessageInfo, 564
- notused
 - unpack_dms_SetCrashAction_t, 646
- Nr5gBandPrefbits1_64
 - NASNr5gBandPrefTlv, 329
- Nr5gBandPrefbits_129_192
 - NASNr5gBandPrefTlv, 329
- Nr5gBandPrefbits_193_256
 - NASNr5gBandPrefTlv, 329
- Nr5gBandPrefbits_65_128
 - NASNr5gBandPrefTlv, 329
- nr5gCellStatus
 - nas_NR5GCellStatusTlv, 254
- num_instances
 - DMScustSettingList, 75
 - sms_qaQmi3GPP2BroadcastCfgInfo, 557
 - sms_qaQmi3GPPBroadcastCfgInfo, 557
- numApp
 - slotInf, 549
 - uim_physlotInfo, 594
 - uim_slotInfo, 602
- numCrashes
 - crashInformation, 55
- numEntries
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 671
- numFeatures
 - uim_personalizationStatus, 593
- numFiles
 - uim_registerRefresh, 598
- NumFlows
 - unpack_qos_SLQSSetQosEventCallback_ind_t, 844
- numInstance
 - nas_operatorPLMNList, 261
 - nas_PLMNNetworkName, 268
- numInstances
 - nas_homeSIDNID, 215
 - nas_roamIndList, 286
 - nas_SccRxInfo, 294
 - sms_getMsgWaitingInfo, 552
 - unpack_sms_SLQSWmsMessageWaitingCall-Back_ind_t, 878
 - voice_arrAlertingPattern, 1036
 - voice_arrAlertingType, 1037
 - voice_arrAlphaID, 1037
 - voice_arrCalledPartyNum, 1038
 - voice_arrCallEndReason, 1039
 - voice_arrCallInfo, 1039

- voice_arrConnectPartyNum, 1040
- voice_arrDiagInfo, 1040
- voice_arrRedirPartyNum, 1041
- voice_arrRemotePartyName, 1041
- voice_arrRemotePartyNum, 1042
- voice_arrSvcOption, 1042
- voice_arrUUSInfo, 1043
- voice_getCallFWExtInfo, 1067
- voice_getCallFWInfo, 1067
- wds_DomainNameList, 1091
- wds_PCSCFFQDNAddressList, 1095
- wds_PCSCFIPv4ServerAddressList, 1096
- numLen
 - voice_calledPartyInfo, 1045
 - voice_callFWExtInfo, 1049
 - voice_callFWInfo, 1051
 - voice_callingPartyInfo, 1054
 - voice_peerNumberInfo, 1071
 - voice_redirNumInfo, 1074
 - voice_remotePartyNum, 1076
- numOfFiles
 - uim_refreshevent, 597
- numOfRoutes
 - sms_setRoutesReq, 562
- numOpt
 - wds_DHCPLeaseOptTlv, 1085
 - wds_DHCPv4OptionList, 1089
 - wdsDhcpv4OptionList, 1103
- numPl
 - voice_peerNumberInfo, 1071
- NumPilots
 - nas_PilotSetData, 266
- numPlan
 - voice_calledPartyInfo, 1045
 - voice_callFWExtInfo, 1050
 - voice_callingPartyInfo, 1054
 - voice_connectNumInfo, 1059
 - voice_peerNumberInfo, 1071
 - voice_redirNumInfo, 1074
- numPresInd
 - voice_connectNumInfo, 1059
- numQosFlow
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- numRadiInterfaces
 - nas_servSystem, 296
- NumRxFilters
 - unpack_qos_QosFlowInfo_t, 838
- numSI
 - voice_peerNumberInfo, 1071
- numSlot
 - uim_cardStatus, 583
- NumSupUSBComps
 - unpack_dms_GetUSBComp_t, 642
- NumTxFilters
 - unpack_qos_QosFlowInfo_t, 839
- numType
 - voice_calledPartyInfo, 1045
 - voice_callFWExtInfo, 1050
 - voice_callingPartyInfo, 1054
 - voice_connectNumInfo, 1060
 - voice_peerNumberInfo, 1071
 - voice_redirNumInfo, 1074
- number
 - voice_calledPartyInfo, 1045
 - voice_callFWExtInfo, 1049
 - voice_callFWInfo, 1050
 - voice_callingPartyInfo, 1054
 - voice_ECTNum, 1064
 - voice_peerNumberInfo, 1071
 - voice_redirNumInfo, 1074
- NumberOfPhySlot
 - uim_GetSlotsStatusTlv, 591
- NumberOfPhySlotInfo
 - uim_GetSlotsInfoTlv, 590
- numberPlan
 - voice_callFwdTypeAndPlan, 1048
- numberType
 - voice_callFwdTypeAndPlan, 1048
- nwldValid
 - nas_NR5GSystemInfoTlv, 258
- NxtHdrProto
 - unpack_qos_swiQosFilter_t, 850
- OKtoRefresh
 - pack_uim_SLQSUIMRefreshOK_t, 488
- OMADMEEnable
 - pack_swioma_SLQSOMADMSetSettingsExt_t, 477
- OMADMEEnabled
 - unpack_swiavms_SLQSAVMSGetSettings_t, 886
 - unpack_swiavms_SLQSAVMSGetSettings_v2_t, 888
 - unpack_swioma_SLQSOMADMGetSettings_t, 908
- OMADMLogEnable
 - pack_swioma_SLQSOMADMSetSettingsExt_t, 477
- OTASPStatus
 - unpack_voice_OTASPStatusCallback_ind_t, 931
- oddEvenInd
 - voice_calledPartySubAdd, 1046
- OfflineReason
 - unpack_dms_GetPower_t, 638
- offset
 - uim_readTransparentInfo, 595
- oldPINLen
 - uim_changeUIMPIN, 584
- oldPINVal
 - uim_changeUIMPIN, 584
- oldPasswd
 - pack_voice_SLQSVoiceSetCallBarringPassword_t, 505
- oldValue
 - pack_dms_UIMChangePIN_t, 364
- opMode
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, 653
- opaqueId
 - pack_loc_SLQSLOCInjectSensorData_t, 390

- operatingMode
 - dms_OperatingModeTlv, 66
- OperatingModeTlv
 - unpack_dms_SetEventReport_ind_t, 647
- operation
 - pack_pds_SetPDSDefaults_t, 435
 - uim_depersionalizationInformation, 585
- OperationMode
 - unpack_dms_GetPower_t, 638
- optCode
 - wds_DHCPopt, 1087
 - wds_DHCPv4Option, 1089
 - wdsDhcpv4Option, 1103
- optList
 - wds_DHCPLeaseOptTlv, 1085
- optListData
 - wds_DHCPLeaseOptTlv, 1086
- optVal
 - wds_DHCPv4Option, 1089
 - wdsDhcpv4Option, 1103
- optValLen
 - wds_DHCPopt, 1087
 - wds_DHCPv4Option, 1089
 - wdsDhcpv4Option, 1103
- otaMsgTlv
 - NASQmiCbkNasSwiOTAMessageInd, 336
- p1Status
 - unpack_dms_UIMGetPINStatus_t, 683
- p1UnblockRetriesLeft
 - unpack_dms_UIMGetPINStatus_t, 683
- p1VerifyRetriesLeft
 - unpack_dms_UIMGetPINStatus_t, 683
- p2Status
 - unpack_dms_UIMGetPINStatus_t, 683
- p2UnblockRetriesLeft
 - unpack_dms_UIMGetPINStatus_t, 683
- p2VerifyRetriesLeft
 - unpack_dms_UIMGetPINStatus_t, 683
- p3GPP2TimeInfo
 - unpack_nas_SLQSGetNetworkTime_t, 764
- p3GPPTimeInfo
 - unpack_nas_SLQSGetNetworkTime_t, 764
- p3GppNetworkInfoInstances
 - unpack_nas_PerformNetworkScan_t, 757
- p3GppNetworkInstanceSize
 - unpack_nas_PerformNetworkScan_t, 757
- p3gppRelease
 - pack_wds_SLQSSet3GPPConfigItem_t, 528
- pAAASPI
 - pack_wds_SetMobileIPProfile_t, 521
- PACK_WDS_IPV4
 - wds.h, 1526
- PACK_WDS_IPV6
 - wds.h, 1526
- pAMRStatus
 - pack_voice_SLQSVoiceGetConfig_t, 501
- pAPN
 - PackSwiAVMSSettingsAPNInfo, 540
- 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_swiaVms_SLQSAVMSSetSettings_t, 465
 - pack_swiaVms_SLQSAVMSSetSettings_v2_t, 467
 - unpack_swiaVms_SLQSAVMSSetSettings_t, 886
 - unpack_swiaVms_SLQSAVMSSetSettings_v2_t, 888
- 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, 736
- pAccelSamplesAccepted
 - unpack_loc_InjectSensorDataCallback_Ind_t, 727
- pAccelTempAcceptReady
 - unpack_loc_SensorStreamingCallback_Ind_t, 736
- pAccelTempSamplesAccepted
 - unpack_loc_InjectSensorDataCallback_Ind_t, 727
- pAccolc
 - unpack_nas_GetACCOLC_t, 743
- pAccuracy
 - unpack_pds_GetPDSDefaults_t, 821
- pAcqOrder
 - nas_acqOrderPref, 168
 - nas_AcqOrderPrefTlv, 168
- pAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 786
- pActPilotPNElements
 - nas_NetworkStat1x, 250
- pActiveTimer
 - pack_dms_SLQSSetPowerSaveModeConfig_t, 360

- unpack_dms_SLQSGetPowerSaveModeConfig_t, 667
- pAddCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 815
- pAddGSMSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pAddHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pAddLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pAddWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pAddrAllocPref
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 127
 - LibPackprofile_3GPP, 134
- pAddress
 - pack_wds_SetMobileIPProfile_t, 521
- pAirTimer
 - pack_voice_SLQSVoiceGetConfig_t, 501
- pAirTimerCnt
 - unpack_voice_SLQSVoiceGetConfig_t, 956
- pAirTimerConfig
 - pack_voice_SLQSVoiceSetConfig_t, 507
- pAirTimerStatus
 - unpack_voice_SLQSVoiceSetConfig_t, 960
- pAlertType
 - unpack_voice_SLQSVoiceGetCallInfo_t, 945
- pAlertingPattern
 - unpack_voice_SLQSVoiceGetCallInfo_t, 945
- pAllowLinger
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 139
- pAlphaID
 - unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 877
- pAlphaIDInfo
 - unpack_voice_SLQSOriginateUSSD_t, 933
 - unpack_voice_SLQSVoiceDialCall_t, 935
 - unpack_voice_SLQSVoiceGetCallBarring_t, 941
 - unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 942
 - unpack_voice_SLQSVoiceGetCallInfo_t, 945
 - unpack_voice_SLQSVoiceGetCallWaiting_t, 947
 - unpack_voice_SLQSVoiceGetCLIP_t, 948
 - unpack_voice_SLQSVoiceGetCLIR_t, 950
 - unpack_voice_SLQSVoiceGetCNAP_t, 951
 - unpack_voice_SLQSVoiceGetCOLP_t, 953
 - unpack_voice_SLQSVoiceGetCOLR_t, 954
 - unpack_voice_SLQSVoiceSetCallBarringPassword_t, 959
 - unpack_voice_SLQSVoiceSetSUPSService_t, 962
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 965
- pAltitudeAssumed
 - unpack_loc_PositionRpt_Ind_t, 735
- pAltitudeWrtEllipsoid
 - pack_pds_SLQSPDSInjectPositionData_t, 440
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pAltitudeWrtMeanSeaLevel
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pAltitudeWrtSealevel
 - pack_pds_SLQSPDSInjectPositionData_t, 440
- pAmrMode
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 691
- pAmrOctetAligned
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 691
- pAmrWBMode
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 691
- pAmrWBOctetAligned
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 692
- pAmrWbEnable
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 691
- pApnString
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 139
- pApnStringSize
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 139
- pApnname
 - pack_wds_SetDefaultProfile_t, 517
- pAppName
 - loc_LocApplicationInfo, 159
- pAppPriority
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pAppProvider
 - loc_LocApplicationInfo, 159
- pAppSubType
 - unpack_nas_SLQSSwiGetHDRProtSubtype_t, 808
- pAppType
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pAppUserData
 - LibpackProfile3GPPV2, 127
- pAppVersion
 - loc_LocApplicationInfo, 159
- pApplication
 - pack_nas_SetCDMANetworkParameters_t, 397
- pApplicationInfo
 - pack_loc_Start_t, 393
- pArrAlertingPattern
 - unpack_voice_allCallStatusCallback_ind_t, 928

- unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrAlertingType
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrAlphaID
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrCallEndReason
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrCallInfo
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrCalledPartyNum
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrConnectPartyNum
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrDiagInfo
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrRedirPartyNum
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrRemotePartyName
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrRemotePartyNum
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrSvcOption
 - unpack_voice_allCallStatusCallback_ind_t, 928
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pArrUUSInfo
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- pAuth
 - pack_wds_SLQSSStartDataSession_t, 532
- 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, 743
- pAuthTimeout
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pAuthenticateResult
 - unpack_uim_SLQSUIMAuthenticate_t, 918
- pAuthenticationPref
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 127
 - LibPackprofile_3GPP, 134
- pAutoAnsStatus
 - unpack_voice_SLQSVoiceSetConfig_t, 960
- pAutoAnswer
 - pack_voice_SLQSVoiceGetConfig_t, 501
 - pack_voice_SLQSVoiceSetConfig_t, 507
- pAutoAnswerStat
 - unpack_voice_SLQSVoiceGetConfig_t, 956
- pAutoReboot
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, 467
 - unpack_swiaavms_SLQSAVMSSetSettings_v2_t, 888
- pAutoSelection
 - unpack_uim_SLQSUIMGetConfiguration_t, 920
- pAutosdm
 - pack_swiaoma_SLQSOMADMSetSettings_t, 476
- pBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPref_t, 784
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 786
- pBdsSVInfo
 - pack_loc_Delete_Assist_Data_t, 379
- pBinaryUpdateSessionInfo
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, 883
 - unpack_swiaavms_SLQSAVMSSessionGetInfo_t, 889
- pBroadcast
 - pack_nas_SetCDMANetworkParameters_t, 397
- pBroadcastConfig
 - pack_sms_SLQSSetSmsBroadcastConfig_t, 455
 - unpack_sms_SLQSGetSmsBroadcastConfig_t, 867
- pBurstDTMFLengths
 - pack_voice_SLQSVoiceBurstDTMF_t, 495
- pCCResType
 - unpack_voice_SLQSVoiceGetCallBarring_t, 941
 - unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 942
 - unpack_voice_SLQSVoiceGetCallWaiting_t, 947
 - unpack_voice_SLQSVoiceGetCLIP_t, 948
 - unpack_voice_SLQSVoiceGetCLIR_t, 950
 - unpack_voice_SLQSVoiceGetCNAP_t, 951
 - unpack_voice_SLQSVoiceGetCOLP_t, 953
 - unpack_voice_SLQSVoiceGetCOLR_t, 954
 - unpack_voice_SLQSVoiceSetCallBarringPassword-t, 959
- pCCResultType
 - unpack_voice_SLQSVoiceDialCall_t, 935
 - unpack_voice_SLQSVoiceSetSUPSService_t, 962
- pCCSUPSType
 - unpack_voice_SLQSVoiceDialCall_t, 935
 - unpack_voice_SLQSVoiceGetCallBarring_t, 941

- unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 942
- unpack_voice_SLQSVoiceGetCallWaiting_t, 947
- unpack_voice_SLQSVoiceGetCLIP_t, 948
- unpack_voice_SLQSVoiceGetCLIR_t, 950
- unpack_voice_SLQSVoiceGetCNAP_t, 951
- unpack_voice_SLQSVoiceGetCOLP_t, 953
- unpack_voice_SLQSVoiceGetCOLR_t, 954
- unpack_voice_SLQSVoiceSetCallBarringPassword_t, 959
- unpack_voice_SLQSVoiceSetSUPSService_t, 962
- pCCSuppsType
 - unpack_voice_SLQSOriinateUSSD_t, 933
- pCDMABroadcastConfig
 - pack_sms_SLQSSetSmsBroadcastConfig_t, 455
 - unpack_sms_SLQSGetSmsBroadcastConfig_t, 867
- pCDMAChannel
 - unpack_nas_SLQSNasGet3GPP2Subscription_t, 789
- pCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 404
- pCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 404
- pCDMAECIOThreshList
 - nas_CDMAECIOThresh, 175
- pCDMAFrameErrRate
 - unpack_nas_SLQSGetErrorRate_t, 763
- pCDMAInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- pCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 404
- pCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pCDMARSSIThreshList
 - nas_CDMARSSIThresh, 177
- pCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 802
- pCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- PCICellInfoLen
 - nas_QmisNasSlqsNasPCInfo, 277
- pCLIPResp
 - unpack_voice_SLQSVoiceGetCLIP_t, 948
- pCLIPstatus
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pCLIRCause
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pCLIRResp
 - unpack_voice_SLQSVoiceGetCLIR_t, 950
- pCLIRType
 - pack_voice_SLQSVoiceDialCall_t, 496
- pCLIRstatus
- unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pCNAPResp
 - unpack_voice_SLQSVoiceGetCNAP_t, 951
- pCNAPstatus
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pCOLPResp
 - unpack_voice_SLQSVoiceGetCOLP_t, 953
- pCOLPstatus
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pCOLRResp
 - unpack_voice_SLQSVoiceGetCOLR_t, 954
- pCOLRstatus
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- PCSCFAddrPCO
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- PCSCFFQDNAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- pCSCFPortName
 - pack_ims_SLQSSetRegMgrConfig_t, 376
- pCSCFPortNameLen
 - pack_ims_SLQSSetRegMgrConfig_t, 376
- PCSFlag
 - LibPackProfileMnc, 141
- pCSGID
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
- PCTlv
 - unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 695
- pCUGIndex
 - unpack_voice_SUPSNotificationCallback_ind_t, 968
- pCUGInfo
 - pack_voice_SLQSVoiceDialCall_t, 496
- PCURTlv
 - unpack_ims_SLQSSMSCfgCallBack_ind_t, 700
- pCallActiveDuration
 - unpack_wds_GetSessionDurationV2_t, 992
- pCallBarPasswd
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 965
- pCallBarringPasswd
 - pack_voice_SLQSVoiceSetSUPSService_t, 509
- pCallFWNum
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pCallFWTimerVal
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pCallForwardingNumber
 - pack_voice_SLQSVoiceSetSUPSService_t, 509
- pCallFwdInfo
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 965
- pCallFwdTypeAndPlan

- pack_voice_SLQSVoiceSetSUPSService_t, 509
- pCallID
 - pack_voice_SLQSVoiceManageCalls_t, 503
 - pack_voice_SLQSVoiceSendFlash_t, 504
 - pack_voice_SLQSVoiceStartContDTMF_t, 510
 - unpack_voice_SLQSVoiceBurstDTMF_t, 934
 - unpack_voice_SLQSVoiceDialCall_t, 935
 - unpack_voice_SLQSVoiceGetCallBarring_t, 941
 - unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 942
 - unpack_voice_SLQSVoiceGetCallWaiting_t, 947
 - unpack_voice_SLQSVoiceGetCLIP_t, 948
 - unpack_voice_SLQSVoiceGetCLIR_t, 950
 - unpack_voice_SLQSVoiceGetCNAP_t, 951
 - unpack_voice_SLQSVoiceGetCOLP_t, 953
 - unpack_voice_SLQSVoiceGetCOLR_t, 954
 - unpack_voice_SLQSVoiceSendFlash_t, 957
 - unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 959
 - unpack_voice_SLQSVoiceSetSUPSService_t, 962
 - unpack_voice_SLQSVoiceStartContDTMF_t, 962
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
 - voice_burstDTMFInfo, 1044
- pCallId
 - pack_voice_SLQSVoiceAnswerCall_t, 494
 - pack_voice_SLQSVoiceEndCall_t, 497
 - unpack_voice_SLQSOOriginateUSSD_t, 933
 - unpack_voice_SLQSVoiceAnswerCall_t, 934
 - unpack_voice_SLQSVoiceEndCall_t, 936
- pCallInfo
 - unpack_voice_SLQSVoiceGetCallInfo_t, 945
- pCallPartySubAdd
 - pack_voice_SLQSVoiceDialCall_t, 496
- pCallType
 - pack_voice_SLQSVoiceDialCall_t, 496
- pCallWaitInd
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pCalledPartyInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pCallerIDInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pCallerNameInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pCallingPartyInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pCampedCiotLteOpMode
 - unpack_nas_SLQSGetSysInfo_t, 779
- pCardResult
 - unpack_uim_ReadTransparent_t, 915
 - unpack_uim_SLQSUIMAuthenticate_t, 918
 - unpack_uim_SLQSUIMGetFileAttributes_t, 921
 - unpack_uim_UnblockPinV2_t, 925
- pCardStatus
 - unpack_uim_GetCardStatus_t, 913
 - unpack_uim_GetCardStatusV2_t, 914
 - unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 924
- pCcResultType
 - unpack_voice_SLQSOOriginateUSSD_t, 933
- pCellDataMask
 - pack_pds_ResetPDSData_t, 433
- pCellDb
 - pack_loc_Delete_Assist_Data_t, 379
- pChangeDuration
 - pack_nas_SLQSInitiateNetworkRegistration_t, 401
- pChgDuration
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
- pCiotAcqOrder
 - nas_ciotAcqOrderPref, 183
 - nas_CiotAcqOrderPrefTlv, 184
- pCiotAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 786
- pCiotLteOpMode
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
- pCiotLteOpModePref
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 786
- pCiotOpModePref
 - pack_nas_PerformNetworkScanPCI_t, 395
- pClatFlag
 - LibpackProfile3GPPV2, 127
- pClientPerformOperationFlag
 - pack_swiaavms_SLQSAVMSSendSelection_t, 463
- pClkInfo
 - pack_loc_Delete_Assist_Data_t, 379
- pCodecSTGain
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pColorCode
 - unpack_nas_SLQSNasGetHDRColorCode_t, 794
- pConfig
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, 883
 - unpack_swiaavms_SLQSAVMSSessionGetInfo_t, 889
- pConfigAltitudeAssumed
 - pack_loc_Start_t, 393
- pConfigurationMask
 - pack_uim_SLQSUIMGetConfiguration_t, 484
- pConnectNumInfo
 - unpack_voice_SLQSVoiceGetCallInfo_t, 945
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pConnectionRequest
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, 883
- pConnectionRetryTimers
 - pack_swiaavms_SLQSAVMSSetSettings_t, 465
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, 467
 - unpack_swiaavms_SLQSAVMSSetSettings_t, 886
 - unpack_swiaavms_SLQSAVMSSetSettings_v2_t, 888
- pCreateProfileOut
 - unpack_wds_SLQSCreateProfile_t, 994

- pCsgId
 - unpack_nas_SLQSNasNetworkRejectCallback_-_Ind_t, [800](#)
- pCurAMRConfig
 - unpack_voice_SLQSVoiceGetConfig_t, [956](#)
- pCurPrefVoiceSO
 - unpack_voice_SLQSVoiceGetConfig_t, [956](#)
- pCurProfile
 - pack_wds_SLQSCreateProfile_t, [522](#)
- pCurVoiceDomainPref
 - unpack_voice_SLQSVoiceGetConfig_t, [956](#)
- pCurVoicePrivacyPref
 - unpack_voice_SLQSVoiceGetConfig_t, [956](#)
- pCurrChannelRateInd
 - pack_wds_SLQSWdsSetEventReport_t, [534](#)
- pCurrDataBearerTechInd
 - pack_wds_SLQSWdsSetEventReport_t, [534](#)
- pCurrImgInfo
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [671](#)
- pCurrPrefDataSysInd
 - pack_wds_SLQSWdsSetEventReport_t, [534](#)
- pCurrTTYMode
 - unpack_voice_SLQSVoiceGetConfig_t, [956](#)
- pCurrentPersonality
 - nas_HDRPersonality_Ind_Data, [211](#)
 - unpack_nas_SLQSSwiGetHDRPersonality_t, [807](#)
- pCurrentPrsnlty
 - unpack_nas_SLQSSwiGetHDRProtSubtype_t, [808](#)
- pCustSettingInfo
 - DMSgetCustomFeatureV2, [76](#)
- pCustSettingList
 - DMSgetCustomFeatureV2, [76](#)
- pCustomSCP
 - pack_nas_SetCDMANetworkParameters_t, [397](#)
- pCwtMute
 - pack_swiaudio_SLQSSetM2MAudioProfile_t, [460](#)
 - pack_swiaudio_SLQSSetM2MAVMute_t, [462](#)
- pCycleLen
 - pack_nas_SLQSNASSeteDRXParams_t, [412](#)
 - unpack_nas_SLQSNASGeteDRXParams_t, [792](#)
 - unpack_nas_SLQSNASGeteDRXParamsExt_t, [793](#)
- pDDTMInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, [410](#)
- PDOP
 - loc_precisionDilution, [160](#)
- PDPTType
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- pDRCPParams
 - unpack_nas_SLQSSwiGetHRPDStats_t, [809](#)
- pDTMFTXGain
 - pack_audio_SLQSSetAudioPathConfig_t, [346](#)
 - unpack_audio_SLQSGetAudioPathConfig_t, [607](#)
- pData
 - pack_cat_CATSendEnvelopeCommand_t, [349](#)
 - pack_cat_CATSendTerminalResponse_t, [350](#)
- pDataBearer
 - unpack_wds_GetDataBearerTechnology_t, [976](#)
- pDataBearerTechInd
 - pack_wds_SLQSWdsSetEventReport_t, [534](#)
- pDataCallStatusChangeInd
 - pack_wds_SLQSWdsSetEventReport_t, [534](#)
- pDataMode
 - LibpackProfile3GPP2, [118](#)
 - LibPackprofile_3GPP2, [140](#)
- pDataRate
 - LibpackProfile3GPP2, [118](#)
 - LibPackprofile_3GPP2, [140](#)
- pDataSessionStatus
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, [883](#)
- pDataSrc
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, [966](#)
- pDataStatusDetail
 - unpack_nas_SLQSSwiNetworkDebug_t, [812](#)
- pDataSystemStatusChangeInd
 - pack_wds_SLQSWdsSetEventReport_t, [534](#)
- pDayltSavAdj
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, [801](#)
- pDefaultPDNEnabled
 - pack_wds_SLQSSet3GPPConfigItem_t, [528](#)
- pDeferTime
 - pack_swiaavms_SLQSAVMSSendSelection_t, [463](#)
 - pack_swioama_SLQSOMADMSendSelection_t, [474](#)
- pDestSMSContent
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, [361](#)
 - packgetDyingGaspCfg, [536](#)
- pDestSMSNum
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, [361](#)
 - packgetDyingGaspCfg, [536](#)
- pDetachAction
 - pack_nas_SLQSSwiPSDetach_t, [432](#)
- pDevCurSubsCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevExplicitCfgIndex
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevMaxActDataSubsCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevMaxCfgListCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevMaxSubsCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevMultiSimCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevMultiSimVoiceDataCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevSrvCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevSubsFeatureModeCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevSubsVoiceDataCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevVoiceCaps

- unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDevVoiceDataCaps
 - unpack_dms_GetDeviceCapabilitiesV2_t, [623](#)
- pDeviceConfigDetail
 - unpack_nas_SLQSSwiNetworkDebug_t, [812](#)
- pDiagInfo
 - unpack_voice_SLQSVoiceGetCallInfo_t, [945](#)
- pDigitBuff
 - voice_burstDTMFInfo, [1044](#)
- pDirNum
 - unpack_nas_SLQSNasGet3GPP2Subscription_t, [789](#)
- pDisplInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, [971](#)
- pDnsWithDHCPFlag
 - LibpackProfile3GPPV2, [127](#)
- pDormancyStatusInd
 - pack_wds_SLQSWdsSetEventReport_t, [535](#)
- pDualStandByPrefInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, [410](#)
- pDuration
 - unpack_pds_GetXTRAVValidity_t, [825](#)
- pDurationDueToOOS
 - unpack_dms_SLQSGetPowerSaveModeConfig_t, [667](#)
- pDurationThreshold
 - unpack_dms_SLQSGetPowerSaveModeConfig_t, [667](#)
- pECIOThresList
 - nas_ECIOThresh, [195](#)
- pECIOThresh
 - pack_nas_SLQSConfigSigInfo_t, [399](#)
- pECMode
 - pack_audio_SLQSSetAudioPathConfig_t, [346](#)
 - unpack_audio_SLQSGetAudioPathConfig_t, [607](#)
- pECTNum
 - unpack_voice_SUPSNotificationCallback_ind_t, [968](#)
- PER
 - nas_NetworkStatEVDO, [252](#)
- pEVDOPageMonPerChangeInd
 - pack_wds_SLQSWdsSetEventReport_t, [535](#)
- pEarMute
 - pack_swiaudio_SLQSSetM2MAudioProfile_t, [460](#)
- pEarlyWakeupTime
 - unpack_dms_SLQSGetPowerSaveModeConfig_t, [667](#)
- pEdrxChangeInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, [410](#)
- pEdrxCiotLteMode
 - pack_nas_SLQSNASSeteDRXParams_t, [412](#)
- pEdrxEnable
 - pack_nas_SLQSNASSeteDRXParams_t, [412](#)
 - unpack_nas_SLQSNASGeteDRXParams_t, [792](#)
 - unpack_nas_SLQSNASGeteDRXParamsExt_t, [793](#)
- pEdrxRAT
 - pack_nas_SLQSNASGeteDRXParamsExt_t, [407](#)
- unpack_nas_SLQSNASGeteDRXParamsExt_t, [793](#)
- pEdrxRatType
 - pack_nas_SLQSNASSeteDRXParams_t, [412](#)
- pEhrpdMTUSize
 - unpack_swidms_SLQSSwiDmsGetMTU_t, [893](#)
- pEmerMode
 - pack_nas_SLQSSetSysSelectionPref_t, [421](#)
 - pack_nas_SLQSSetSysSelectionPrefExt_t, [431](#)
 - unpack_nas_SLQSGetSysSelectionPref_t, [784](#)
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, [787](#)
- pEmergencyCategory
 - pack_voice_SLQSVoiceDialCall_t, [496](#)
- pEnableNotification
 - pack_wds_DHCPv4ClientLeaseChange_t, [511](#)
- pEnabled
 - pack_wds_SetMobileIPProfile_t, [521](#)
- pEnabledStatus
 - unpack_pds_GetPDSSState_t, [822](#)
- pEncryptData
 - pack_uim_ReadTransparent_t, [481](#)
- pEncryptedData
 - unpack_uim_ReadTransparent_t, [915](#)
- pEncryptedPIN1
 - pack_uim_VerifyPin_t, [491](#)
 - unpack_uim_ChangePin_t, [913](#)
 - unpack_uim_SetPinProtection_t, [916](#)
 - unpack_uim_UnblockPin_t, [924](#)
 - unpack_uim_UnblockPinV2_t, [925](#)
 - unpack_uim_VerifyPin_t, [926](#)
- pErrorRateInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, [410](#)
- pExtDisplInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, [971](#)
- pExtDispRecInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, [971](#)
- pExtErrCode
 - UnPackGetProfileSettingOut, [1023](#)
 - UnPackGetProfileSettingOutV2, [1023](#)
- pExtErrorCode
 - unpack_wds_SLQSMModifyProfile_t, [1009](#)
- pFailCause
 - unpack_voice_SLQSVoiceGetCallBarring_t, [941](#)
 - unpack_voice_SLQSVoiceGetCallForwarding-Status_t, [942](#)
 - unpack_voice_SLQSVoiceGetCallWaiting_t, [947](#)
 - unpack_voice_SLQSVoiceGetCLIP_t, [949](#)
 - unpack_voice_SLQSVoiceGetCLIR_t, [950](#)
 - unpack_voice_SLQSVoiceGetCNAP_t, [951](#)
 - unpack_voice_SLQSVoiceGetCOLP_t, [953](#)
 - unpack_voice_SLQSVoiceGetCOLR_t, [954](#)
 - unpack_voice_SLQSVoiceManageCalls_t, [957](#)
 - unpack_voice_SLQSVoiceSetCallBarringPassword-t, [959](#)
 - unpack_voice_SLQSVoiceSetSUPSService_t, [962](#)
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, [966](#)
- pFailureReason

- unpack_wds_SLQSSStartDataSession_t, 1019
- pFileAttributes
 - unpack_uim_SLQSUIMGetFileAttributes_t, 921
- pFixId
 - unpack_loc_PositionRpt_Ind_t, 735
- pFlag
 - audio_RXPCMIIRFiltr, 45
 - audio_TXPCMIIRFiltr, 47
- pFlashPayLd
 - pack_voice_SLQSVoiceSendFlash_t, 504
- pFlashType
 - pack_voice_SLQSVoiceSendFlash_t, 504
- pFollowOnDC
 - sms_sendAsyncsmsParams, 560
- pForbiddenNetworks3GPP
 - unpack_nas_SLQSNASGetForbiddenNetworks_t, 794
- pForceOnDC
 - sms_sendAsyncsmsParams, 560
- pForceRev0
 - pack_nas_SetCDMANetworkParameters_t, 397
- pFwAutoCheck
 - pack_swima_SLQSOMADMSetSettings_t, 476
- pFwAutoSDM
 - pack_swiaavms_SLQSAVMSSetSettings_t, 465
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, 467
- pGERANInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- pGPRMinimumQoS
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 127
 - LibPackprofile_3GPP, 134
- pGPRRequestedQoS
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 127
 - LibPackprofile_3GPP, 134
- pGPSDataMask
 - pack_pds_ResetPDSDData_t, 434
- pGPSWeek
 - unpack_pds_GetXTRAVValidity_t, 825
- pGPSWeekOffset
 - unpack_pds_GetXTRAVValidity_t, 825
- pGSMBER
 - unpack_nas_SLQSGetErrorRate_t, 763
- pGSMCallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pGSMCipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pGSMRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pGSMRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pGSMRSSIThreshList
 - nas_GSMRSSIThresh, 205
- pGSMSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 802
- pGSMSrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pGSMSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pGWAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPref_t, 784
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pGenerator
 - pack_swiaudio_SLQSGetM2MAudioProfile_t, 457
 - pack_swiaudio_SLQSSetM2MAudioProfile_t, 460
- pGetCallFWExtInfo
 - unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 943
- pGetCallFWInfo
 - unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 943
- pGetCustomInput
 - DMSgetCustomFeatureV2, 76
- pGetDyingGaspCfg
 - unpack_dms_SLQSSwiGetDyingGaspCfg_t, 670
- pGetDyingGaspStatistics
 - unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 670
- pGetIndicationRegInfo
 - unpack_sms_SLQSGetIndicationRegister_t, 865
- pGetMsgWaitingInfoResp
 - unpack_sms_SLQSGetMessageWaiting_t, 865
- pGetSlotsInfoTlv
 - unpack_uim_SLQSUIMGetSlotsStatusV2_t, 922
- pGetSlotsStatusTlv
 - unpack_uim_SLQSUIMGetSlotsStatusV2_t, 922
- pGetTransLayerInfo
 - unpack_sms_SLQSGetTransLayerInfo_t, 868
- pGetTransNWRegInfo
 - unpack_sms_SLQSGetTransNWRegInfo_t, 869
- pGnssData
 - pack_loc_Delete_Assist_Data_t, 379
- pGpsTime
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pGyroAcceptReady
 - unpack_loc_SensorStreamingCallback_Ind_t, 737
- pGyroSamplesAccepted
 - unpack_loc_InjectSensorDataCallback_Ind_t, 728
- pGyroTempAcceptReady
 - unpack_loc_SensorStreamingCallback_Ind_t, 737
- pGyroTempSamplesAccepted
 - unpack_loc_InjectSensorDataCallback_Ind_t, 728
- pHA2002bis
 - pack_wds_SetMobileIPParameters_t, 519
- pHAAAuthenticator
 - pack_wds_SetMobileIPParameters_t, 519
- pHASPI
 - pack_wds_SetMobileIPProfile_t, 521

- pHDRECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHDRECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHDRECIOThreshList
 - nas_HDRECIOThresh, [210](#)
- pHDRIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHDRIOTThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHDRIOTThreshList
 - nas_HDRIOTThresh, [210](#)
- pHDRNewUATIAssInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, [410](#)
- pHDRPackErrRate
 - unpack_nas_SLQSGetErrorRate_t, [763](#)
- pHDRPersInd
 - unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t, [811](#)
- pHRRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHRRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHRRSSIThreshList
 - nas_HRRSSIThresh, [211](#)
- pHRSINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHRSINRThresList
 - nas_HRSINRThresh, [212](#)
- pHRSINRThresh
 - pack_nas_SLQSConfigSigInfo_t, [399](#)
 - pack_nas_SLQSNasConfigSigInfo2_t, [405](#)
- pHRSINRThreshList
 - nas_HRSINRThreshold, [213](#)
- pHDRSessionCloseInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, [410](#)
- pHDRSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, [802](#)
- pHDRSrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, [779](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [816](#)
- pHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [779](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [816](#)
- pHTTPStatus
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, [883](#)
- pHWWatchdog
 - unpack_swidms_SLQSSwiDmsGetHWWatchdog_t, [892](#)
- pHaltSubscription
 - unpack_uim_SLQSUIMGetConfiguration_t, [920](#)
- pHeading
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHeadingUnc
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHomeSIDNID
 - unpack_nas_SLQSNasGet3GPP2Subscription_t, [789](#)
- pHorCirConf
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
- pHorConfidence
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHorEllpConf
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
- pHorReliability
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHorUncCircular
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHorUncEllipseOrientAzimuth
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHorUncEllipseSemiMajor
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHorUncEllipseSemiMinor
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pHorizontalAccuracyLvl
 - pack_loc_Start_t, [393](#)
- pHorizontalConfidence
 - pack_pds_SLQSPDSInjectPositionData_t, [440](#)
- pHorizontalUncCircular
 - pack_pds_SLQSPDSInjectPositionData_t, [440](#)
- pHotSwapStatus
 - unpack_uim_GetCardStatus_t, [913](#)
 - unpack_uim_GetCardStatusV2_t, [914](#)
- pHrpdMTUSize
 - unpack_swidms_SLQSSwiDmsGetMTU_t, [893](#)
- pHwConfig
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, [531](#)
 - unpack_wds_SLQSSetDHCPv4ClientConfig_t, [1018](#)
- PI
 - voice_calledPartyInfo, [1045](#)
 - voice_callerIDInfo, [1047](#)
 - voice_callFWExtInfo, [1050](#)
 - voice_callingPartyInfo, [1054](#)
 - voice_redirNumInfo, [1074](#)
- PIFACEId
 - pack_swiaudio_SLQSSetM2MAudioAVCFG_t, [459](#)
- pIMSDomain
 - pack_ims_SLQSSetIMSUserConfig_t, [373](#)
 - unpack_ims_SLQSGetIMSUserConfig_t, [689](#)
- pIMSDomainLen
 - pack_ims_SLQSSetIMSUserConfig_t, [373](#)
 - unpack_ims_SLQSGetIMSUserConfig_t, [689](#)
- pIMSTestMode
 - pack_ims_SLQSSetRegMgrConfig_t, [376](#)
 - unpack_ims_SLQSGetRegMgrConfig_t, [693](#)
- plOTThresList
 - nas_IOTThresh, [217](#)

- plOTthresh
 - pack_nas_SLQSSConfigSigInfo_t, 399
- plPV6DelegFlag
 - LibpackProfile3GPPV2, 128
- plPv4AddrInfo
 - unpack_loc_GetServer_Ind_t, 724
- plPv4AddrPref
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 128
 - LibPackprofile_3GPP, 134
- plPv4Config
 - pack_loc_SLQSLOCSetServer_t, 392
- plPv6AddPref
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 128
 - LibPackprofile_3GPP, 134
- plPv6AddrInfo
 - unpack_loc_GetServer_Ind_t, 724
- plPv6Config
 - pack_loc_SLQSLOCSetServer_t, 392
- plWLANtoLTEHandoverFlag
 - LibpackProfile3GPPV2, 128
- plgnoreHotSwapSwitch
 - pack_uim_SLQSUIPowerUp_t, 486
- plmCnFlag
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 128
 - LibPackprofile_3GPP, 134
- plmgeList
 - pack_fms_SetImagesPreference_t, 370
 - unpack_fms_GetImagesPreference_t, 686
- plndicationToken
 - pack_uim_ChangePin_t, 480
 - pack_uim_ReadTransparent_t, 481
 - pack_uim_SetPinProtection_t, 482
 - pack_uim_SLQSUIAuthenticate_t, 483
 - pack_uim_SLQSUIGetFileAttributes_t, 485
 - pack_uim_UnblockPin_t, 490
 - pack_uim_VerifyPin_t, 491
 - unpack_uim_ChangePin_t, 913
 - unpack_uim_ReadTransparent_t, 915
 - unpack_uim_SetPinProtection_t, 916
 - unpack_uim_SLQSUIAuthenticate_t, 918
 - unpack_uim_SLQSUIGetFileAttributes_t, 921
 - unpack_uim_UnblockPin_t, 925
 - unpack_uim_UnblockPinV2_t, 926
 - unpack_uim_VerifyPin_t, 926
- plInfo
 - pack_voice_AnswerUSSD_t, 492
 - pack_voice_OriginateUSSD_t, 492
- plInstanceSize
 - nas_RfBandInfoList, 282
- plInstancesSize
 - nas_RfBandInfoExtFormat, 281
 - nas_RfBandwidthInfo, 283
 - nas_RfDedicatedBandInfo, 284
- plInterfaceCfg
 - unpack_swidms_SLQSSwiDmsGetUsbComp_t, 895
- plIntermediateReportState
 - pack_loc_Start_t, 393
- plInterval
 - unpack_pds_GetPDSDDefaults_t, 821
 - unpack_pds_GetXTRAAutomaticDownload_t, 824
- plpcpAckTimeout
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- plpcpCreqRetryCount
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- plsPcscfAddressNedded
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pKeyReferenceID
 - pack_uim_ChangePin_t, 480
 - pack_uim_SetPinProtection_t, 482
 - pack_uim_UnblockPin_t, 490
 - pack_uim_VerifyPin_t, 491
- PLMNData
 - nas_operatorPLMNList, 261
- PLMNName
 - nas_operatorNameString, 259
- PLMNNetName
 - nas_PLMNNetworkName, 268
- PLMNRecID
 - nas_OperatorPLMNData, 260
- plLTEAttachProfile
 - pack_wds_SLQSSet3GPPConfigItem_t, 528
- plLTEAttachProfileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 528
- plTEBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPref_t, 784
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- plTEBandPrefExt
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- plTECphyCa
 - pack_nas_SLQSNasIndicationRegisterExt_t, 410
- plTEInfo
 - unpack_nas_SLQSNasSwtModemStatus_t, 804
- plTEInfoInterfreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- plTEInfoIntrafreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- plTEInfoNeighboringGSM
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- plTEInfoNeighboringWCDMA
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- plTEOperationMode
 - nas_LTEOperationMode, 233
 - unpack_nas_SLQSNasGetRFInfo_t, 795
- plTERSRPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405

- pLTERSRPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTERSRPThreshList
 - nas_LTERSRPThresh, 236
- pLTERSRQDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTERSRQThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTERSRQThreshList
 - nas_LTERSRQThresh, 237
- pLTERSSIDDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTERSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTERSSIThreshList
 - nas_LTERSSIThresh, 237
- pLTESNRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTESNRThresList
 - nas_LTESNRThresh, 240
- pLTESNRThresh
 - pack_nas_SLQSConfigSigInfo_t, 399
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTESNRThreshList
 - nas_LTESNRThreshold, 241
- pLTESigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 802
- pLTESigRptCfg
 - pack_nas_SLQSConfigSigInfo_t, 399
- pLTESigRptConfig
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pLTESrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pLTEVoiceSupportSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 779
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pLTEtoWLANHandoverFlag
 - LibpackProfile3GPPV2, 128
- pLastCallActiveDuration
 - unpack_wds_GetSessionDurationV2_t, 992
- pLastCallDuration
 - unpack_wds_GetSessionDurationV2_t, 992
- pLatitude
 - pack_pds_SLQSPDSInjectPositionData_t, 440
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pLcpAckTimeout
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pLcpCreqRetryCount
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pLeapSeconds
 - unpack_loc_PositionRpt_Ind_t, 735
- pLineCtrlInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pLinktimer
 - pack_sms_SendSMS_t, 448
 - sms_sendAsynsmsParams, 560
- pLongitude
 - pack_pds_SLQSPDSInjectPositionData_t, 440
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pLteCiotOpModeTlv
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pLteEARFCN
 - pack_nas_SLQSNASSwiSetChannelLock_t, 414
 - unpack_nas_SLQSNASSwiGetChannelLock_t, 803
- pLteM1BandPref
 - pack_nas_PerformNetworkScanPCI_t, 395
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pLteNB1BandPref
 - pack_nas_PerformNetworkScanPCI_t, 395
- pLteNb1BandPref
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pLteOpMode
 - pack_nas_SLQSNASGeteDRXParamsExt_t, 407
 - unpack_nas_SLQSNASGeteDRXParamsExt_t, 793
 - unpack_nas_SLQSNasNetworkRejectCallback_Ind_t, 800
- pLteOpModeTlv
 - unpack_nas_PerformNetworkScan_t, 757
- pLtePCI
 - pack_nas_SLQSNASSwiSetChannelLock_t, 414
 - unpack_nas_SLQSNASSwiGetChannelLock_t, 803
- pLteRoamPDPTYPE
 - LibpackProfile3GPPV2, 128
- pMCC
 - unpack_nas_GetHomeNetwork3GPP2_t, 748
- pMICGainSelect
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pMIPStatusInd
 - pack_wds_SLQSWdsSetEventReport_t, 535
- pMNA
 - pack_wds_SetMobileIPProfile_t, 521
- pMNC
 - unpack_nas_GetHomeNetwork3GPP2_t, 748
- pMNCIncPCSDigStat
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
- pMNHA
 - pack_wds_SetMobileIPProfile_t, 521
- pMNRInfo
 - pack_nas_SLQSIInitiateNetworkRegistration_t, 401
- pMTUSize3gpp
 - unpack_swidms_SLQSSwiDmsGetMTU_t, 893
- pMagneticDeviation

- unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pManagedRoamingInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 410
- pMaxPDN
 - LibpackProfile3GPPV2, 128
- pMaxPDNTimer
 - LibpackProfile3GPPV2, 128
- pMaxStorageSizeReq
 - pack_sms_SLQSSmsGetMaxStorageSize_t, 456
- pMaxStorageSizeResp
 - unpack_sms_SLQSSmsGetMaxStorageSize_t, 873
- pMcc
 - LibpackProfile3GPPV2, 128
- pMessage
 - pack_sms_SaveSMS_t, 448
 - pack_sms_SendSMS_t, 449
 - sms_sendAsynsmsParams, 560
- pMessageIndex
 - pack_sms_SLQSDDeleteSMS_t, 450
 - unpack_sms_SaveSMS_t, 861
- pMessageMode
 - pack_sms_SLQSDDeleteSMS_t, 450
 - pack_sms_SLQSGetSMS_t, 451
 - pack_sms_SLQSGetSMSList_t, 452
 - pack_sms_SLQSMModifySMSStatus_t, 453
 - sms_maxStorageSizeReq, 554
- pMessageProtocol
 - unpack_sms_SLQSSmsGetMessageProtocol_t, 873
- pMessageTag
 - pack_sms_SLQSDDeleteSMS_t, 450
- pMicMute
 - pack_swiaudio_SLQSSetM2MAudioProfile_t, 460
- pMinBasedIMSI
 - unpack_nas_SLQSNasGet3GPP2Subscription_t, 789
- pMinIntervalTime
 - pack_loc_Start_t, 393
- pMinSessionExpiryTimer
 - pack_ims_SLQSSetIMSVoIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVoIPConfig_t, 692
- pMnc
 - LibpackProfile3GPPV2, 128
- pMncPcsDigitStatus
 - pack_nas_SLQSInitiateNetworkRegistration_t, 401
- pMncPcsStatus
 - pack_nas_SLQSGetPLMNName_t, 400
- pMode
 - pack_wds_SetMobileIPParameters_t, 519
 - unpack_loc_GetOpMode_Ind_t, 723
- pModePref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPref_t, 784
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pMsisdFlag
 - LibpackProfile3GPPV2, 128
- pMuxID
 - pack_qos_BindDataPort_t, 445
- pNAI
 - pack_wds_SetMobileIPProfile_t, 521
- pNAMNameInfo
 - unpack_nas_SLQSNasGet3GPP2Subscription_t, 789
- pNID
 - unpack_nas_GetHomeNetwork3GPP2_t, 748
- pNITZInformation
 - unpack_nas_SLQSGetOperatorNameData_t, 765
- pNR5GCellStatus
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pNR5GSerStatTlv
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pNR5GSystemInfoTlv
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pNSEnable
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pNSSAudioCtrl
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pNSSRelease
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- PNTlv
 - unpack_ims_SLQSRcgMgrCfgCallBack_ind_t, 695
- pNamID
 - pack_voice_SLQSVoiceGetConfig_t, 501
- pName
 - pack_wds_SetDefaultProfile_t, 517
 - unpack_nas_GetHomeNetwork3GPP2_t, 748
- pNegoDnsSrvrPref
 - LibpackProfile3GPP2, 118
 - LibPackprofile_3GPP2, 140
- pNeighborSetPilotPN
 - nas_NetworkStat1x, 250
- pNetSelPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 431
 - unpack_nas_SLQSGetSysSelectionPref_t, 784
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pNetworkMode
 - pack_pds_SLQSGetAGPSConfig_t, 438
 - pack_pds_SLQSSetAGPSConfig_t, 441
- pNetworkRejectInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pNetworkStat1x
 - unpack_nas_SLQSSwiNetworkDebug_t, 812
- pNetworkStatEVDO
 - unpack_nas_SLQSSwiNetworkDebug_t, 812
- pNetworkTimeInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pNetworkType
 - pack_nas_PerformNetworkScanPCI_t, 395
- pNewPwdData
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966

- pNotifStore
 - pack_swiaavms_SLQSAVMSSetSettings_t, [465](#)
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, [467](#)
- pNotification
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, [883](#)
 - unpack_swiaavms_SLQSAVMSSessionGetInfo_t, [890](#)
- pNotificationStore
 - unpack_swiaavms_SLQSAVMSGetSettings_t, [886](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [888](#)
- pNr5gBandPref
 - pack_nas_SLQSSetSysSelectionPrefExt_t, [431](#)
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, [787](#)
- pNumberOfPhySlot
 - unpack_uim_SLQSUIMGetSlotsStatus_t, [921](#)
- pNw2DescDisp
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
- pNw2DescEnc
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
- pNw2DescLen
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
- pNw2MCC
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
- pNw2MNC
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
- pNw2Name
 - unpack_nas_GetHomeNetwork3GPP2_t, [748](#)
- pOTASPStatus
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, [939](#)
 - unpack_voice_SLQSVoiceGetCallInfo_t, [945](#)
- pObjectVer
 - unpack_nas_SLQSSwiNetworkDebug_t, [812](#)
- pOffLength
 - unpack_voice_DTMFEventCallback_ind_t, [929](#)
- pOnLength
 - unpack_voice_DTMFEventCallback_ind_t, [929](#)
- pOpaqueIdentifier
 - unpack_loc_InjectSensorDataCallback_Ind_t, [728](#)
- pOperation
 - unpack_pds_GetPDSDDefaults_t, [821](#)
- pOperatorNameString
 - unpack_nas_SLQSGetOperatorNameData_t, [765](#)
- pOperatorPCOID
 - LibpackProfile3GPPV2, [128](#)
- pOperatorPLMNList
 - unpack_nas_SLQSGetOperatorNameData_t, [765](#)
- pOptList
 - wds_DHCPv4OptionList, [1089](#)
 - wdsDhcpv4OptionList, [1104](#)
- pOptVal
 - wds_DHCPOpt, [1087](#)
- pOverridePDPTYPE
 - LibpackProfile3GPPV2, [128](#)
- pPCInfo
 - unpack_nas_PerformNetworkScan_t, [757](#)
- pPCMPParams
 - pack_swiaudio_SLQSSetM2MAudioAVCFG_t, [459](#)
- pPCOIDList
 - LibpackProfile3GPPV2, [128](#)
- pPCSCFPort
 - unpack_ims_SLQSGetRegMgrConfig_t, [693](#)
- pPCSInstance
 - unpack_nas_PerformNetworkScan_t, [757](#)
- pPCSInstanceSize
 - unpack_nas_PerformNetworkScan_t, [757](#)
- pPDNDisconnectWaitTimer
 - LibpackProfile3GPPV2, [128](#)
- pPDNInactivTimeout
 - LibpackProfile3GPP, [112](#)
 - LibpackProfile3GPPV2, [128](#)
 - LibPackprofile_3GPP, [134](#)
- pPDNInactivTimeout3GPP2
 - LibpackProfile3GPP2, [118](#)
 - LibPackprofile_3GPP2, [140](#)
- pPDNThrottleTimer
 - LibpackProfile3GPPV2, [128](#)
- pPDNWaitTimer
 - LibpackProfile3GPPV2, [128](#)
- pPDPTYPE
 - LibpackProfile3GPP, [112](#)
 - LibpackProfile3GPPV2, [128](#)
 - LibPackprofile_3GPP, [135](#)
- pPLMNNetworkName
 - unpack_nas_SLQSGetOperatorNameData_t, [765](#)
- pPRLPref
 - pack_nas_SLQSSetSysSelectionPref_t, [421](#)
 - pack_nas_SLQSSetSysSelectionPrefExt_t, [431](#)
 - unpack_nas_SLQSGetSysSelectionPref_t, [784](#)
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, [787](#)
- pPWD
 - PackSwiAVMSSettingsAPNInfo, [540](#)
- pPackageID
 - pack_swiaavms_SLQSAVMSSendSelection_t, [463](#)
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, [883](#)
 - unpack_swiaavms_SLQSAVMSSessionGetInfo_t, [890](#)
- pPagingTimeWindow
 - pack_nas_SLQSNASSeteDRXParams_t, [412](#)
 - unpack_nas_SLQSNASGeteDRXParams_t, [792](#)
 - unpack_nas_SLQSNASGeteDRXParamsExt_t, [793](#)
- pPass
 - pack_wds_SLQSStartDataSession_t, [532](#)
- pPassword
 - LibpackProfile3GPP, [112](#)
 - LibpackProfile3GPPV2, [128](#)
 - LibPackprofile_3GPP, [134](#)
 - pack_wds_SetDefaultProfile_t, [517](#)
- 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, [465](#)
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, [467](#)
- pPeriodicUpdateTimer
 - pack_dms_SLQSSetPowerSaveModeConfig_t, [360](#)
 - unpack_dms_SLQSSetPowerSaveModeConfig_t, [667](#)
- pPeripheralEndPointID
 - pack_qos_BindDataPort_t, [445](#)
- pPeroidsInfo
 - unpack_swiaavms_SLQSAVMSGetSettings_t, [886](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [888](#)
- pPersistFlag
 - LibpackProfile3GPPV2, [129](#)
- pPersonalityListLength
 - nas_HDRPersonality_Ind_Data, [211](#)
 - unpack_nas_SLQSSwiGetHDRPersonality_t, [807](#)
 - unpack_nas_SLQSSwiGetHDRProtSubtype_t, [808](#)
- pPersonalizationStatus
 - unpack_uim_SLQSUIGetConfiguration_t, [920](#)
- pPhoneCtxtURI
 - pack_ims_SLQSSetIMSSMSConfig_t, [372](#)
 - unpack_ims_SLQSSetIMSSMSConfig_t, [688](#)
- pPhoneCtxtURILen
 - pack_ims_SLQSSetIMSSMSConfig_t, [372](#)
 - unpack_ims_SLQSSetIMSSMSConfig_t, [688](#)
- pPilotSetData
 - unpack_nas_SLQSSwiGetHRPDStats_t, [809](#)
- pPilotSetInfo
 - nas_PilotSetData, [266](#)
- pPlmnId
 - unpack_nas_SLQSNasNetworkRejectCallback_Ind_t, [800](#)
- pPollingTimer
 - pack_swiaavms_SLQSAVMSSetSettings_t, [465](#)
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, [467](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_t, [886](#)
 - unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [888](#)
- pPositionSource
 - pack_pds_SLQSPDSInjectPositionData_t, [440](#)
- pPppSessCloseTimer1x
 - LibpackProfile3GPP2, [118](#)
 - LibPackprofile_3GPP2, [140](#)
- pPppSessCloseTimerDO
 - LibpackProfile3GPP2, [119](#)
 - LibPackprofile_3GPP2, [140](#)
- pPrecisionDilution
 - unpack_loc_BestAvailPos_Ind_t, [716](#)
 - unpack_loc_PositionRpt_Ind_t, [735](#)
- pPrefVoiceDomain
 - pack_voice_SLQSVoiceSetConfig_t, [507](#)
- pPrefVoicePrivacy
 - pack_voice_SLQSVoiceGetConfig_t, [501](#)
- pPrefVoiceSO
 - pack_voice_SLQSVoiceGetConfig_t, [501](#)
 - pack_voice_SLQSVoiceSetConfig_t, [507](#)
- pPrefVoiceSOSStatus
 - unpack_voice_SLQSVoiceSetConfig_t, [960](#)
- pPreference
 - unpack_pds_GetXTRANetwork_t, [824](#)
- pPriCSCFPort
 - pack_ims_SLQSSetRegMgrConfig_t, [376](#)
- pPriCSCFPortName
 - unpack_ims_SLQSSetRegMgrConfig_t, [693](#)
- pPriCSCFPortNameLen
 - unpack_ims_SLQSSetRegMgrConfig_t, [693](#)
- 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, [521](#)
- pPrimaryID
 - LibpackProfile3GPP, [112](#)
 - LibpackProfile3GPPV2, [129](#)
 - LibPackprofile_3GPP, [135](#)
- pPrimaryV4DnsAddress
 - LibpackProfile3GPP2, [119](#)
 - LibPackprofile_3GPP2, [140](#)
- pProfileID

- unpack_wds_SLQSCreateProfile_t, 994
- pProfileId
 - pack_wds_SLQSCreateProfile_t, 522
 - pack_wds_SLQSModifyProfile_t, 527
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, 530
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, 531
- pProfileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 528
- pProfileSettings
 - unpack_wds_SLQSGetProfileSettings_t, 1004
 - unpack_wds_SLQSGetProfileSettingsV2_t, 1005
- pProfileType
 - pack_wds_SLQSCreateProfile_t, 522
 - pack_wds_SLQSModifyProfile_t, 527
- pProfileName
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pProfileNameSize
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pProtoSubTypeElmnt
 - unpack_nas_SLQSSwiGetHDRProtSubtype_t, 808
- pProtocol
 - pack_nas_SetCDMANetworkParameters_t, 397
- pProtocolSubtypeElement
 - nas_HDRPersonality_Ind_Data, 211
 - unpack_nas_SLQSSwiGetHDRPersonality_t, 807
- pPsmEnableState
 - pack_dms_SLQSSetPowerSaveModeConfig_t, 360
 - unpack_dms_SLQSGetPowerSaveModeConfig_t, 667
- pQosClassID
 - LibpackProfile3GPP, 112
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pRAT
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 432
- pRATInstance
 - unpack_nas_PerformNetworkScan_t, 757
- pRATInstanceSize
 - unpack_nas_PerformNetworkScan_t, 757
- pRATType
 - LibpackProfile3GPP2, 119
 - LibPackprofile_3GPP2, 140
- pRFBandInfoParam
 - nas_RfBandInfoList, 282
- PRIStrng
 - unpack_dms_GetFirmwareRevision_t, 629
 - unpack_dms_GetFirmwareRevisions_t, 631
- PRLEnable
 - pack_swima_SLQSOMADMSetSettingsExt_t, 477
- PRLInd
 - unpack_nas_SLQSGetServingSystem_t, 773
- PRLPTlv
 - NASQmiCbkNasSystemSelPrefInd, 338
- PRLPref
 - nas_PRLPrefTlv, 271
 - NASPRLPreferenceTlv, 335
- pRSRPThresList
 - nas_RSRPThresh, 288
- pRSRPThresh
 - pack_nas_SLQSConfigSigInfo_t, 399
- pRSRQThresList
 - nas_RSRQThresh, 289
- pRSRQThresh
 - pack_nas_SLQSConfigSigInfo_t, 399
- pRSSIThresList
 - nas_RSSIThresh, 290
- pRSSIThresh
 - pack_nas_SLQSConfigSigInfo_t, 399
- pRTPRTCPInactTimer
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 692
- pRXAGCList
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pRXAIG
 - audio_RXAGCList, 42
- pRXAVCAGCSwitch
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pRXAVCList
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pRXChain0Info
 - unpack_nas_SLQSNasGetTxRxInfo_t, 797
- pRXChain1Info
 - unpack_nas_SLQSNasGetTxRxInfo_t, 797
- pRXComprSlope
 - audio_RXAGCList, 43
- pRXComprThres
 - audio_RXAGCList, 43
- pRXDroppedCount
 - unpack_wds_GetPacketStatistics_t, 988
- pRXExpSlope
 - audio_RXAGCList, 43
- pRXExpThres
 - audio_RXAGCList, 43
- pRXOKBytesLastCall
 - unpack_wds_GetPacketStatistics_t, 988
- pRXOkBytesCount
 - unpack_wds_GetPacketStatistics_t, 988
- pRXPCMIIRFtr
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pRXPacketErrors
 - unpack_wds_GetPacketStatistics_t, 988
- pRXPacketOverflows
 - unpack_wds_GetPacketStatistics_t, 988
- pRXPacketSuccesses
 - unpack_wds_GetPacketStatistics_t, 988

- pRXStaticGain
 - audio_RXAGCList, [43](#)
- pRXTotalBytes
 - unpack_wds_GetByteTotals_t, [974](#)
- pRadioInterface
 - unpack_nas_SLQSNasNetworkTimeCallback_ind_t, [801](#)
- pRandomizationWindow
 - unpack_dms_SLQSGetPowerSaveModeConfig_t, [667](#)
- pRankIndicatorInd
 - pack_nas_SLQSNasSwiIndicationRegister_t, [414](#)
- pRatDisabledMask
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, [787](#)
- pReRegPeriod
 - pack_wds_SetMobileIPParameters_t, [519](#)
- pReRegTraffic
 - pack_wds_SetMobileIPParameters_t, [519](#)
- pReadResult
 - unpack_uim_ReadTransparent_t, [915](#)
- pReason
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, [966](#)
- pReasonMask
 - unpack_dms_GetOfflineReason_t, [637](#)
- pRecurrenceType
 - pack_loc_Start_t, [393](#)
- pRedirNumInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, [971](#)
- pRefreshEvent
 - unpack_uim_SLQSUIMRefreshGetLastEvent_t, [923](#)
- pRegCallStatInfoEvt
 - sms_getIndicationReg, [552](#)
 - sms_setIndicationReg, [561](#)
- pRegDTMFEvents
 - pack_voice_SLQSVoiceIndicationRegister_t, [502](#)
- pRegInd
 - sms_getTransLayerInfo, [553](#)
- pRegMgrConfigEvents
 - pack_ims_SLQSImsConfigIndicationRegister_t, [371](#)
- pRegStatus
 - sms_getTransNWRegInfo, [553](#)
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, [883](#)
- pRegTransLayerInfoEvt
 - sms_getIndicationReg, [552](#)
 - sms_setIndicationReg, [561](#)
- pRegTransNWRegInfoEvt
 - sms_getIndicationReg, [552](#)
 - sms_setIndicationReg, [561](#)
- pRegVoicePrivacyEvents
 - pack_voice_SLQSVoiceIndicationRegister_t, [502](#)
- pRejectReason
 - pack_swiaavms_SLQSAVMSSendSelection_t, [463](#)
 - pack_swima_SLQSOMADMSendSelection_t, [474](#)
- pRemainingRetries
 - unpack_uim_ChangePin_t, [913](#)
 - unpack_uim_SetPinProtection_t, [916](#)
 - unpack_uim_SLQSUIMDepersonalization_t, [918](#)
 - unpack_uim_UnblockPin_t, [925](#)
 - unpack_uim_UnblockPinV2_t, [926](#)
 - unpack_uim_VerifyPin_t, [926](#)
- pRemotePartyName
 - unpack_voice_SLQSVoiceGetCallInfo_t, [945](#)
- pRemotePartyNum
 - unpack_voice_SLQSVoiceGetCallInfo_t, [945](#)
- pReportChannelRate
 - pack_wds_SLQSGetDUNCallInfo_t, [524](#)
- pReportConnStatus
 - pack_wds_SLQSGetDUNCallInfo_t, [524](#)
- pReportDataBearerTech
 - pack_wds_SLQSGetDUNCallInfo_t, [524](#)
- pReportDormStatus
 - pack_wds_SLQSGetDUNCallInfo_t, [524](#)
- pReqSettings
 - pack_wds_SLQSGetRuntimeSettings_t, [526](#)
- pRequestOptionList
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, [531](#)
 - unpack_wds_SLQSSetDHCPv4ClientConfig_t, [1018](#)
- pRequestedTag
 - pack_sms_SLQSGetSMSList_t, [452](#)
- pRetryInterval
 - pack_wds_SetMobileIPParameters_t, [519](#)
- pRetryLimit
 - pack_wds_SetMobileIPParameters_t, [519](#)
- pRetryMessage
 - sms_sendAsynsmsParams, [560](#)
- pRetryMessageId
 - sms_sendAsynsmsParams, [560](#)
- pRevInUse
 - nas_CDMASysInfo, [181](#)
- pRevInUseValid
 - nas_CDMASysInfo, [181](#)
- pRevTunneling
 - pack_wds_SetMobileIPProfile_t, [521](#)
- pRfBandInfoExtFormat
 - unpack_nas_SLQSNasGetRFInfo_t, [795](#)
- pRfBandInfoExtFormatParam
 - nas_RfBandInfoExtFormat, [281](#)
- pRfBandwidthInfo
 - unpack_nas_SLQSNasGetRFInfo_t, [795](#)
- pRfBandwidthInfoParam
 - nas_RfBandwidthInfo, [283](#)
- pRfDedicatedBandInfo
 - unpack_nas_SLQSNasGetRFInfo_t, [795](#)
- pRfDedicatedBandInfoParam
 - nas_RfDedicatedBandInfo, [284](#)
- pRingBackTimer
 - pack_ims_SLQSSetIMSVoIPConfig_t, [375](#)
 - unpack_ims_SLQSGetIMSVoIPConfig_t, [692](#)
- pRingingTimer
 - pack_ims_SLQSSetIMSVoIPConfig_t, [375](#)

- unpack_ims_SLQSGetIMSVolIPConfig_t, 692
- pRoamDisallowFlag
 - LibpackProfile3GPPV2, 129
- pRoamPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 432
 - unpack_nas_SLQSGetSysSelectionPref_t, 784
 - unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pRoamTimer
 - pack_voice_SLQSVoiceGetConfig_t, 501
- pRoamTimerCnt
 - unpack_voice_SLQSVoiceGetConfig_t, 956
- pRoamTimerConfig
 - pack_voice_SLQSVoiceSetConfig_t, 507
- pRoamTimerStatus
 - unpack_voice_SLQSVoiceSetConfig_t, 960
- pRoaming
 - pack_nas_SetCDMANetworkParameters_t, 397
- pRscp
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 802
- PSDomain
 - unpack_nas_GetServingNetwork_t, 754
- pSID
 - unpack_nas_GetHomeNetwork3GPP2_t, 748
- pSIODDataPort
 - pack_qos_BindDataPort_t, 445
- pSIPConfigEvents
 - pack_ims_SLQSImsConfigIndicationRegister_t, 371
- pSIPLocalPort
 - pack_ims_SLQSSetSIPConfig_t, 377
 - unpack_ims_SLQSGetSIPConfig_t, 694
- PSMCfgChangelInfo
 - pack_dms_SetIndicationRegister_t, 357
- pSMSAttemptedFlag
 - packgetDyingGaspStatistics, 537
- pSMSCAddress
 - pack_sms_SetSMSCAddress_t, 449
 - unpack_sms_GetSMSCAddress_t, 860
- pSMSCType
 - pack_sms_SetSMSCAddress_t, 449
 - unpack_sms_GetSMSCAddress_t, 860
- pSMSConfigEvents
 - pack_ims_SLQSImsConfigIndicationRegister_t, 371
- pSMSFormat
 - pack_ims_SLQSSetIMSSMSConfig_t, 372
 - unpack_ims_SLQSGetIMSSMSConfig_t, 688
- pSMSOverIPNwInd
 - pack_ims_SLQSSetIMSSMSConfig_t, 372
 - unpack_ims_SLQSGetIMSSMSConfig_t, 688
- PSMStatus
 - pack_dms_SetIndicationRegister_t, 357
- pSPC
 - pack_nas_SetCDMANetworkParameters_t, 397
 - pack_wds_SetMobileIPParameters_t, 519
- pSV
 - loc_BdsSVInfo, 150
 - loc_SVInfo, 164
- pSVInfo
 - pack_loc_Delete_Assist_Data_t, 379
- pSarRFState
 - unpack_sar_SLQSGetRfSarState_t, 859
- pSatelliteInfo
 - unpack_loc_GnssSvInfo_Ind_t, 725
- pScanResult
 - unpack_nas_PerformNetworkScan_t, 757
- pScanType
 - pack_nas_PerformNetworkScanPCI_t, 395
- pSccRxInfo
 - unpack_nas_SLQSSwiGetLteSccRxInfo_t, 810
- pScrAmrEnable
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 692
- pScrAmrWbEnable
 - pack_ims_SLQSSetIMSVolIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVolIPConfig_t, 692
- 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, 521
- pSecondaryV4DnsAddress
 - LibpackProfile3GPP2, 119
 - LibPackprofile_3GPP2, 140
- pSectorID
 - nas_NetworkStatEVDO, 252
- pSendSmsParams
 - pack_sms_SLQSSendAsyncSMS_t, 453
- pSensorDataUsage
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pServerAddress
 - pack_pds_SLQSSetAGPSConfig_t, 441
 - unpack_pds_SLQSGetAGPSConfig_t, 829
- pServerPort
 - pack_pds_SLQSSetAGPSConfig_t, 441
 - unpack_pds_SLQSGetAGPSConfig_t, 829
- pServerURL
 - pack_pds_SLQSSetAGPSConfig_t, 441
 - unpack_pds_SLQSGetAGPSConfig_t, 829
- pServerURLLength
 - pack_pds_SLQSSetAGPSConfig_t, 441
 - unpack_pds_SLQSGetAGPSConfig_t, 829

- pServiceClass
 - pack_voice_SLQSVoiceSetSUPSService_t, 509
- pServiceOption
 - sms_sendAsynsmsParams, 560
- pServingSystemInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pSessionExpiryTimer
 - pack_ims_SLQSSetIMSVoIPConfig_t, 375
 - unpack_ims_SLQSGetIMSVoIPConfig_t, 692
- pSessionType
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, 883
- pSetIndicationRegReq
 - pack_sms_SLQSSetIndicationRegister_t, 454
- pSetRoutesReq
 - pack_sms_SLQSSmsSetRoutes_t, 456
- pSettingResp
 - unpack_ims_SLQSGetIMSSMSConfig_t, 688
 - unpack_ims_SLQSGetIMSUserConfig_t, 689
 - unpack_ims_SLQSGetIMSVoIPConfig_t, 692
 - unpack_ims_SLQSGetRegMgrConfig_t, 693
 - unpack_ims_SLQSGetSIPConfig_t, 694
 - unpack_ims_SLQSSetIMSSMSConfig_t, 696
 - unpack_ims_SLQSSetIMSUserConfig_t, 696
 - unpack_ims_SLQSSetIMSVoIPConfig_t, 697
 - unpack_ims_SLQSSetRegMgrConfig_t, 697
 - unpack_ims_SLQSSetSIPConfig_t, 698
- pSigCompEnabled
 - pack_ims_SLQSSetSIPConfig_t, 376
 - unpack_ims_SLQSGetSIPConfig_t, 694
- pSigIndReq
 - pack_nas_SLQSSetSignalStrengthsCallback_t, 416
- pSignalInfo
 - unpack_voice_VoiceInfoRecCallback_ind_t, 971
- pSignalStrengthInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pSimBusyStatus
 - unpack_uim_GetCardStatusV2_t, 914
- pSmsOnIms
 - sms_sendAsynsmsParams, 560
- pSmsStorage
 - unpack_sms_SLQSSwiGetSMSStorage_t, 874
- pSourceIP
 - LibPackTFTIDParams, 144
- pSpeedHorizontal
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pSpeedUnc
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pSpeedVertical
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pSpeedVerticalUnc
 - unpack_loc_BestAvailPos_Ind_t, 716
- pSrvDomainPref
 - pack_nas_SLQSSetSysSelectionPref_t, 421
- pack_nas_SLQSSetSysSelectionPrefExt_t, 432
- unpack_nas_SLQSGetSysSelectionPref_t, 784
- unpack_nas_SLQSGetSysSelectionPrefExt_t, 787
- pSrvOpt
 - unpack_voice_SLQSVoiceGetCallInfo_t, 945
- pSrvRegRestriction
 - pack_nas_SLQSSetSysSelectionPref_t, 421
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 432
- pSrvProviderName
 - unpack_nas_SLQSGetOperatorNameData_t, 765
- 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, 514
- pSubscribeTimer
 - pack_ims_SLQSSetSIPConfig_t, 377
 - unpack_ims_SLQSGetSIPConfig_t, 694
- pSubscriptionInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pSupportEmergencyCalls
 - LibpackProfile3GPP, 113
 - LibpackProfile3GPV2, 129
 - LibPackprofile_3GPP, 135
- pSupportedBitmasks
 - unpack_swidms_SLQSSwiDmsGetUsbComp_t, 895
- pSuppressSysInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pSuppsNotifEvents
 - pack_voice_SLQSVoiceIndicationRegister_t, 502
- pSvUsedforFix
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pSvcClass
 - pack_voice_SLQSVoiceGetCallBarring_t, 498
 - pack_voice_SLQSVoiceGetCallForwardingStatus_t, 499
 - pack_voice_SLQSVoiceGetCallWaiting_t, 500
 - unpack_voice_SLQSVoiceGetCallBarring_t, 941
 - unpack_voice_SLQSVoiceGetCallWaiting_t, 947
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966

- pSvcType
 - pack_voice_SLQSVoiceDialCall_t, 497
- pSysInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pSysInfoNoChange
 - unpack_nas_SLQSSysInfoCallback_ind_t, 816
- pSysTime
 - unpack_dms_GetNetworkTimeV2_t, 636
- pSystemSelectionInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 411
- pTDSCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMAECIOThreshList
 - nas_TDSCDMAECIOThresh, 305
- pTDSCDMARSCPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMARSCPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMARSCPThreshList
 - nas_TDSCDMARSCPThresh, 305
- pTDSCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMARSSIThreshList
 - nas_TDSCDMARSSIThresh, 306
- pTDSCDMASINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMASINRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 405
- pTDSCDMASINRThreshList
 - nas_TDSCDMASINRThresh, 307
- pTDSCDMASigInfoExt
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 802
- pTFTID1Params
 - LibpackProfile3GPP, 113
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pTFTID2Params
 - LibpackProfile3GPP, 113
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pTTYConfigStatus
 - unpack_voice_SLQSVoiceSetConfig_t, 960
- pTTYMode
 - pack_voice_SLQSVoiceGetConfig_t, 501
 - pack_voice_SLQSVoiceSetConfig_t, 507
- pTXAGCList
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pTXAIG
 - audio_TXAGCList, 45
- pTXAVCSwitch
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pTXComprSlope
 - audio_TXAGCList, 46
- pTXComprThres
 - audio_TXAGCList, 46
- pTXDroppedCount
 - unpack_wds_GetPacketStatistics_t, 988
- pTXExpSlope
 - audio_TXAGCList, 46
- pTXExpThres
 - audio_TXAGCList, 46
- pTXGain
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pTXInfo
 - unpack_nas_SLQSNasGetTxRxInfo_t, 797
- pTXOKBytesLastCall
 - unpack_wds_GetPacketStatistics_t, 988
- pTXOkBytesCount
 - unpack_wds_GetPacketStatistics_t, 988
- pTXPCMIIRFtr
 - pack_audio_SLQSSetAudioPathConfig_t, 346
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
- pTXPacketErrors
 - unpack_wds_GetPacketStatistics_t, 988
- pTXPacketOverflows
 - unpack_wds_GetPacketStatistics_t, 988
- pTXPacketSuccesses
 - unpack_wds_GetPacketStatistics_t, 988
- pTXStaticGain
 - audio_TXAGCList, 46
- pTXTotalBytes
 - unpack_wds_GetByteTotals_t, 974
- pTdsdmaBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 422
 - pack_nas_SLQSSetSysSelectionPrefExt_t, 432
- pTech
 - pack_wds_SLQSSStartDataSession_t, 532
- pTechnologyMask
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pTempReport
 - pack_dms_SwiSetEventReport_t, 363
- pTimeSrc
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pTimeStamp
 - pack_pds_SLQSPDSInjectPositionData_t, 440
 - packgetDyingGaspStatistics, 537
- pTimeType
 - pack_pds_SLQSPDSInjectPositionData_t, 440
- pTimeUnc
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pTimeZone
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind-
_t, 801
- pTimeout
 - unpack_pds_GetPDSDefaults_t, 821
- pTimer

- pack_nas_SLQSNasSwiIndicationRegister_t, 414
- pTimerSIPReg
 - pack_ims_SLQSSetSIPConfig_t, 377
 - unpack_ims_SLQSGetSIPConfig_t, 694
- pTimerT1
 - pack_ims_SLQSSetSIPConfig_t, 377
 - unpack_ims_SLQSGetSIPConfig_t, 694
- pTimerT2
 - pack_ims_SLQSSetSIPConfig_t, 377
 - unpack_ims_SLQSGetSIPConfig_t, 694
- pTimerTf
 - pack_ims_SLQSSetSIPConfig_t, 377
 - unpack_ims_SLQSGetSIPConfig_t, 694
- pTimerVal
 - pack_voice_SLQSVoiceSetSUPSService_t, 509
- pTimestampUtc
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 735
- pTrackingStatus
 - unpack_pds_GetPDSSState_t, 822
- pTransLayerInfo
 - sms_getTransLayerInfo, 553
 - unpack_sms_SLQSTransLayerInfoCallback_ind_t, 875
- pTransferStatInd
 - pack_wds_SLQSGetDUNCallInfo_t, 524
 - pack_wds_SLQSWdsSetEventReport_t, 535
- pTransferStatusReport
 - sms_setRoutesReq, 562
- pTrueIMSI
 - unpack_nas_SLQSNasGet3GPP2Subscription_t, 789
- pUATI
 - unpack_nas_SLQSSwiGetHRPStats_t, 809
- pUIMStatusReport
 - pack_dms_SwiSetEventReport_t, 363
- pUMTSCellID
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- pUMTSInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 791
- pUMTSMinQoS
 - LibpackProfile3GPP, 113
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pUMTSMinQoSSigInd
 - 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, 724
- pURLAddr
 - pack_loc_SLQSLOCSetServer_t, 392
- pUSSDInfo
 - unpack_voice_SLQSOriginateUSSD_t, 933
- pUSSInfo
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, 966
- pUUSInFo
 - pack_voice_SLQSVoiceDialCall_t, 497
- pUUSInfo
 - unpack_voice_SLQSVoiceGetCallInfo_t, 946
- pUimAutoSwitchActSlot
 - unpack_dms_SLQSDmsSwiGetUimSelection_t, 656
- pUimSlotsStatus
 - unpack_uim_SLQSUIMGetSlotsStatus_t, 921
- pUmtsRoamPDPTType
 - LibpackProfile3GPPV2, 129
- pUname
 - PackSwiAVMSSettingsAPNInfo, 540
- pUsbMTUSize
 - unpack_swidms_SLQSSwiDmsGetMTU_t, 893
- pUser
 - pack_wds_SLQSSStartDataSession_t, 532
- pUserConfigEvents
 - pack_ims_SLQSImsConfigIndicationRegister_t, 371
- pUserData
 - sms_sendAsynsmsParams, 560
- pUserId
 - LibpackProfile3GPP2, 119
 - LibPackprofile_3GPP2, 140
- pUserIdSize
 - LibpackProfile3GPP2, 119
 - LibPackprofile_3GPP2, 140
- pUsername
 - LibpackProfile3GPP, 113
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
 - pack_wds_SetDefaultProfile_t, 517
- pUsernameSize
 - LibpackProfile3GPP, 113
 - LibpackProfile3GPPV2, 129
 - LibPackprofile_3GPP, 135
- pUsrTime
 - unpack_dms_GetNetworkTimeV2_t, 636
- pValidCardStatus
 - unpack_uim_GetCardStatusV2_t, 914
- pVerboseFailReasonType
 - unpack_wds_SLQSSStartDataSession_t, 1020
- pVerboseFailureReason
 - unpack_wds_SLQSSStartDataSession_t, 1020
- pVertConfidence
 - unpack_loc_BestAvailPos_Ind_t, 716
 - unpack_loc_PositionRpt_Ind_t, 736
- pVertReliability
 - unpack_loc_BestAvailPos_Ind_t, 717
 - unpack_loc_PositionRpt_Ind_t, 736

- pVertUnc
 - unpack_loc_BestAvailPos_Ind_t, [717](#)
 - unpack_loc_PositionRpt_Ind_t, [736](#)
- pVerticalConfidence
 - pack_pds_SLQSPDSInjectPositionData_t, [440](#)
- pVerticalUnc
 - pack_pds_SLQSPDSInjectPositionData_t, [440](#)
- pVoIPConfigEvents
 - pack_ims_SLQSImsConfigIndicationRegister_t, [371](#)
- pVoiceDomainPref
 - pack_voice_SLQSVoiceGetConfig_t, [501](#)
- pVoiceDomainPrefStatus
 - unpack_voice_SLQSVoiceSetConfig_t, [960](#)
- pVoicePrivacy
 - unpack_voice_SLQSVoiceGetAllCallInfo_t, [939](#)
 - unpack_voice_SLQSVoiceGetCallInfo_t, [946](#)
- pVoltReport
 - pack_dms_SwiSetEventReport_t, [363](#)
- pVolume
 - pack_swiaudio_SLQSSetM2MAudioProfile_t, [460](#)
- pWAMSParaChanged
 - unpack_swiavms_SLQSAVMSEventReportInd_t, [883](#)
- pWCDMABER
 - unpack_nas_SLQSGetErrorRate_t, [763](#)
- pWCDMACallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [779](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [816](#)
- pWCDMACipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [779](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [816](#)
- pWCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [406](#)
- pWCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [406](#)
- pWCDMAECIOThreshList
 - nas_WCDMAECIOThresh, [316](#)
- pWCDMAInfoLTENeighborCell
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [791](#)
- pWCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [406](#)
- pWCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [406](#)
- pWCDMARSSIThreshList
 - nas_WCDMARSSIThresh, [318](#)
- pWCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, [803](#)
- pWCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, [780](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [816](#)
- pWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [780](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [816](#)
- pWcdmaUARFCN
 - pack_nas_SLQSNASSwiSetChannelLock_t, [414](#)
 - unpack_nas_SLQSNASSwiGetChannelLock_t, [803](#)
- pWifiState
 - pack_pds_SLQSSetPositionMethodState_t, [442](#)
- pXid
 - unpack_loc_BestAvailPos_Ind_t, [717](#)
- pXtraDataState
 - pack_pds_SLQSSetPositionMethodState_t, [442](#)
- pXtraTimeState
 - pack_pds_SLQSSetPositionMethodState_t, [442](#)
- pack_audio_SLQSGetAudioPathConfig
 - audio.h, [1106](#)
- pack_audio_SLQSGetAudioPathConfig_t, [342](#)
 - Item, [343](#)
 - Profile, [343](#)
- pack_audio_SLQSGetAudioProfile
 - audio.h, [1106](#)
- pack_audio_SLQSGetAudioProfile_t, [343](#)
 - Generator, [343](#)
- pack_audio_SLQSGetAudioVoITLBConfig
 - audio.h, [1107](#)
- pack_audio_SLQSGetAudioVoITLBConfig_t, [344](#)
 - Generator, [344](#)
 - Item, [344](#)
 - Profile, [344](#)
 - Volume, [344](#)
- pack_audio_SLQSSetAudioPathConfig
 - audio.h, [1107](#)
- pack_audio_SLQSSetAudioPathConfig_t, [344](#)
 - pCodecSTGain, [346](#)
 - pECMode, [346](#)
 - pNSEnable, [346](#)
 - pRXAGCList, [346](#)
 - pRXAVCList, [346](#)
 - pTXAGCList, [346](#)
 - pTXAVCSwitch, [346](#)
 - pTXGain, [346](#)
 - Profile, [346](#)
- pack_audio_SLQSSetAudioProfile
 - audio.h, [1108](#)
- pack_audio_SLQSSetAudioProfile_t, [346](#)
 - EarMute, [347](#)
 - Generator, [347](#)
 - MicMute, [347](#)
 - Profile, [347](#)
 - Volume, [347](#)
- pack_audio_SLQSSetAudioVoITLBConfig
 - audio.h, [1108](#)
- pack_audio_SLQSSetAudioVoITLBConfig_t, [348](#)
 - Generator, [348](#)
 - Item, [348](#)
 - Profile, [348](#)
 - VolValue, [348](#)
 - Volume, [348](#)
- pack_cat_CATSendEnvelopeCommand
 - cat.h, [1112](#)
- pack_cat_CATSendEnvelopeCommand_t, [348](#)
 - cmdID, [349](#)
 - dataLen, [349](#)
 - pData, [349](#)

[pack_cat_CATSendTerminalResponse](#)
[cat.h, 1112](#)
[pack_cat_CATSendTerminalResponse_t, 349](#)
[dataLen, 350](#)
[pData, 350](#)
[refID, 350](#)
[pack_cat_SetCATEventCallback](#)
[cat.h, 1112](#)
[pack_cat_SetCATEventCallback_t, 350](#)
[eventMask, 350](#)
[pack_dms_ActivateAutomatic](#)
[dms.h, 1130](#)
[pack_dms_ActivateAutomatic_t, 351](#)
[actCode, 351](#)
[pack_dms_GetActivationState](#)
[dms.h, 1130](#)
[pack_dms_GetBandCapability](#)
[dms.h, 1131](#)
[pack_dms_GetCrashAction](#)
[dms.h, 1131](#)
[pack_dms_GetCustFeature](#)
[dms.h, 1131](#)
[pack_dms_GetCustFeaturesV2](#)
[dms.h, 1132](#)
[pack_dms_GetCustFeaturesV2_t, 351](#)
[cust_id, 351](#)
[list_type, 351](#)
[Tlvresult, 351](#)
[pack_dms_GetDeviceCap](#)
[dms.h, 1132](#)
[pack_dms_GetDeviceCapabilities](#)
[dms.h, 1133](#)
[pack_dms_GetDeviceCapabilitiesV2](#)
[dms.h, 1133](#)
[pack_dms_GetDeviceHardwareRev](#)
[dms.h, 1133](#)
[pack_dms_GetDeviceMfr](#)
[dms.h, 1134](#)
[pack_dms_GetDeviceSerialNumbers](#)
[dms.h, 1134](#)
[pack_dms_GetFSN](#)
[dms.h, 1135](#)
[pack_dms_GetFirmwareInfo](#)
[dms.h, 1134](#)
[pack_dms_GetFirmwareRevision](#)
[dms.h, 1135](#)
[pack_dms_GetFirmwareRevisions](#)
[dms.h, 1135](#)
[pack_dms_GetHardwareRevision](#)
[dms.h, 1136](#)
[pack_dms_GetIMSI](#)
[dms.h, 1136](#)
[pack_dms_GetManufacturer](#)
[dms.h, 1136](#)
[pack_dms_GetModelID](#)
[dms.h, 1137](#)
[pack_dms_GetNetworkTime](#)
[dms.h, 1137](#)
[pack_dms_GetNetworkTimeV2](#)
[dms.h, 1137](#)
[pack_dms_GetOfflineReason](#)
[dms.h, 1138](#)
[pack_dms_GetPRLVersion](#)
[dms.h, 1138](#)
[pack_dms_GetPower](#)
[dms.h, 1138](#)
[pack_dms_GetSerialNumbers](#)
[dms.h, 1139](#)
[pack_dms_GetUSBComp](#)
[dms.h, 1139](#)
[pack_dms_GetVoiceNumber](#)
[dms.h, 1139](#)
[pack_dms_ResetToFactoryDefaults](#)
[dms.h, 1140](#)
[pack_dms_ResetToFactoryDefaults_t, 352](#)
[spc, 352](#)
[pack_dms_SLQSDmsSwiGetPCInfo](#)
[dms.h, 1144](#)
[pack_dms_SLQSDmsSwiGetResetInfo](#)
[dms.h, 1144](#)
[pack_dms_SLQSDmsSwiGetUimSelection](#)
[dms.h, 1144](#)
[pack_dms_SLQSDmsSwiIndicationRegister](#)
[dms.h, 1145](#)
[pack_dms_SLQSDmsSwiIndicationRegister_t, 359](#)
[resetInfoInd, 359](#)
[pack_dms_SLQSGetBandCapability](#)
[dms.h, 1145](#)
[pack_dms_SLQSGetERIFile](#)
[dms.h, 1145](#)
[pack_dms_SLQSGetPowerSaveModeConfig](#)
[dms.h, 1146](#)
[pack_dms_SLQSSetPowerSaveModeConfig](#)
[dms.h, 1146](#)
[pack_dms_SLQSSetPowerSaveModeConfig_t, 359](#)
[pActiveTimer, 360](#)
[pPeriodicUpdateTimer, 360](#)
[pPsmEnableState, 360](#)
[pack_dms_SLQSSwiClearDyingGaspStatistics](#)
[dms.h, 1146](#)
[pack_dms_SLQSSwiGetCrashInfo](#)
[dms.h, 1147](#)
[pack_dms_SLQSSwiGetCrashInfo_t, 360](#)
[clear, 360](#)
[pack_dms_SLQSSwiGetDyingGaspCfg](#)
[dms.h, 1147](#)
[pack_dms_SLQSSwiGetDyingGaspStatistics](#)
[dms.h, 1147](#)
[pack_dms_SLQSSwiGetFirmwareCurr](#)
[dms.h, 1148](#)
[pack_dms_SLQSSwiGetFwUpdateStatus](#)
[dms.h, 1148](#)
[pack_dms_SLQSSwiGetHostDevInfo](#)
[dms.h, 1148](#)
[pack_dms_SLQSSwiGetOSInfo](#)
[dms.h, 1149](#)

- pack_dms_SLQSSwiGetSerialNoExt
 - dms.h, [1149](#)
- pack_dms_SLQSSwiSetDyingGaspCfg
 - dms.h, [1149](#)
- pack_dms_SLQSSwiSetDyingGaspCfg_t, [360](#)
 - pDestSMSContent, [361](#)
 - pDestSMSNum, [361](#)
- pack_dms_SLQSSwiSetHostDevInfo
 - dms.h, [1150](#)
- pack_dms_SLQSSwiSetHostDevInfo_t, [361](#)
 - hostID, [362](#)
 - manString, [362](#)
 - modelString, [362](#)
 - plasmaIDString, [362](#)
 - swVerString, [362](#)
- pack_dms_SLQSSwiSetOSInfo
 - dms.h, [1150](#)
- pack_dms_SLQSSwiSetOSInfo_t, [362](#)
 - nameString, [362](#)
 - versionString, [362](#)
- pack_dms_SLQSUIMGetState
 - dms.h, [1151](#)
- pack_dms_SetActivationStatusCallback
 - dms.h, [1140](#)
- pack_dms_SetActivationStatusCallback_t, [352](#)
 - activationState, [352](#)
- pack_dms_SetCrashAction
 - dms.h, [1141](#)
- pack_dms_SetCrashAction_t, [352](#)
 - crashAction, [353](#)
- pack_dms_SetCustFeature
 - dms.h, [1141](#)
- pack_dms_SetCustFeature_t, [353](#)
 - DHCPRelayEnabled, [355](#)
 - DisableIMSI, [355](#)
 - GPSPMP, [355](#)
 - GPSSel, [355](#)
 - GpsEnable, [355](#)
 - IPFamSupport, [355](#)
 - IsVoiceEnabled, [355](#)
 - RMAutoConnect, [355](#)
 - SMSSupport, [355](#)
- pack_dms_SetCustFeaturesV2
 - dms.h, [1141](#)
- pack_dms_SetCustFeaturesV2_t, [355](#)
 - cust_id, [356](#)
 - cust_value, [356](#)
 - Tlvresult, [356](#)
 - value_length, [356](#)
- pack_dms_SetEventReport
 - dms.h, [1142](#)
- pack_dms_SetEventReport_t, [356](#)
 - mode, [356](#)
- pack_dms_SetFirmwarePreference
 - dms.h, [1142](#)
- pack_dms_SetIndicationRegister
 - dms.h, [1142](#)
- pack_dms_SetIndicationRegister_t, [356](#)
 - PSMCfgChangeInfo, [357](#)
 - PSMStatus, [357](#)
 - RptIMSCapability, [357](#)
- pack_dms_SetPower
 - dms.h, [1143](#)
- pack_dms_SetPower_t, [357](#)
 - mode, [358](#)
 - Tlvresult, [358](#)
- pack_dms_SetUSBComp
 - dms.h, [1143](#)
- pack_dms_SetUSBComp_t, [358](#)
 - Tlvresult, [358](#)
 - USBComp, [358](#)
- pack_dms_SwiSetEventReport
 - dms.h, [1151](#)
- pack_dms_SwiSetEventReport_t, [362](#)
 - pTempReport, [363](#)
 - pUIMStatusReport, [363](#)
 - pVoltReport, [363](#)
- pack_dms_SwiUimSelect
 - dms.h, [1151](#)
- pack_dms_SwiUimSelect_t, [363](#)
 - uim_select, [363](#)
- pack_dms_UIMChangePIN
 - dms.h, [1152](#)
- pack_dms_UIMChangePIN_t, [364](#)
 - id, [364](#)
 - newValue, [364](#)
 - oldValue, [364](#)
- pack_dms_UIMGetControlKeyStatus
 - dms.h, [1152](#)
- pack_dms_UIMGetControlKeyStatus_t, [364](#)
 - facility, [365](#)
- pack_dms_UIMGetICCID
 - dms.h, [1152](#)
- pack_dms_UIMGetICCID_t, [365](#)
 - ParamPresenceMask, [365](#)
 - Tlvresult, [365](#)
- pack_dms_UIMGetPINStatus
 - dms.h, [1153](#)
- pack_dms_UIMSetControlKeyProtection
 - dms.h, [1153](#)
- pack_dms_UIMSetControlKeyProtection_t, [365](#)
 - facility, [366](#)
 - facilityCk, [366](#)
 - facilityState, [366](#)
- pack_dms_UIMSetPINProtection
 - dms.h, [1154](#)
- pack_dms_UIMSetPINProtection_t, [366](#)
 - bEnable, [367](#)
 - id, [367](#)
 - value, [367](#)
- pack_dms_UIMUnblockControlKey
 - dms.h, [1154](#)
- pack_dms_UIMUnblockControlKey_t, [367](#)
 - facility, [367](#)
 - facilityCk, [367](#)
- pack_dms_UIMUnblockPIN

- dms.h, [1154](#)
- pack_dms_UIMUnblockPIN_t, [367](#)
 - id, [368](#)
 - newPin, [368](#)
 - pukValue, [368](#)
- pack_dms_UIMVerifyPIN
 - dms.h, [1155](#)
- pack_dms_UIMVerifyPIN_t, [368](#)
 - id, [368](#)
 - value, [368](#)
- pack_dms_ValidateSPC
 - dms.h, [1155](#)
- pack_fms_GetImagesPreference
 - fms.h, [1184](#)
- pack_fms_GetImagesPreference_t, [368](#)
 - Tlvresult, [369](#)
- pack_fms_GetStoredImages
 - fms.h, [1184](#)
- pack_fms_GetStoredImages_t, [369](#)
 - Tlvresult, [369](#)
- pack_fms_SetImagesPreference
 - fms.h, [1185](#)
- pack_fms_SetImagesPreference_t, [369](#)
 - bForceDownload, [370](#)
 - imageListSize, [370](#)
 - modemindex, [370](#)
 - pImageList, [370](#)
 - Tlvresult, [370](#)
- pack_ims_SLQSGetIMSSMSConfig
 - ims.h, [1188](#)
- pack_ims_SLQSGetIMSUserConfig
 - ims.h, [1188](#)
- pack_ims_SLQSGetIMSVoIPConfig
 - ims.h, [1189](#)
- pack_ims_SLQSGetRegMgrConfig
 - ims.h, [1189](#)
- pack_ims_SLQSGetSIPConfig
 - ims.h, [1189](#)
- pack_ims_SLQSImsConfigIndicationRegister
 - ims.h, [1190](#)
- pack_ims_SLQSImsConfigIndicationRegister_t, [370](#)
 - pRegMgrConfigEvents, [371](#)
 - pSIPConfigEvents, [371](#)
 - pSMSConfigEvents, [371](#)
 - pUserConfigEvents, [371](#)
 - pVoIPConfigEvents, [371](#)
- pack_ims_SLQSSetIMSSMSConfig
 - ims.h, [1190](#)
- pack_ims_SLQSSetIMSSMSConfig_t, [371](#)
- pack_ims_SLQSSetIMSUserConfig
 - ims.h, [1191](#)
- pack_ims_SLQSSetIMSUserConfig_t, [372](#)
 - pIMSDomain, [373](#)
- pack_ims_SLQSSetIMSVoIPConfig
 - ims.h, [1191](#)
- pack_ims_SLQSSetIMSVoIPConfig_t, [373](#)
 - pAmrMode, [375](#)
 - pRingingTimer, [375](#)
- pack_ims_SLQSSetRegMgrConfig
 - ims.h, [1192](#)
- pack_ims_SLQSSetRegMgrConfig_t, [375](#)
 - pCSCFPortName, [376](#)
 - pIMSTestMode, [376](#)
 - pPriCSCFPort, [376](#)
- pack_ims_SLQSSetSIPConfig
 - ims.h, [1192](#)
- pack_ims_SLQSSetSIPConfig_t, [376](#)
 - pSIPLocalPort, [377](#)
 - pSigCompEnabled, [376](#)
 - pSubscribeTimer, [377](#)
 - pTimerSIPReg, [377](#)
 - pTimerT1, [377](#)
 - pTimerT2, [377](#)
 - pTimerTf, [377](#)
- pack_imsa_SLQSGetIMSARegStatus
 - imsa.h, [1199](#)
- pack_imsa_SLQSGetIMSAServiceStatus
 - imsa.h, [1200](#)
- pack_imsa_SLQSRegisterIMSAIndication
 - imsa.h, [1200](#)
- pack_imsa_SLQSRegisterIMSAIndication_t, [377](#)
 - PdpStatusConfig, [378](#)
 - RatHandoverStatusConfig, [378](#)
 - RegStatusConfig, [378](#)
 - ServiceStatusConfig, [378](#)
- pack_loc_Delete_Assist_Data_t, [378](#)
 - pBdsSVInfo, [379](#)
 - pCellIdb, [379](#)
 - pClkInfo, [379](#)
 - pGnssData, [379](#)
 - pSVInfo, [379](#)
 - Tlvresult, [379](#)
- pack_loc_DeleteAssistData
 - loc.h, [1224](#)
- pack_loc_EventRegister
 - loc.h, [1224](#)
- pack_loc_EventRegister_t, [379](#)
 - eventRegister, [381](#)
 - Tlvresult, [381](#)
- pack_loc_SLQSLOCGetBestAvailPos
 - loc.h, [1225](#)
- pack_loc_SLQSLOCGetBestAvailPos_t, [382](#)
 - Tlvresult, [382](#)
 - xid, [382](#)
- pack_loc_SLQSLOCGetOpMode
 - loc.h, [1225](#)
- pack_loc_SLQSLOCGetServer
 - loc.h, [1226](#)
- pack_loc_SLQSLOCGetServer_t, [383](#)
 - has_serverAddrTypeMask, [383](#)
 - serverAddrTypeMask, [383](#)
 - serverType, [383](#)
- pack_loc_SLQSLOCInjectPosition
 - loc.h, [1226](#)
- pack_loc_SLQSLOCInjectPosition_t, [383](#)
 - altitudeSrcInfo, [387](#)

- altitudeWrtEllipsoid, [387](#)
- altitudeWrtMeanSeaLevel, [387](#)
- has_altitudeSrcInfo, [387](#)
- has_altitudeWrtEllipsoid, [387](#)
- has_horConfidence, [387](#)
- has_horReliability, [387](#)
- has_horUncCircular, [387](#)
- has_latitude, [387](#)
- has_longitude, [387](#)
- has_positionSrc, [387](#)
- has_rawHorConfidence, [387](#)
- has_rawHorUncCircular, [387](#)
- has_timestampAge, [387](#)
- has_timestampUtc, [387](#)
- has_vertConfidence, [387](#)
- has_vertReliability, [387](#)
- has_vertUnc, [388](#)
- horConfidence, [388](#)
- horReliability, [388](#)
- horUncCircular, [388](#)
- latitude, [388](#)
- longitude, [388](#)
- positionSrc, [388](#)
- rawHorConfidence, [388](#)
- rawHorUncCircular, [388](#)
- timestampAge, [388](#)
- timestampUtc, [388](#)
- vertConfidence, [388](#)
- vertReliability, [388](#)
- vertUnc, [388](#)
- pack_loc_SLQSLOCInjectSensorData
 - loc.h, [1227](#)
- pack_loc_SLQSLOCInjectSensorData_t, [388](#)
 - accelTemp, [389](#)
 - acceleroData, [389](#)
 - acceleroTimeSrc, [389](#)
 - gyroData, [389](#)
 - gyroTemp, [389](#)
 - gyroTimeSrc, [390](#)
 - has_accelTemp, [390](#)
 - has_acceleroTimeSrc, [390](#)
 - has_acceleroData, [390](#)
 - has_gyroData, [390](#)
 - has_gyroTemp, [390](#)
 - has_gyroTimeSrc, [390](#)
 - has_opaqueId, [390](#)
 - opaqueId, [390](#)
- pack_loc_SLQSLOCInjectUTCTime
 - loc.h, [1227](#)
- pack_loc_SLQSLOCInjectUTCTime_t, [390](#)
 - timeMsec, [390](#)
 - timeUncMsec, [390](#)
- pack_loc_SLQSLOCSetCradleMountConfig
 - loc.h, [1227](#)
- pack_loc_SLQSLOCSetCradleMountConfig_t, [390](#)
 - confidence, [391](#)
 - has_confidence, [391](#)
 - state, [391](#)
- pack_loc_SLQSLOCSetServer
 - loc.h, [1228](#)
- pack_loc_SLQSLOCSetServer_t, [391](#)
 - pIPv4Config, [392](#)
 - pIPv6Config, [392](#)
 - pURLAddr, [392](#)
 - serverType, [392](#)
- pack_loc_SetExtPowerState
 - loc.h, [1224](#)
- pack_loc_SetExtPowerState_t, [381](#)
 - extPowerState, [381](#)
 - Tlvresult, [381](#)
- pack_loc_SetOperationMode
 - loc.h, [1225](#)
- pack_loc_SetOperationMode_t, [381](#)
 - mode, [382](#)
 - Tlvresult, [382](#)
- pack_loc_Start
 - loc.h, [1228](#)
- pack_loc_Start_t, [392](#)
 - pApplicationInfo, [393](#)
 - pConfigAltitudeAssumed, [393](#)
 - pHorizontalAccuracyLvl, [393](#)
 - pIntermediateReportState, [393](#)
 - pMinIntervalTime, [393](#)
 - pRecurrenceType, [393](#)
 - SessionId, [394](#)
 - Tlvresult, [394](#)
- pack_loc_Stop
 - loc.h, [1229](#)
- pack_loc_Stop_t, [394](#)
 - SessionId, [394](#)
 - Tlvresult, [394](#)
- pack_nas_GetACCOLC
 - nas.h, [1255](#)
- pack_nas_GetANAAAuthenticationStatus
 - nas.h, [1256](#)
- pack_nas_GetCDMANetworkParameters
 - nas.h, [1256](#)
- pack_nas_GetHomeNetwork
 - nas.h, [1256](#)
- pack_nas_GetHomeNetwork3GPP2
 - nas.h, [1257](#)
- pack_nas_GetNetworkPreference
 - nas.h, [1257](#)
- pack_nas_GetRFInfo
 - nas.h, [1257](#)
- pack_nas_GetServingNetwork
 - nas.h, [1257](#)
- pack_nas_GetServingNetworkCapabilities
 - nas.h, [1258](#)
- pack_nas_GetSignalStrengths
 - nas.h, [1258](#)
- pack_nas_InitiateDomainAttach
 - nas.h, [1258](#)
- pack_nas_InitiateDomainAttach_t, [394](#)
 - action, [394](#)
- pack_nas_PerformNetworkScan

- nas.h, [1259](#)
- pack_nas_PerformNetworkScanPCI
 - nas.h, [1259](#)
- pack_nas_PerformNetworkScanPCI_t, [395](#)
 - pCiotOpModePref, [395](#)
 - pLteM1BandPref, [395](#)
 - pLteNB1BandPref, [395](#)
 - pNetworkType, [395](#)
 - pScanType, [395](#)
- pack_nas_SLQSConfigSigInfo
 - nas.h, [1261](#)
- pack_nas_SLQSConfigSigInfo_t, [398](#)
 - pECIOThresh, [399](#)
 - pIOTThresh, [399](#)
 - pLTESNRThresh, [399](#)
 - pLTESigRptCfg, [399](#)
 - pRSRPThresh, [399](#)
 - pRSRQThresh, [399](#)
 - pRSSIThresh, [399](#)
- pack_nas_SLQSGetErrorRate
 - nas.h, [1262](#)
- pack_nas_SLQSGetNetworkTime
 - nas.h, [1262](#)
- pack_nas_SLQSGetOperatorNameData
 - nas.h, [1263](#)
- pack_nas_SLQSGetPLMNName
 - nas.h, [1263](#)
- pack_nas_SLQSGetPLMNName_t, [399](#)
 - mcc, [400](#)
 - mnc, [400](#)
 - pMncPcsStatus, [400](#)
- pack_nas_SLQSGetServingSystem
 - nas.h, [1263](#)
- pack_nas_SLQSGetSignalStrength
 - nas.h, [1264](#)
- pack_nas_SLQSGetSysInfo
 - nas.h, [1264](#)
- pack_nas_SLQSGetSysSelectionPref
 - nas.h, [1264](#)
- pack_nas_SLQSGetSysSelectionPrefExt
 - nas.h, [1265](#)
- pack_nas_SLQSInitiateNetworkRegistration
 - nas.h, [1265](#)
- pack_nas_SLQSInitiateNetworkRegistration_t, [400](#)
 - pChangeDuration, [401](#)
 - pMNRInfo, [401](#)
 - pMncPcsDigitStatus, [401](#)
 - regAction, [401](#)
- pack_nas_SLQSNASGetForbiddenNetworks
 - nas.h, [1267](#)
- pack_nas_SLQSNASGetDRXParams
 - nas.h, [1267](#)
- pack_nas_SLQSNASGetDRXParamsExt
 - nas.h, [1267](#)
- pack_nas_SLQSNASGetDRXParamsExt_t, [406](#)
- pack_nas_SLQSNASSetDRXParams
 - nas.h, [1269](#)
- pack_nas_SLQSNASSetDRXParams_t, [411](#)
 - pCycleLen, [412](#)
 - pEdrxEnable, [412](#)
- pack_nas_SLQSNASSwiGetChannelLock
 - nas.h, [1270](#)
- pack_nas_SLQSNASSwiSetChannelLock
 - nas.h, [1271](#)
- pack_nas_SLQSNASSwiSetChannelLock_t, [414](#)
- pack_nas_SLQSNasConfigSigInfo2
 - nas.h, [1265](#)
- pack_nas_SLQSNasConfigSigInfo2_t, [401](#)
 - pHDRIODelta, [405](#)
 - pHDRIOTresh, [405](#)
 - pLTESigRptConfig, [405](#)
- pack_nas_SLQSNasGet3GPP2Subscription
 - nas.h, [1266](#)
- pack_nas_SLQSNasGet3GPP2Subscription_t, [406](#)
 - namID, [406](#)
- pack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [1266](#)
- pack_nas_SLQSNasGetHDRColorCode
 - nas.h, [1268](#)
- pack_nas_SLQSNasGetRFInfo
 - nas.h, [1268](#)
- pack_nas_SLQSNasGetSigInfo
 - nas.h, [1268](#)
- pack_nas_SLQSNasGetTxRxInfo
 - nas.h, [1269](#)
- pack_nas_SLQSNasGetTxRxInfo_t, [407](#)
 - radio_if, [408](#)
- pack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [1269](#)
- pack_nas_SLQSNasIndicationRegisterExt_t, [408](#)
 - pDDTMInd, [410](#)
 - pDualStandByPrefInd, [410](#)
 - pEdrxChangeInfoInd, [410](#)
 - pErrorRateInd, [410](#)
 - pHDRSessionCloseInd, [410](#)
 - pLTECphyCa, [410](#)
 - pManagedRoamingInd, [410](#)
 - pNetworkRejectInd, [411](#)
 - pNetworkTimeInd, [411](#)
 - pServingSystemInd, [411](#)
 - pSignalStrengthInd, [411](#)
 - pSubscriptionInfoInd, [411](#)
 - pSuppressSysInfoInd, [411](#)
 - pSysInfoInd, [411](#)
 - pSystemSelectionInd, [411](#)
- pack_nas_SLQSNasSwiIndicationRegister
 - nas.h, [1270](#)
- pack_nas_SLQSNasSwiIndicationRegister_t, [412](#)
 - gsmUmtsDI, [413](#)
 - gsmUmtsUI, [413](#)
 - lteEmmDI, [413](#)
 - lteEmmUI, [413](#)
 - lteEsmDI, [413](#)
 - lteEsmUI, [413](#)
 - pRankIndicatorInd, [414](#)
 - pTimer, [414](#)

- pack_nas_SLQSNasSwiModemStatus
 - nas.h, [1271](#)
- pack_nas_SLQSSetBandPreference
 - nas.h, [1271](#)
- pack_nas_SLQSSetBandPreference_t, [414](#)
 - bandPref, [415](#)
- pack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [1272](#)
- pack_nas_SLQSSetSignalStrengthsCallback_t, [416](#)
 - bEnable, [416](#)
 - pSigIndReq, [416](#)
- pack_nas_SLQSSetSysSelectionPref
 - nas.h, [1272](#)
- pack_nas_SLQSSetSysSelectionPref_t, [416](#)
 - pAcqOrderPref, [421](#)
 - pBandPref, [421](#)
 - pCSGID, [421](#)
 - pChgDuration, [421](#)
 - pEmerMode, [421](#)
 - pGWAcqOrderPref, [421](#)
 - pLTEBandPref, [421](#)
 - pModePref, [421](#)
 - pNetSelPref, [421](#)
 - pPRLPref, [421](#)
 - pRAT, [421](#)
 - pRoamPref, [421](#)
 - pSrvDomainPref, [421](#)
 - pSrvRegRestriction, [421](#)
 - pTdsdmaBandPref, [422](#)
- pack_nas_SLQSSetSysSelectionPrefExt
 - nas.h, [1272](#)
- pack_nas_SLQSSetSysSelectionPrefExt_t, [422](#)
 - pAcqOrderPref, [431](#)
 - pBandPref, [431](#)
 - pCSGID, [431](#)
 - pChgDuration, [431](#)
 - pCiotAcqOrderPref, [431](#)
 - pCiotLteOpMode, [431](#)
 - pCiotLteOpModePref, [431](#)
 - pEmerMode, [431](#)
 - pGWAcqOrderPref, [431](#)
 - pLTEBandPref, [431](#)
 - pLteM1BandPref, [431](#)
 - pLteNb1BandPref, [431](#)
 - pModePref, [431](#)
 - pNetSelPref, [431](#)
 - pNr5gBandPref, [431](#)
 - pPRLPref, [431](#)
 - pRAT, [432](#)
 - pRoamPref, [432](#)
 - pSrvDomainPref, [432](#)
 - pSrvRegRestriction, [432](#)
 - pTdsdmaBandPref, [432](#)
- pack_nas_SLQSSwiGetHDRPersonality
 - nas.h, [1273](#)
- pack_nas_SLQSSwiGetHDRProtSubtype
 - nas.h, [1273](#)
- pack_nas_SLQSSwiGetHRPDStats
 - nas.h, [1273](#)
- pack_nas_SLQSSwiGetLteCQI
 - nas.h, [1274](#)
- pack_nas_SLQSSwiGetLteSccRxInfo
 - nas.h, [1274](#)
- pack_nas_SLQSSwiNetworkDebug
 - nas.h, [1274](#)
- pack_nas_SLQSSwiPSDetach
 - nas.h, [1275](#)
- pack_nas_SLQSSwiPSDetach_t, [432](#)
 - pDetachAction, [432](#)
- pack_nas_SetACCOLC
 - nas.h, [1260](#)
- pack_nas_SetACCOLC_t, [395](#)
 - accolc, [395](#)
 - spc, [395](#)
- pack_nas_SetCDMANetworkParameters
 - nas.h, [1260](#)
- pack_nas_SetCDMANetworkParameters_t, [395](#)
 - pApplication, [397](#)
 - pBroadcast, [397](#)
 - pCustomSCP, [397](#)
 - pForceRev0, [397](#)
 - pProtocol, [397](#)
 - pRoaming, [397](#)
 - pSPC, [397](#)
- pack_nas_SetLURejectCallback
 - nas.h, [1260](#)
- pack_nas_SetNetworkPreference
 - nas.h, [1261](#)
- pack_nas_SetNetworkPreference_t, [397](#)
 - Duration, [398](#)
 - TechnologyPref, [398](#)
 - Tlvresult, [398](#)
- pack_nas_SetRFInfoCallback
 - nas.h, [1261](#)
- pack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [1262](#)
- pack_pds_ForceXTRADownload
 - pds.h, [1306](#)
- pack_pds_GetPDSDefaults
 - pds.h, [1306](#)
- pack_pds_GetPDSSState
 - pds.h, [1307](#)
- pack_pds_GetPortAutomaticTracking
 - pds.h, [1307](#)
- pack_pds_GetServiceAutomaticTracking
 - pds.h, [1307](#)
- pack_pds_GetXTRAAutomaticDownload
 - pds.h, [1308](#)
- pack_pds_GetXTRANetwork
 - pds.h, [1308](#)
- pack_pds_GetXTRAValidity
 - pds.h, [1308](#)
- pack_pds_PDSInjectTimeReference
 - pds.h, [1309](#)
- pack_pds_PDSInjectTimeReference_t, [432](#)
 - systemDiscontinuities, [433](#)

- systemTime, [433](#)
- pack_pds_ResetPDSDData
 - pds.h, [1309](#)
- pack_pds_ResetPDSDData_t, [433](#)
 - pCellDataMask, [433](#)
 - pGPSDataMask, [434](#)
- pack_pds_SLQSGetAGPSConfig
 - pds.h, [1312](#)
- pack_pds_SLQSGetAGPSConfig_t, [437](#)
 - pNetworkMode, [438](#)
- pack_pds_SLQSGetGPSStateInfo
 - pds.h, [1312](#)
- pack_pds_SLQSPDSDeterminePosition
 - pds.h, [1313](#)
- pack_pds_SLQSPDSInjectAbsoluteTimeReference
 - pds.h, [1313](#)
- pack_pds_SLQSPDSInjectAbsoluteTimeReference_t,
 - [438](#)
 - forceFlag, [438](#)
 - timeBase, [438](#)
 - timeMsec, [438](#)
 - timeUncMsec, [439](#)
- pack_pds_SLQSPDSInjectPositionData
 - pds.h, [1313](#)
- pack_pds_SLQSPDSInjectPositionData_t, [439](#)
 - pAltitudeWrtEllipsoid, [440](#)
 - pAltitudeWrtSealevel, [440](#)
 - pHorizontalConfidence, [440](#)
 - pHorizontalUncCircular, [440](#)
 - pLatitude, [440](#)
 - pLongitude, [440](#)
 - pPositionSource, [440](#)
 - pTimeStamp, [440](#)
 - pTimeType, [440](#)
 - pVerticalConfidence, [440](#)
 - pVerticalUnc, [440](#)
- pack_pds_SLQSSetAGPSConfig
 - pds.h, [1314](#)
- pack_pds_SLQSSetAGPSConfig_t, [441](#)
 - pNetworkMode, [441](#)
 - pServerAddress, [441](#)
 - pServerPort, [441](#)
 - pServerURL, [441](#)
- pack_pds_SLQSSetPositionMethodState
 - pds.h, [1314](#)
- pack_pds_SLQSSetPositionMethodState_t, [441](#)
 - pWifiState, [442](#)
 - pXtraDataState, [442](#)
 - pXtraTimeState, [442](#)
- pack_pds_SetEventReportCallback
 - pds.h, [1309](#)
- pack_pds_SetEventReportCallback_t, [434](#)
 - posDataNmea, [434](#)
 - rptPosData, [434](#)
- pack_pds_SetPDSDDefaults
 - pds.h, [1310](#)
- pack_pds_SetPDSDDefaults_t, [434](#)
 - accuracy, [435](#)
 - interval, [435](#)
 - operation, [435](#)
 - timeout, [435](#)
- pack_pds_SetPDSSState
 - pds.h, [1310](#)
- pack_pds_SetPDSSState_t, [435](#)
 - enable, [435](#)
- pack_pds_SetPortAutomaticTracking
 - pds.h, [1310](#)
- pack_pds_SetPortAutomaticTracking_t, [435](#)
 - bAuto, [436](#)
- pack_pds_SetServiceAutomaticTracking
 - pds.h, [1311](#)
- pack_pds_SetServiceAutomaticTracking_t, [436](#)
 - bAuto, [436](#)
- pack_pds_SetXTRAAutomaticDownload
 - pds.h, [1311](#)
- pack_pds_SetXTRAAutomaticDownload_t, [436](#)
 - bEnabled, [437](#)
 - interval, [437](#)
- pack_pds_SetXTRANetwork
 - pds.h, [1311](#)
- pack_pds_SetXTRANetwork_t, [437](#)
 - preference, [437](#)
- pack_pds_StartPDSTrackingSessionExt
 - pds.h, [1314](#)
- pack_pds_StartPDSTrackingSessionExt_t, [442](#)
 - fixAccuracy, [443](#)
 - fixCount, [443](#)
 - fixInterval, [443](#)
 - fixTimeout, [443](#)
 - sessionControl, [443](#)
 - sessionOperation, [443](#)
 - sessionServerOption, [443](#)
 - sessionType, [443](#)
- pack_pds_StopPDSTrackingSession
 - pds.h, [1315](#)
- pack_qmi_t, [443](#)
 - msgid, [444](#)
 - svc, [444](#)
 - timeout, [444](#)
 - xid, [444](#)
- pack_qos_BindDataPort
 - qos.h, [1374](#)
- pack_qos_BindDataPort_t, [444](#)
 - pMuxID, [445](#)
 - pPeripheralEndPointID, [445](#)
 - pSIODDataPort, [445](#)
- pack_qos_SLQSQosGetNetworkStatus
 - qos.h, [1374](#)
- pack_qos_SLQSQosSwiReadApnExtraParams
 - qos.h, [1375](#)
- pack_qos_SLQSQosSwiReadApnExtraParams_t, [445](#)
 - apnId, [445](#)
- pack_qos_SLQSQosSwiReadDataStats
 - qos.h, [1376](#)
- pack_qos_SLQSQosSwiReadDataStats_t, [445](#)
 - apnId, [445](#)

- pack_qos_SLQSSetQosEventCallback
 - qos.h, [1376](#)
- pack_qos_SLQSSetQosEventCallback_t, [445](#)
 - enable, [446](#)
- pack_rms_GetSMSWake
 - rms.h, [1382](#)
- pack_rms_SetSMSWake
 - rms.h, [1383](#)
- pack_rms_SetSMSWake_t, [446](#)
 - enabled, [446](#)
 - wake_mask, [446](#)
- pack_sar_SLQSGetRfSarState
 - sar.h, [1384](#)
- pack_sar_SLQSSetRfSarState
 - sar.h, [1384](#)
- pack_sar_SLQSSetRfSarState_t, [446](#)
 - RfSarState, [447](#)
- pack_sms_GetSMSCAddress
 - sms.h, [1390](#)
- pack_sms_SLQSDeleteSMS
 - sms.h, [1392](#)
- pack_sms_SLQSDeleteSMS_t, [449](#)
 - pMessageIndex, [450](#)
 - pMessageMode, [450](#)
 - pMessageTag, [450](#)
 - storageType, [450](#)
- pack_sms_SLQSGetIndicationRegister
 - sms.h, [1392](#)
- pack_sms_SLQSGetMessageWaiting
 - sms.h, [1392](#)
- pack_sms_SLQSGetSMS
 - sms.h, [1393](#)
- pack_sms_SLQSGetSMS_t, [450](#)
 - messageIndex, [451](#)
 - pMessageMode, [451](#)
 - storageType, [451](#)
- pack_sms_SLQSGetSMSList
 - sms.h, [1393](#)
- pack_sms_SLQSGetSMSList_t, [451](#)
 - pMessageMode, [452](#)
 - pRequestedTag, [452](#)
 - storageType, [452](#)
- pack_sms_SLQSGetSmsBroadcastConfig
 - sms.h, [1393](#)
- pack_sms_SLQSGetSmsBroadcastConfig_t, [451](#)
 - mode, [451](#)
- pack_sms_SLQSGetTransLayerInfo
 - sms.h, [1394](#)
- pack_sms_SLQSGetTransNWRegInfo
 - sms.h, [1394](#)
- pack_sms_SLQSModifySMSStatus
 - sms.h, [1394](#)
- pack_sms_SLQSModifySMSStatus_t, [452](#)
 - messageIndex, [453](#)
 - messageTag, [453](#)
 - pMessageMode, [453](#)
 - storageType, [453](#)
- pack_sms_SLQSSendAsyncSMS
 - sms.h, [1395](#)
- pack_sms_SLQSSendAsyncSMS_t, [453](#)
 - pSendSmsParams, [453](#)
- pack_sms_SLQSSetIndicationRegister
 - sms.h, [1395](#)
- pack_sms_SLQSSetIndicationRegister_t, [453](#)
 - pSetIndicationRegReq, [454](#)
- pack_sms_SLQSSetSmsBroadcastActivation
 - sms.h, [1395](#)
- pack_sms_SLQSSetSmsBroadcastActivation_t, [454](#)
 - broadcastActivate, [454](#)
 - mode, [454](#)
- pack_sms_SLQSSetSmsBroadcastConfig
 - sms.h, [1396](#)
- pack_sms_SLQSSetSmsBroadcastConfig_t, [454](#)
 - mode, [455](#)
 - pBroadcastConfig, [455](#)
 - pCDMABroadcastConfig, [455](#)
- pack_sms_SLQSSetSmsStorage
 - sms.h, [1396](#)
- pack_sms_SLQSSetSmsStorage_t, [455](#)
 - smsStorage, [455](#)
- pack_sms_SLQSSmsGetMaxStorageSize
 - sms.h, [1397](#)
- pack_sms_SLQSSmsGetMaxStorageSize_t, [456](#)
 - pMaxStorageSizeReq, [456](#)
- pack_sms_SLQSSmsGetMessageProtocol
 - sms.h, [1397](#)
- pack_sms_SLQSSmsSetRoutes
 - sms.h, [1397](#)
- pack_sms_SLQSSmsSetRoutes_t, [456](#)
 - pSetRoutesReq, [456](#)
- pack_sms_SLQSSwiGetSMSStorage
 - sms.h, [1398](#)
- pack_sms_SaveSMS
 - sms.h, [1390](#)
- pack_sms_SaveSMS_t, [447](#)
 - messageFormat, [448](#)
 - messageSize, [448](#)
 - pMessage, [448](#)
 - storageType, [448](#)
- pack_sms_SendSMS
 - sms.h, [1391](#)
- pack_sms_SendSMS_t, [448](#)
 - messageFormat, [448](#)
 - messageSize, [448](#)
 - pLinktimer, [448](#)
 - pMessage, [449](#)
- pack_sms_SetNewSMSCallback
 - sms.h, [1391](#)
- pack_sms_SetNewSMSCallback_t, [449](#)
 - status, [449](#)
- pack_sms_SetSMSCAddress
 - sms.h, [1391](#)
- pack_sms_SetSMSCAddress_t, [449](#)
 - pSMSCAddress, [449](#)
 - pSMSCType, [449](#)
- pack_swiaudio_SLQSGetM2MAVMute

- swiaudio.h, [1411](#)
- pack_swiaudio_SLQSGetM2MAVMute_t, [457](#)
 - Profile, [458](#)
- pack_swiaudio_SLQSGetM2MAudioProfile
 - swiaudio.h, [1410](#)
- pack_swiaudio_SLQSGetM2MAudioProfile_t, [456](#)
 - pGenerator, [457](#)
- pack_swiaudio_SLQSGetM2MAudioVolume
 - swiaudio.h, [1410](#)
- pack_swiaudio_SLQSGetM2MAudioVolume_t, [457](#)
 - Generator, [457](#)
 - Profile, [457](#)
- pack_swiaudio_SLQSGetM2MSprGain
 - swiaudio.h, [1411](#)
- pack_swiaudio_SLQSGetM2MSprGain_t, [458](#)
 - Profile, [458](#)
- pack_swiaudio_SLQSSetM2MAVMute
 - swiaudio.h, [1413](#)
- pack_swiaudio_SLQSSetM2MAVMute_t, [461](#)
 - EarMute, [462](#)
 - MicMute, [462](#)
 - pCwtMute, [462](#)
 - Profile, [462](#)
- pack_swiaudio_SLQSSetM2MAudioAVCFG
 - swiaudio.h, [1412](#)
- pack_swiaudio_SLQSSetM2MAudioAVCFG_t, [458](#)
 - Device, [459](#)
 - Profile, [459](#)
- pack_swiaudio_SLQSSetM2MAudioLPBK
 - swiaudio.h, [1412](#)
- pack_swiaudio_SLQSSetM2MAudioLPBK_t, [459](#)
 - Enable, [459](#)
- pack_swiaudio_SLQSSetM2MAudioNVDef
 - swiaudio.h, [1412](#)
- pack_swiaudio_SLQSSetM2MAudioProfile
 - swiaudio.h, [1413](#)
- pack_swiaudio_SLQSSetM2MAudioProfile_t, [459](#)
 - pCwtMute, [460](#)
 - pEarMute, [460](#)
 - pGenerator, [460](#)
 - pMicMute, [460](#)
 - pVolume, [460](#)
 - Profile, [460](#)
- pack_swiaudio_SLQSSetM2MAudioVolume
 - swiaudio.h, [1413](#)
- pack_swiaudio_SLQSSetM2MAudioVolume_t, [461](#)
 - Generator, [461](#)
 - Level, [461](#)
 - Profile, [461](#)
- pack_swiaudio_SLQSSetM2MSprGain
 - swiaudio.h, [1414](#)
- pack_swiaudio_SLQSSetM2MSprGain_t, [462](#)
 - Profile, [462](#)
 - Value, [462](#)
- pack_swiaavms_SLQSAVMSGetSettings
 - swiaavms.h, [1420](#)
- pack_swiaavms_SLQSAVMSGetSettings_v2
 - swiaavms.h, [1421](#)
- pack_swiaavms_SLQSAVMSSendSelection
 - swiaavms.h, [1422](#)
- pack_swiaavms_SLQSAVMSSendSelection_t, [463](#)
 - pDeferTime, [463](#)
 - pPackageID, [463](#)
 - pRejectReason, [463](#)
 - selection, [463](#)
- pack_swiaavms_SLQSAVMSSessionGetInfo
 - swiaavms.h, [1422](#)
- pack_swiaavms_SLQSAVMSSetSettings
 - swiaavms.h, [1423](#)
- pack_swiaavms_SLQSAVMSSetSettings_t, [463](#)
 - AutoConnect, [465](#)
 - AutoReboot, [465](#)
 - pAPNInfo, [465](#)
 - pConnectionRetryTimers, [465](#)
 - pNotifStore, [465](#)
 - pPeriodInfo, [465](#)
 - pPollingTimer, [465](#)
 - PromptFwDownload, [465](#)
 - PromptFwUpdate, [465](#)
- pack_swiaavms_SLQSAVMSSetSettings_v2
 - swiaavms.h, [1424](#)
- pack_swiaavms_SLQSAVMSSetSettings_v2_t, [465](#)
 - AutoConnect, [466](#)
 - pAutoReboot, [467](#)
 - pNotifStore, [467](#)
 - pPeriodInfo, [467](#)
 - pPollingTimer, [467](#)
 - PromptFwDownload, [467](#)
 - PromptFwUpdate, [467](#)
- pack_swiaavms_SLQSAVMSSetSettingsNoAutoReboot-Field
 - swiaavms.h, [1425](#)
- pack_swiaavms_SLQSAVMSStartSession
 - swiaavms.h, [1425](#)
- pack_swiaavms_SLQSAVMSStartSession_t, [467](#)
 - sessionType, [467](#)
- pack_swiaavms_SLQSAVMSStopSession
 - swiaavms.h, [1426](#)
- pack_swiaavms_SLQSAVMSStopSession_t, [467](#)
 - sessionType, [468](#)
- pack_swiaavms_SLQSAVmsSetEventReport
 - swiaavms.h, [1423](#)
- pack_swidms_SLQSSwiDmsGetHWWatchdog
 - swidms.h, [1435](#)
- pack_swidms_SLQSSwiDmsGetMTU
 - swidms.h, [1435](#)
- pack_swidms_SLQSSwiDmsGetSecureInfo
 - swidms.h, [1435](#)
- pack_swidms_SLQSSwiDmsGetUsbComp
 - swidms.h, [1436](#)
- pack_swidms_SLQSSwiDmsGetUsbNetNum
 - swidms.h, [1436](#)
- pack_swidms_SLQSSwiDmsSetHWWatchdog
 - swidms.h, [1436](#)
- pack_swidms_SLQSSwiDmsSetHWWatchdog_t, [468](#)
 - enable, [468](#)

- resetDelay, [468](#)
- timeout, [468](#)
- pack_swidms_SLQSSwiDmsSetMTU
 - swidms.h, [1437](#)
- pack_swidms_SLQSSwiDmsSetMTU_t, [468](#)
 - MTUSize, [469](#)
- pack_swidms_SLQSSwiDmsSetUsbComp
 - swidms.h, [1437](#)
- pack_swidms_SLQSSwiDmsSetUsbComp_t, [469](#)
 - CfgValue, [470](#)
- pack_swidms_SLQSSwiDmsSetUsbNetNum
 - swidms.h, [1437](#)
- pack_swidms_SLQSSwiDmsSetUsbNetNum_t, [470](#)
 - nUsbNetNum, [470](#)
- pack_swiloc_SwiLocGetAutoStart
 - swiloc.h, [1442](#)
- pack_swiloc_SwiLocSetAutoStart
 - swiloc.h, [1442](#)
- pack_swiloc_SwiLocSetAutoStart_t, [471](#)
 - fix_rate, [472](#)
 - fix_type, [472](#)
 - function, [472](#)
 - max_dist, [472](#)
 - max_time, [472](#)
 - set_fix_rate, [472](#)
 - set_fix_type, [472](#)
 - set_function, [472](#)
 - set_max_dist, [472](#)
 - set_max_time, [472](#)
- pack_swisma_SLQSOMADMAAlertCallback
 - swisma.h, [1444](#)
- pack_swisma_SLQSOMADMCancelSession
 - swisma.h, [1445](#)
- pack_swisma_SLQSOMADMCancelSession_t, [472](#)
 - sessionType, [473](#)
- pack_swisma_SLQSOMADMCancelSessionExt
 - swismaext.h, [1454](#)
- pack_swisma_SLQSOMADMCancelSessionExt_t, [473](#)
 - sessionType, [473](#)
- pack_swisma_SLQSOMADMGetSessionInfo
 - swisma.h, [1446](#)
- pack_swisma_SLQSOMADMGetSessionInfo_t, [473](#)
 - SessionType, [474](#)
- pack_swisma_SLQSOMADMGetSessionInfoExt
 - swismaext.h, [1454](#)
- pack_swisma_SLQSOMADMGetSettings
 - swisma.h, [1446](#)
- pack_swisma_SLQSOMADMSendSelection
 - swisma.h, [1447](#)
- pack_swisma_SLQSOMADMSendSelection_t, [474](#)
 - pDeferTime, [474](#)
 - pRejectReason, [474](#)
 - selection, [474](#)
- pack_swisma_SLQSOMADMSendSelectionExt
 - swismaext.h, [1455](#)
- pack_swisma_SLQSOMADMSendSelectionExt_t, [474](#)
 - selection, [475](#)
- pack_swisma_SLQSOMADMSetSettings
 - swisma.h, [1447](#)
- pack_swisma_SLQSOMADMSetSettings_t, [475](#)
 - FOTAdownload, [476](#)
 - pAutosdm, [476](#)
 - pFwAutoCheck, [476](#)
- pack_swisma_SLQSOMADMSetSettingsExt
 - swismaext.h, [1455](#)
- pack_swisma_SLQSOMADMSetSettingsExt_t, [476](#)
- pack_swisma_SLQSOMADMStartSession
 - swisma.h, [1448](#)
- pack_swisma_SLQSOMADMStartSession_t, [477](#)
 - sessionType, [477](#)
- pack_swisma_SLQSOMADMStartSessionExt
 - swismaext.h, [1456](#)
- pack_swisma_SLQSOMADMStartSessionExt_t, [477](#)
 - sessionType, [478](#)
- pack_tmd_SLQSTmdDeRegNotMitigationLvl
 - tmd.h, [1461](#)
- pack_tmd_SLQSTmdDeRegNotMitigationLvl_t, [478](#)
 - mitigationDevID, [478](#)
 - mitigationDevIDLen, [478](#)
- pack_tmd_SLQSTmdGetMitigationDevList
 - tmd.h, [1462](#)
- pack_tmd_SLQSTmdGetMitigationLvl
 - tmd.h, [1462](#)
- pack_tmd_SLQSTmdGetMitigationLvl_t, [478](#)
 - mitigationDevID, [479](#)
 - mitigationDevIDLen, [479](#)
- pack_tmd_SLQSTmdRegNotMitigationLvl
 - tmd.h, [1463](#)
- pack_tmd_SLQSTmdRegNotMitigationLvl_t, [479](#)
 - mitigationDevID, [479](#)
 - mitigationDevIDLen, [479](#)
- pack_uim_ChangePin
 - uim.h, [1469](#)
- pack_uim_ChangePin_t, [479](#)
 - changePIN, [480](#)
 - EncryptedPIN1, [480](#)
 - pIndicationToken, [480](#)
 - pKeyReferenceID, [480](#)
 - sessionInfo, [480](#)
 - Tlvresult, [480](#)
- pack_uim_GetCardStatus
 - uim.h, [1469](#)
- pack_uim_ReadTransparent
 - uim.h, [1470](#)
- pack_uim_ReadTransparent_t, [480](#)
 - fileIndex, [481](#)
 - pEncryptData, [481](#)
 - pIndicationToken, [481](#)
 - readTransparent, [481](#)
 - sessionInfo, [481](#)
 - Tlvresult, [481](#)
- pack_uim_SLQSUIMAuthenticate
 - uim.h, [1470](#)
- pack_uim_SLQSUIMAuthenticate_t, [483](#)
 - authData, [483](#)
 - pIndicationToken, [483](#)

- sessionInfo, [483](#)
- pack_uim_SLQSUIMDepersonalization
 - uim.h, [1471](#)
- pack_uim_SLQSUIMDepersonalization_t, [483](#)
 - depersonilisationInfo, [484](#)
- pack_uim_SLQSUIMEventRegister
 - uim.h, [1471](#)
- pack_uim_SLQSUIMEventRegister_t, [484](#)
 - eventMask, [484](#)
- pack_uim_SLQSUIMGetConfiguration
 - uim.h, [1471](#)
- pack_uim_SLQSUIMGetConfiguration_t, [484](#)
 - pConfigurationMask, [484](#)
- pack_uim_SLQSUIMGetFileAttributes
 - uim.h, [1472](#)
- pack_uim_SLQSUIMGetFileAttributes_t, [485](#)
 - fileIndex, [485](#)
 - pIndicationToken, [485](#)
 - sessionInfo, [485](#)
- pack_uim_SLQSUIMGetSlotsStatus
 - uim.h, [1472](#)
- pack_uim_SLQSUIMPowerDown
 - uim.h, [1473](#)
- pack_uim_SLQSUIMPowerDown_t, [485](#)
 - slot, [486](#)
- pack_uim_SLQSUIMPowerUp
 - uim.h, [1473](#)
- pack_uim_SLQSUIMPowerUp_t, [486](#)
 - pIgnoreHotSwapSwitch, [486](#)
 - slot, [486](#)
- pack_uim_SLQSUIMRefreshComplete
 - uim.h, [1473](#)
- pack_uim_SLQSUIMRefreshComplete_t, [486](#)
 - refreshComplete, [487](#)
 - sessionInfo, [487](#)
- pack_uim_SLQSUIMRefreshGetLastEvent
 - uim.h, [1474](#)
- pack_uim_SLQSUIMRefreshGetLastEvent_t, [487](#)
 - sessionInfo, [487](#)
- pack_uim_SLQSUIMRefreshOK
 - uim.h, [1474](#)
- pack_uim_SLQSUIMRefreshOK_t, [487](#)
 - OKtoRefresh, [488](#)
 - sessionInfo, [488](#)
- pack_uim_SLQSUIMRefreshRegister
 - uim.h, [1474](#)
- pack_uim_SLQSUIMRefreshRegister_t, [488](#)
 - regRefresh, [488](#)
 - sessionInfo, [488](#)
- pack_uim_SLQSUIMReset
 - uim.h, [1475](#)
- pack_uim_SLQSUIMSwitchSlot
 - uim.h, [1475](#)
- pack_uim_SLQSUIMSwitchSlot_t, [488](#)
 - bLogicalSlot, [489](#)
 - ulPhysicalSlot, [489](#)
- pack_uim_SetPinProtection
 - uim.h, [1470](#)
- pack_uim_SetPinProtection_t, [482](#)
 - EncryptedPIN1, [482](#)
 - pIndicationToken, [482](#)
 - pKeyReferenceID, [482](#)
 - pinProtection, [482](#)
 - sessionInfo, [482](#)
 - Tlvresult, [482](#)
- pack_uim_UnblockPin
 - uim.h, [1475](#)
- pack_uim_UnblockPin_t, [489](#)
 - EncryptedPIN1, [490](#)
 - pIndicationToken, [490](#)
 - pKeyReferenceID, [490](#)
 - pinProtection, [490](#)
 - sessionInfo, [490](#)
 - Tlvresult, [490](#)
- pack_uim_VerifyPin
 - uim.h, [1476](#)
- pack_uim_VerifyPin_t, [490](#)
 - pEncryptedPIN1, [491](#)
 - pIndicationToken, [491](#)
 - pKeyReferenceID, [491](#)
 - sessionInfo, [491](#)
 - Tlvresult, [491](#)
 - verifyPIN, [491](#)
- pack_voice_AnswerUSSD
 - voice.h, [1492](#)
- pack_voice_AnswerUSSD_t, [491](#)
 - pInfo, [492](#)
- pack_voice_CancelUSSD
 - voice.h, [1492](#)
- pack_voice_OriginateUSSD
 - voice.h, [1493](#)
- pack_voice_OriginateUSSD_t, [492](#)
 - pInfo, [492](#)
- pack_voice_SLQSOrganateUSSD
 - voice.h, [1493](#)
- pack_voice_SLQSOrganateUSSD_t, [492](#)
 - ussDCS, [493](#)
 - ussData, [493](#)
 - ussLen, [493](#)
- pack_voice_SLQSVoiceALSSelectLine
 - voice.h, [1493](#)
- pack_voice_SLQSVoiceALSSelectLine_t, [493](#)
 - lineValue, [493](#)
- pack_voice_SLQSVoiceALSSetLineSwitching
 - voice.h, [1494](#)
- pack_voice_SLQSVoiceALSSetLineSwitching_t, [493](#)
 - switchOption, [493](#)
- pack_voice_SLQSVoiceAnswerCall
 - voice.h, [1494](#)
- pack_voice_SLQSVoiceAnswerCall_t, [494](#)
 - pCallId, [494](#)
- pack_voice_SLQSVoiceBindSubscription
 - voice.h, [1494](#)
- pack_voice_SLQSVoiceBindSubscription_t, [494](#)
 - subsType, [494](#)
- pack_voice_SLQSVoiceBurstDTMF

- voice.h, [1495](#)
- pack_voice_SLQSVoiceBurstDTMF_t, [494](#)
- pack_voice_SLQSVoiceDialCall
 - voice.h, [1495](#)
- pack_voice_SLQSVoiceDialCall_t, [495](#)
 - callNumber, [496](#)
 - pCLIRType, [496](#)
 - pCUGInfo, [496](#)
 - pCallPartySubAdd, [496](#)
 - pCallType, [496](#)
 - pEmergencyCategory, [496](#)
 - pSvcType, [497](#)
 - pUUSInfo, [497](#)
- pack_voice_SLQSVoiceEndCall
 - voice.h, [1495](#)
- pack_voice_SLQSVoiceEndCall_t, [497](#)
 - pCallId, [497](#)
- pack_voice_SLQSVoiceGetAllCallInfo
 - voice.h, [1496](#)
- pack_voice_SLQSVoiceGetCLIP
 - voice.h, [1498](#)
- pack_voice_SLQSVoiceGetCLIR
 - voice.h, [1498](#)
- pack_voice_SLQSVoiceGetCNAP
 - voice.h, [1498](#)
- pack_voice_SLQSVoiceGetCOLP
 - voice.h, [1499](#)
- pack_voice_SLQSVoiceGetCOLR
 - voice.h, [1499](#)
- pack_voice_SLQSVoiceGetCallBarring
 - voice.h, [1496](#)
- pack_voice_SLQSVoiceGetCallBarring_t, [497](#)
 - pSvcClass, [498](#)
 - reason, [498](#)
- pack_voice_SLQSVoiceGetCallForwardingStatus
 - voice.h, [1497](#)
- pack_voice_SLQSVoiceGetCallForwardingStatus_t, [498](#)
 - pSvcClass, [499](#)
 - Reason, [499](#)
- pack_voice_SLQSVoiceGetCallInfo
 - voice.h, [1497](#)
- pack_voice_SLQSVoiceGetCallInfo_t, [499](#)
 - callId, [499](#)
- pack_voice_SLQSVoiceGetCallWaiting
 - voice.h, [1497](#)
- pack_voice_SLQSVoiceGetCallWaiting_t, [499](#)
 - pSvcClass, [500](#)
- pack_voice_SLQSVoiceGetConfig
 - voice.h, [1499](#)
- pack_voice_SLQSVoiceGetConfig_t, [500](#)
 - pAMRStatus, [501](#)
 - pAirTimer, [501](#)
 - pAutoAnswer, [501](#)
 - pNamID, [501](#)
 - pPrefVoicePrivacy, [501](#)
 - pPrefVoiceSO, [501](#)
 - pRoamTimer, [501](#)
 - pTTYMode, [501](#)
 - pVoiceDomainPref, [501](#)
- pack_voice_SLQSVoiceIndicationRegister
 - voice.h, [1500](#)
- pack_voice_SLQSVoiceIndicationRegister_t, [502](#)
 - pRegDTMFEvents, [502](#)
 - pRegVoicePrivacyEvents, [502](#)
 - pSuppsNotifEvents, [502](#)
- pack_voice_SLQSVoiceManageCalls
 - voice.h, [1500](#)
- pack_voice_SLQSVoiceManageCalls_t, [502](#)
 - pCallId, [503](#)
 - SUPSType, [503](#)
- pack_voice_SLQSVoiceOrigUSSDNoWait
 - voice.h, [1500](#)
- pack_voice_SLQSVoiceOrigUSSDNoWait_t, [503](#)
- pack_voice_SLQSVoiceSendFlash
 - voice.h, [1501](#)
- pack_voice_SLQSVoiceSendFlash_t, [504](#)
 - pCallId, [504](#)
 - pFlashPayLd, [504](#)
 - pFlashType, [504](#)
- pack_voice_SLQSVoiceSetCallBarringPassword
 - voice.h, [1501](#)
- pack_voice_SLQSVoiceSetCallBarringPassword_t, [504](#)
 - newPasswd, [505](#)
 - newPasswdAgain, [505](#)
 - oldPasswd, [505](#)
 - Reason, [505](#)
- pack_voice_SLQSVoiceSetConfig
 - voice.h, [1501](#)
- pack_voice_SLQSVoiceSetConfig_t, [505](#)
 - pAirTimerConfig, [507](#)
 - pAutoAnswer, [507](#)
 - pPrefVoiceDomain, [507](#)
 - pPrefVoiceSO, [507](#)
 - pRoamTimerConfig, [507](#)
 - pTTYMode, [507](#)
- pack_voice_SLQSVoiceSetPreferredPrivacy
 - voice.h, [1502](#)
- pack_voice_SLQSVoiceSetPreferredPrivacy_t, [507](#)
 - privacyPref, [507](#)
- pack_voice_SLQSVoiceSetSUPSService
 - voice.h, [1502](#)
- pack_voice_SLQSVoiceSetSUPSService_t, [507](#)
 - pCallBarringPasswd, [509](#)
 - pCallForwardingNumber, [509](#)
 - pServiceClass, [509](#)
 - pTimerVal, [509](#)
 - reason, [509](#)
 - voiceSvc, [509](#)
- pack_voice_SLQSVoiceStartContDTMF
 - voice.h, [1503](#)
- pack_voice_SLQSVoiceStartContDTMF_t, [509](#)
 - DTMFdigit, [510](#)
 - pCallId, [510](#)
- pack_voice_SLQSVoiceStopContDTMF
 - voice.h, [1503](#)
- pack_voice_SLQSVoiceStopContDTMF_t, [510](#)

- callID, [510](#)
- pack_wds_DHCPv4ClientLeaseChange
 - wds.h, [1527](#)
- pack_wds_DHCPv4ClientLeaseChange_t, [510](#)
 - pEnableNotification, [511](#)
- pack_wds_GetAutoconnect
 - wds.h, [1528](#)
- pack_wds_GetByteTotals
 - wds.h, [1528](#)
- pack_wds_GetConnectionRate
 - wds.h, [1528](#)
- pack_wds_GetDataBearerTechnology
 - wds.h, [1529](#)
- pack_wds_GetDefaultProfile
 - wds.h, [1529](#)
- pack_wds_GetDefaultProfile_t, [511](#)
 - profiletype, [511](#)
- pack_wds_GetDefaultProfileNum
 - wds.h, [1530](#)
- pack_wds_GetDefaultProfileNum_t, [511](#)
 - family, [511](#)
 - type, [511](#)
- pack_wds_GetDefaultProfileV2
 - wds.h, [1530](#)
- pack_wds_GetDefaultProfileV2_t, [512](#)
 - profiletype, [512](#)
- pack_wds_GetDormancyState
 - wds.h, [1530](#)
- pack_wds_GetDormancyState_t, [512](#)
- pack_wds_GetLastMobileIPError
 - wds.h, [1531](#)
- pack_wds_GetLastMobileIPError_t, [512](#)
- pack_wds_GetMobileIP
 - wds.h, [1531](#)
- pack_wds_GetMobileIP_t, [512](#)
- pack_wds_GetMobileIPProfile
 - wds.h, [1532](#)
- pack_wds_GetMobileIPProfile_t, [513](#)
 - index, [513](#)
- pack_wds_GetPacketStatistics
 - wds.h, [1532](#)
- pack_wds_GetPacketStatistics_t, [513](#)
 - pStatMask, [514](#)
- pack_wds_GetPacketStatus
 - wds.h, [1533](#)
- pack_wds_GetPacketStatus_t, [514](#)
 - statmask, [514](#)
- pack_wds_GetSessionDuration
 - wds.h, [1533](#)
- pack_wds_GetSessionDuration_t, [514](#)
- pack_wds_GetSessionDurationV2
 - wds.h, [1533](#)
- pack_wds_GetSessionState
 - wds.h, [1534](#)
- pack_wds_RMSetTransferStatistics
 - wds.h, [1534](#)
- pack_wds_RMSetTransferStatistics_t, [514](#)
 - RmTrasnferStaticsReq, [515](#)
- pack_wds_SLQSCreateProfile
 - wds.h, [1538](#)
- pack_wds_SLQSCreateProfile_t, [521](#)
 - pCurProfile, [522](#)
 - pProfileId, [522](#)
 - pProfileType, [522](#)
- pack_wds_SLQSDeleteProfile
 - wds.h, [1538](#)
- pack_wds_SLQSDeleteProfile_t, [522](#)
 - profileIndex, [522](#)
 - profileType, [523](#)
- pack_wds_SLQSGet3GPPConfigItem
 - wds.h, [1538](#)
- pack_wds_SLQSGetCurrDataSystemStat
 - wds.h, [1539](#)
- pack_wds_SLQSGetCurrDataSystemStat_t, [523](#)
- pack_wds_SLQSGetCurrentChannelRate
 - wds.h, [1539](#)
- pack_wds_SLQSGetDUNCallInfo
 - wds.h, [1540](#)
- pack_wds_SLQSGetDUNCallInfo_t, [523](#)
 - Mask, [524](#)
 - pReportChannelRate, [524](#)
 - pReportConnStatus, [524](#)
 - pReportDataBearerTech, [524](#)
 - pReportDormStatus, [524](#)
 - pTransferStatInd, [524](#)
- pack_wds_SLQSGetDataBearerTechnology
 - wds.h, [1540](#)
- pack_wds_SLQSGetDataBearerTechnology_t, [523](#)
- pack_wds_SLQSGetProfileSettings
 - wds.h, [1541](#)
- pack_wds_SLQSGetProfileSettings_t, [525](#)
 - ProfileId, [525](#)
 - ProfileType, [525](#)
- pack_wds_SLQSGetProfileSettingsV2
 - wds.h, [1541](#)
- pack_wds_SLQSGetRuntimeSettings
 - wds.h, [1541](#)
- pack_wds_SLQSGetRuntimeSettings_t, [525](#)
 - pReqSettings, [526](#)
- pack_wds_SLQSModifyProfile
 - wds.h, [1542](#)
- pack_wds_SLQSModifyProfile_t, [526](#)
 - curProfile, [527](#)
 - pProfileId, [527](#)
 - pProfileType, [527](#)
- pack_wds_SLQSResetPacketStatics
 - wds.h, [1542](#)
- pack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [1544](#)
- pack_wds_SLQSSetDHCPv4ClientConfig_t, [530](#)
 - pProfileId, [530](#)
- pack_wds_SLQSSetLoopback
 - wds.h, [1544](#)
- pack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [1545](#)
- pack_wds_SLQSSetDHCPv4ClientConfig_t, [530](#)

- pHwConfig, [531](#)
 - pProfileId, [531](#)
- pack_wds_SLQSSetLoopback
 - wds.h, [1545](#)
- pack_wds_SLQSSetLoopback_t, [531](#)
 - loopbackMode, [531](#)
 - loopbackMultiplier, [531](#)
- pack_wds_SLQSSet3GPPConfigItem
 - wds.h, [1543](#)
- pack_wds_SLQSSet3GPPConfigItem_t, [527](#)
 - p3gppRelease, [528](#)
 - pProfileList, [528](#)
- pack_wds_SLQSSetIPFamilyPreference
 - wds.h, [1543](#)
- pack_wds_SLQSSetIPFamilyPreference_t, [529](#)
 - IPFamilyPreference, [529](#)
- pack_wds_SLQSSetWdsEventCallback
 - wds.h, [1543](#)
- pack_wds_SLQSSetWdsEventCallback_t, [529](#)
 - currentDataBearer, [530](#)
 - dataBearer, [530](#)
 - dataBearerTechExt, [530](#)
 - dataSystemStatus, [530](#)
 - dormancyStatus, [530](#)
 - interval, [530](#)
 - mobileIP, [530](#)
 - transferStats, [530](#)
- pack_wds_SLQSStartDataSession
 - wds.h, [1545](#)
- pack_wds_SLQSStartDataSession_t, [531](#)
 - pAuth, [532](#)
 - pPass, [532](#)
 - pTech, [532](#)
 - pUser, [532](#)
 - pprofileid3gpp, [532](#)
 - pprofileid3gpp2, [532](#)
- pack_wds_SLQSStopDataSession
 - wds.h, [1546](#)
- pack_wds_SLQSStopDataSession_t, [533](#)
 - psid, [533](#)
- pack_wds_SLQSWdsGoActive
 - wds.h, [1546](#)
- pack_wds_SLQSWdsGoDormant
 - wds.h, [1547](#)
- pack_wds_SLQSWdsSetEventReport
 - wds.h, [1547](#)
- pack_wds_SLQSWdsSetEventReport_t, [533](#)
 - pCurrChannelRateInd, [534](#)
 - pCurrDataBearerTechInd, [534](#)
 - pCurrPrefDataSysInd, [534](#)
 - pDataBearerTechInd, [534](#)
 - pDataCallStatusChangeInd, [534](#)
 - pDataSystemStatusChangeInd, [534](#)
 - pDormancyStatusInd, [535](#)
 - pMIPStatusInd, [535](#)
 - pTransferStatInd, [535](#)
- pack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, [1548](#)
- pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [535](#)
 - contextId, [535](#)
 - contextType, [535](#)
- pack_wds_SetAutoconnect
 - wds.h, [1535](#)
- pack_wds_SetAutoconnect_t, [515](#)
 - acroamsetting, [515](#)
 - acsetting, [515](#)
- pack_wds_SetDefaultProfile
 - wds.h, [1535](#)
- pack_wds_SetDefaultProfile_t, [515](#)
 - authentication, [517](#)
 - ipAddress, [517](#)
 - pApnname, [517](#)
 - pName, [517](#)
 - pPassword, [517](#)
 - pUsername, [517](#)
 - pdpType, [517](#)
 - primaryDNS, [517](#)
 - profileType, [517](#)
 - secondaryDNS, [517](#)
- pack_wds_SetDefaultProfileNum
 - wds.h, [1535](#)
- pack_wds_SetDefaultProfileNum_t, [517](#)
 - family, [518](#)
 - index, [518](#)
 - type, [518](#)
- pack_wds_SetMobileIP
 - wds.h, [1536](#)
- pack_wds_SetMobileIP_t, [518](#)
 - mode, [518](#)
- pack_wds_SetMobileIPParameters
 - wds.h, [1536](#)
- pack_wds_SetMobileIPParameters_t, [518](#)
 - pHA2002bis, [519](#)
 - pHAAAuthenticator, [519](#)
 - pMode, [519](#)
 - pReRegPeriod, [519](#)
 - pReRegTraffic, [519](#)
 - pRetryInterval, [519](#)
 - pRetryLimit, [519](#)
 - pSPC, [519](#)
- pack_wds_SetMobileIPProfile
 - wds.h, [1537](#)
- pack_wds_SetMobileIPProfile_t, [519](#)
 - index, [521](#)
 - pAAASPI, [521](#)
 - pAddress, [521](#)
 - pEnabled, [521](#)
 - pHASPI, [521](#)
 - pMNAHA, [521](#)
 - pMNHA, [521](#)
 - pNAI, [521](#)
 - pPrimaryHA, [521](#)
 - pRevTunneling, [521](#)
 - pSecondaryHA, [521](#)
 - spc, [521](#)
- pack_wds_SetMuxID

- wds.h, 1537
- PackCreateProfileOut, 535
 - ExtErrorCode, 536
 - ProfileIndex, 536
 - ProfileType, 536
- PackSwiAVMSSettingsAPNInfo, 539
 - bAPNLength, 539
 - bPWDLength, 539
 - bUnameLength, 540
 - pAPN, 540
 - pPWD, 540
 - pUname, 540
- PackSwiAVMSSettingsConnectionRetryTimers, 540
 - Timers, 540
- PackSwiAVMSSettingsPeriodsInfo, 541
 - max, 541
 - min, 541
- PackSwiAvmsSetSettingsAPNInfo, 537
 - bAPNLength, 538
 - bPWDLength, 538
 - bUnameLength, 538
 - szAPN, 538
 - szPWD, 538
 - szUname, 538
- PackSwiAvmsSetSettingsConnectionRetryTimers, 538
 - Timers, 538
- PackSwiAvmsSetSettingsPeriodInfo, 539
 - ulMax, 539
 - ulMin, 539
- package_name
 - unpack_omaDmFotaTlv_t, 819
- packageldStr
 - _litefw_FirmwareFileInfo, 36
- packageid_str
 - unpack_dms_GetFirmwareInfo_t, 628
- packetZone
 - nas_CDMA SysInfo, 181
- packetZoneValid
 - nas_CDMA SysInfo, 181
- packgetDyingGaspCfg, 536
 - pDestSMSContent, 536
 - pDestSMSNum, 536
- packgetDyingGaspStatistics, 536
 - pSMSAttemptedFlag, 537
 - pTimeStamp, 537
- ParamPresenceMask
 - pack_dms_UIMGetICCID_t, 365
 - unpack_audio_SLQSGetAudioPathConfig_t, 607
 - unpack_audio_SLQSGetAudioProfile_t, 609
 - unpack_audio_SLQSGetAudioVoITLBConfig_t, 609
 - unpack_audio_SLQSSetAudioVoITLBConfig_t, 610
 - unpack_cat_SetCatEventCallback_ind_t, 610
 - unpack_cat_SetCATEventCallback_t, 611
 - unpack_dms_GetActivationState_t, 612
 - unpack_dms_GetBandCapability_t, 614
 - unpack_dms_GetCrashAction_t, 614
 - unpack_dms_GetCustFeature_t, 617
 - unpack_dms_GetCustFeaturesV2_t, 618
 - unpack_dms_GetDeviceCap_t, 619
 - unpack_dms_GetDeviceCapabilities_t, 621
 - unpack_dms_GetDeviceCapabilitiesV2_t, 623
 - unpack_dms_GetDeviceHardwareRev_t, 624
 - unpack_dms_GetDeviceMfr_t, 625
 - unpack_dms_GetDeviceSerialNumbers_t, 627
 - unpack_dms_GetFirmwareInfo_t, 628
 - unpack_dms_GetFirmwareRevision_t, 629
 - unpack_dms_GetFirmwareRevisions_t, 631
 - unpack_dms_GetFSN_t, 631
 - unpack_dms_GetHardwareRevision_t, 632
 - unpack_dms_GetIMSI_t, 633
 - unpack_dms_GetManufacturer_t, 633
 - unpack_dms_GetModelID_t, 634
 - unpack_dms_GetNetworkTime_t, 635
 - unpack_dms_GetNetworkTimeV2_t, 636
 - unpack_dms_GetOfflineReason_t, 637
 - unpack_dms_GetPower_t, 638
 - unpack_dms_GetPRLVersion_t, 639
 - unpack_dms_GetSerialNumbers_t, 640
 - unpack_dms_GetUSBComp_t, 642
 - unpack_dms_GetVoiceNumber_t, 643
 - unpack_dms_PSMCfgChange_ind_t, 644
 - unpack_dms_ResetToFactoryDefaults_t, 645
 - unpack_dms_SetActivationStatusCallback_t, 645
 - unpack_dms_SetCrashAction_t, 646
 - unpack_dms_SetCustFeature_t, 646
 - unpack_dms_SetCustFeaturesV2_t, 647
 - unpack_dms_SetEventReport_ind_t, 648
 - unpack_dms_SetEventReport_t, 648
 - unpack_dms_SetFirmwarePreference_t, 649
 - unpack_dms_SetIndicationRegister_t, 649
 - unpack_dms_SetPower_t, 650
 - unpack_dms_SetUSBComp_t, 650
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, 653
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 654
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 655
 - unpack_dms_SLQSDmsSwiGetUimSelection_t, 656
 - unpack_dms_SLQSDmsSwiIndicationRegister_t, 657
 - unpack_dms_SLQSGetBandCapability_t, 661
 - unpack_dms_SLQSGetBandCapabilityExt_t, 665
 - unpack_dms_SLQSGetERIFile_t, 666
 - unpack_dms_SLQSGetPowerSaveModeConfig_t, 667
 - unpack_dms_SLQSSetPowerSaveModeConfig_t, 668
 - unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 668
 - unpack_dms_SLQSSwiGetCrashInfo_t, 669
 - unpack_dms_SLQSSwiGetDyingGaspCfg_t, 670
 - unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 670
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 671

- unpack_dms_SLQSSwiGetFwUpdateStatus_t, 673
- unpack_dms_SLQSSwiGetHostDevInfo_t, 674
- unpack_dms_SLQSSwiGetOSInfo_t, 675
- unpack_dms_SLQSSwiGetSerialNoExt_t, 676
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, 676
- unpack_dms_SLQSSwiSetHostDevInfo_t, 677
- unpack_dms_SLQSSwiSetOSInfo_t, 677
- unpack_dms_SLQSUIMGetState_t, 678
- unpack_dms_SwiEventReportCallBack_ind_t, 679
- unpack_dms_SwiSetEventReport_t, 679
- unpack_dms_SwiUimSelect_t, 680
- unpack_dms_UIMGetControlKeyStatus_t, 681
- unpack_dms_UIMGetICCID_t, 681
- unpack_dms_UIMGetPINStatus_t, 683
- unpack_dms_UIMSetControlKeyProtection_t, 684
- unpack_dms_UIMSetPINProtection_t, 685
- unpack_dms_UIMUnblockControlKey_t, 685
- unpack_fms_GetImagesPreference_t, 686
- unpack_fms_GetStoredImages_t, 687
- unpack_fms_SetImagesPreference_t, 687
- unpack_ims_SLQSGetIMSSMSConfig_t, 688
- unpack_ims_SLQSGetIMSUserConfig_t, 689
- unpack_ims_SLQSGetIMSVoIPConfig_t, 692
- unpack_ims_SLQSGetRegMgrConfig_t, 693
- unpack_ims_SLQSGetSIPConfig_t, 694
- unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 695
- unpack_ims_SLQSSetIMSSMSConfig_t, 696
- unpack_ims_SLQSSetIMSUserConfig_t, 696
- unpack_ims_SLQSSetIMSVoIPConfig_t, 697
- unpack_ims_SLQSSetRegMgrConfig_t, 697
- unpack_ims_SLQSSetSIPConfig_t, 698
- unpack_ims_SLQSSIPCfgCallBack_ind_t, 699
- unpack_ims_SLQSSMSCfgCallBack_ind_t, 700
- unpack_ims_SLQSUserCfgCallBack_ind_t, 701
- unpack_ims_SLQSVoIPCfgCallBack_ind_t, 703
- unpack_imsa_SLQSGetIMSARegStatus_t, 704
- unpack_imsa_SLQSGetIMSAServiceStatus_t, 706
- unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, 707
- unpack_imsa_SLQSImsaRatStatusCallBack_ind_t, 708
- unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 708
- unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 710
- unpack_loc_BestAvailPos_Ind_t, 716
- unpack_loc_CradleMountCallBack_Ind_t, 717
- unpack_loc_Delete_Assist_Data_t, 718
- unpack_loc_DeleteAssistData_Ind_t, 719
- unpack_loc_EngineState_Ind_t, 720
- unpack_loc_EventNMEA_Ind_t, 720
- unpack_loc_EventRegister_t, 721
- unpack_loc_EventTimeSyncCallback_Ind_t, 721
- unpack_loc_GetOpMode_Ind_t, 723
- unpack_loc_GetServer_Ind_t, 724
- unpack_loc_GnssSvInfo_Ind_t, 725
- unpack_loc_InjectPositionCallback_Ind_t, 726
- unpack_loc_InjectSensorDataCallback_Ind_t, 728
- unpack_loc_InjectTimeSyncDataCallback_Ind_t, 728
- unpack_loc_InjectUTCTimeCallback_Ind_t, 729
- unpack_loc_PositionRpt_Ind_t, 735
- unpack_loc_SensorStreamingCallback_Ind_t, 736
- unpack_loc_SetExtPowerConfig_Ind_t, 737
- unpack_loc_SetExtPowerState_t, 738
- unpack_loc_SetOperationMode_Ind_t, 739
- unpack_loc_SetOperationMode_t, 739
- unpack_loc_SetServer_Ind_t, 740
- unpack_loc_SLQSLOCGetBestAvailPos_t, 741
- unpack_loc_SLQSLOCGetOpMode_t, 741
- unpack_loc_Start_t, 742
- unpack_loc_Stop_t, 742
- unpack_nas_GetACCOLC_t, 743
- unpack_nas_GetANAAAAAuthenticationStatus_t, 743
- unpack_nas_GetCDMANetworkParameters_t, 745
- unpack_nas_GetHomeNetwork3GPP2_t, 748
- unpack_nas_GetHomeNetwork_t, 749
- unpack_nas_GetNetworkPreference_t, 750
- unpack_nas_GetRFInfo_t, 751
- unpack_nas_GetServingNetwork_t, 754
- unpack_nas_GetServingNetworkCapabilities_t, 755
- unpack_nas_GetSignalStrengths_t, 755
- unpack_nas_PerformNetworkScan_t, 757
- unpack_nas_SetDataCapabilitiesCallback_ind_t, 758
- unpack_nas_SetEventReportInd_t, 759
- unpack_nas_SetNasLTECphyCalIndCallback_ind_t, 760
- unpack_nas_SetNetworkPreference_t, 760
- unpack_nas_SetRoamingIndicatorCallback_ind_t, 761
- unpack_nas_SetServingSystemCallback_ind_t, 762
- unpack_nas_SLQSGetErrorRate_t, 763
- unpack_nas_SLqsGetLTECphyCAInfo_t, 764
- unpack_nas_SLQSGetNetworkTime_t, 764
- unpack_nas_SLQSGetOperatorNameData_t, 765
- unpack_nas_SLQSGetPLMNName_t, 768
- unpack_nas_SLQSGetServingSystem_t, 773
- unpack_nas_SLQSGetSignalStrength_t, 776
- unpack_nas_SLQSGetSysInfo_t, 779
- unpack_nas_SLQSGetSysSelectionPref_t, 784
- unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t, 788
- unpack_nas_SLQSNasGet3GPP2Subscription_t, 789
- unpack_nas_SLQSNasGetCellLocationInfo_t, 790
- unpack_nas_SLQSNASGeteDRXParams_t, 792
- unpack_nas_SLQSNASGeteDRXParamsExt_t, 793
- unpack_nas_SLQSNASGetForbiddenNetworks_t, 794
- unpack_nas_SLQSNasGetHDRColorCode_t, 794
- unpack_nas_SLQSNasGetSigInfo_t, 796

- unpack_nas_SLQSNasGetTxRxInfo_t, 797
- unpack_nas_SLQSNasNetworkRejectCallback_-
Ind_t, 800
- unpack_nas_SLQSNasNetworkTimeCallBack_ind_
t, 801
- unpack_nas_SLQSNasSigInfoCallback_ind_t, 802
- unpack_nas_SLQSNASSwiGetChannelLock_t,
803
- unpack_nas_SLQSNasSwiModemStatus_t, 804
- unpack_nas_SLQSNasSwiOTAMessageCallback_
ind_t, 805
- unpack_nas_SLQSNasTimerCallback_ind_t, 805
- unpack_nas_SLQSSetSysSelectionPrefCallBack_
ind_t, 806
- unpack_nas_SLQSSwiGetHDRPersonality_t, 807
- unpack_nas_SLQSSwiGetHDRProtSubtype_t, 808
- unpack_nas_SLQSSwiGetHRPDStats_t, 809
- unpack_nas_SLQSSwiGetLteCQI_t, 810
- unpack_nas_SLQSSwiGetLteScCRxInfo_t, 810
- unpack_nas_SLQSSwiHDRPersonalityCallback_-
Ind_t, 811
- unpack_nas_SLQSSwiNetworkDebug_t, 812
- unpack_nas_SLQSSwiRandIndicatorCallback_-
Ind_t, 812
- unpack_nas_SLQSSysInfoCallback_ind_t, 816
- unpack_pds_GetPDSDDefaults_t, 821
- unpack_pds_GetPDSSState_t, 822
- unpack_pds_GetPortAutomaticTracking_t, 822
- unpack_pds_GetServiceAutomaticTracking_t, 823
- unpack_pds_GetXTRAAutomaticDownload_t, 824
- unpack_pds_GetXTRANetwork_t, 824
- unpack_pds_GetXTRAValidity_t, 825
- unpack_pds_SetEventReport_Ind_t, 827
- unpack_pds_SetPdsState_Ind_t, 828
- unpack_pds_SLQSGetAGPSConfig_t, 829
- unpack_pds_SLQSGetGPSStateInfo_t, 833
- unpack_qos_BindDataPort_t, 834
- unpack_qos_SLQSQosGetNetworkStatus_t, 840
- unpack_qos_SLQSQosSwiReadApnExtraParams_
t, 841
- unpack_qos_SLQSQosSwiReadDataStats_t, 843
- unpack_qos_SLQSSetQosEventCallback_ind_t,
844
- unpack_qos_SLQSSetQosEventCallback_t, 844
- unpack_qos_SLQSSetQosNWStatusCallback_ind_
t, 845
- unpack_qos_SLQSSetQosPriEventCallback_ind_-
t, 845
- unpack_qos_SLQSSetQosStatusCallback_ind_t,
847
- unpack_result_t, 856
- unpack_rms_GetSMSWake_t, 857
- unpack_rms_SetSMSWake_t, 858
- unpack_RMTransferStatistics_ind_t, 858
- unpack_sar_SLQSGetRfSarState_t, 859
- unpack_sms_GetSMSCAddress_t, 860
- unpack_sms_SaveSMS_t, 861
- unpack_sms_SendSMS_t, 861
- unpack_sms_SetNewSMSCallback_ind_t, 863
- unpack_sms_SetNewSMSCallback_t, 863
- unpack_sms_SetSMSCAddress_t, 864
- unpack_sms_SLQSDeleteSMS_t, 864
- unpack_sms_SLQSGetIndicationRegister_t, 865
- unpack_sms_SLQSGetMessageWaiting_t, 865
- unpack_sms_SLQSGetSMS_t, 866
- unpack_sms_SLQSGetSmsBroadcastConfig_t,
867
- unpack_sms_SLQSGetSMSList_t, 868
- unpack_sms_SLQSGetTransLayerInfo_t, 868
- unpack_sms_SLQSGetTransNWRegInfo_t, 869
- unpack_sms_SLQSModifySMSStatus_t, 869
- unpack_sms_SLQSNWRegInfoCallback_ind_t,
870
- unpack_sms_SLQSSendAsyncSMS_t, 870
- unpack_sms_SLQSSetIndicationRegister_t, 871
- unpack_sms_SLQSSetSmsBroadcastActivation_t,
871
- unpack_sms_SLQSSetSmsBroadcastConfig_t,
872
- unpack_sms_SLQSSetSmsStorage_t, 872
- unpack_sms_SLQSSmsGetMaxStorageSize_t,
872
- unpack_sms_SLQSSmsGetMessageProtocol_t,
873
- unpack_sms_SLQSSmsSetRoutes_t, 873
- unpack_sms_SLQSSwiGetSMSStorage_t, 874
- unpack_sms_SLQSTransLayerInfoCallback_ind_t,
875
- unpack_sms_SLQSWmsAsyncRawSendCallBack_
ind_t, 877
- unpack_sms_SLQSWmsMemoryFullCallBack_ind_
t, 878
- unpack_sms_SLQSWmsMessageWaitingCall-
Back_ind_t, 878
- unpack_swiaudio_SLQSGetM2MAudioProfile_t,
880
- unpack_swiaudio_SLQSGetM2MAudioVolume_t,
880
- unpack_swiaudio_SLQSGetM2MAVMute_t, 881
- unpack_swiaudio_SLQSGetM2MSpkrGain_t, 882
- unpack_swiaavms_SLQSAVMSSEventReportInd_t,
883
- unpack_swiaavms_SLQSAVMSGetSettings_t, 886
- unpack_swiaavms_SLQSAVMSGetSettings_v2_t,
888
- unpack_swiaavms_SLQSAVMSSEndSelection_t,
889
- unpack_swiaavms_SLQSAVMSSEndSessionGetInfo_t,
889
- unpack_swiaavms_SLQSAvmsSetEventReport_t,
890
- unpack_swiaavms_SLQSAVMSSEndSession_t, 890
- unpack_swiaavms_SLQSAVMSSEndSession_v2_t,
891
- unpack_swiaavms_SLQSAVMSStartSession_t, 891
- unpack_swiaavms_SLQSAVMSStopSession_t, 892

- unpack_swidms_SLQSSwiDmsGetHWWatchdog_t, 892
- unpack_swidms_SLQSSwiDmsGetMTU_t, 893
- unpack_swidms_SLQSSwiDmsGetSecureInfo_t, 894
- unpack_swidms_SLQSSwiDmsGetUsbComp_t, 895
- unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 895
- unpack_swidms_SLQSSwiDmsSetHWWatchdog_t, 896
- unpack_swidms_SLQSSwiDmsSetMTU_t, 896
- unpack_swidms_SLQSSwiDmsSetUsbComp_t, 897
- unpack_swiloc_SwiLocGetAutoStart_t, 899
- unpack_swioama_SLQSOMADMAAlertCallback_ind_t, 900
- unpack_swioama_SLQSOMADMGetSessionInfo_t, 903
- unpack_swioama_SLQSOMADMGetSessionInfo_Ext_t, 906
- unpack_swioama_SLQSOMADMGetSettings_t, 908
- unpack_swioama_SLQSOMADMStartSession_t, 909
- unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 909
- unpack_tmd_SLQSTmdGetMitigationDevList_t, 910
- unpack_tmd_SLQSTmdGetMitigationLvl_t, 911
- unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, 911
- unpack_tmd_SLQSTmdRegNotMitigationLvl_t, 912
- unpack_uim_ChangePin_t, 913
- unpack_uim_GetCardStatus_t, 913
- unpack_uim_GetCardStatusV2_t, 914
- unpack_uim_ReadTransparent_t, 915
- unpack_uim_SetPinProtection_t, 916
- unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 917
- unpack_uim_SLQSUIMAuthenticate_t, 918
- unpack_uim_SLQSUIMDepersonalization_t, 918
- unpack_uim_SLQSUIMEventRegister_t, 919
- unpack_uim_SLQSUIMGetConfiguration_t, 920
- unpack_uim_SLQSUIMGetFileAttributes_t, 921
- unpack_uim_SLQSUIMGetSlotsStatus_t, 921
- unpack_uim_SLQSUIMGetSlotsStatusV2_t, 922
- unpack_uim_SLQSUIMRefreshCallback_Ind_t, 923
- unpack_uim_SLQSUIMRefreshGetLastEvent_t, 923
- unpack_uim_SLQSUIMSetStatusChangeCallback_ind_t, 924
- unpack_uim_UnblockPin_t, 924
- unpack_uim_UnblockPinV2_t, 925
- unpack_uim_VerifyPin_t, 926
- unpack_voice_allCallStatusCallback_ind_t, 928
- unpack_voice_DTMFEventCallback_ind_t, 929
- unpack_voice_OTASPStatusCallback_ind_t, 931
- unpack_voice_SLQSOriinateUSSD_t, 933
- unpack_voice_SLQSVoiceAnswerCall_t, 934
- unpack_voice_SLQSVoiceBurstDTMF_t, 934
- unpack_voice_SLQSVoiceDialCall_t, 935
- unpack_voice_SLQSVoiceEndCall_t, 936
- unpack_voice_SLQSVoiceGetAllCallInfo_t, 939
- unpack_voice_SLQSVoiceGetCallBarring_t, 941
- unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 942
- unpack_voice_SLQSVoiceGetCallInfo_t, 945
- unpack_voice_SLQSVoiceGetCallWaiting_t, 947
- unpack_voice_SLQSVoiceGetCLIP_t, 948
- unpack_voice_SLQSVoiceGetCLIR_t, 950
- unpack_voice_SLQSVoiceGetCNAP_t, 951
- unpack_voice_SLQSVoiceGetCOLP_t, 953
- unpack_voice_SLQSVoiceGetCOLR_t, 954
- unpack_voice_SLQSVoiceGetConfig_t, 956
- unpack_voice_SLQSVoiceManageCalls_t, 957
- unpack_voice_SLQSVoiceSendFlash_t, 957
- unpack_voice_SLQSVoiceSetCallBarringPassword_t, 959
- unpack_voice_SLQSVoiceSetConfig_t, 960
- unpack_voice_SLQSVoiceSetSUPSService_t, 962
- unpack_voice_SLQSVoiceStartContDTMF_t, 962
- unpack_voice_SLQSVoiceStopContDTMF_t, 963
- unpack_voice_SLQSVoiceSUPSCallback_ind_t, 965
- unpack_voice_SUPSNotificationCallback_ind_t, 968
- unpack_voice_USSDNotificationCallback_ind_t, 968
- unpack_voice_VoiceInfoRecCallback_ind_t, 971
- unpack_voice_voicePrivacyChangeCallback_ind_t, 972
- unpack_wds_DHCPv4ClientLease_ind_t, 973
- unpack_wds_GetAutoconnect_t, 973
- unpack_wds_GetByteTotals_t, 974
- unpack_wds_GetConnectionRate_t, 975
- unpack_wds_GetDataBearerTechnology_t, 976
- unpack_wds_GetDefaultProfile_t, 979
- unpack_wds_GetDefaultProfileNum_t, 979
- unpack_wds_GetDefaultProfileV2_t, 982
- unpack_wds_GetDormancyState_t, 983
- unpack_wds_GetLastMobileIPError_t, 983
- unpack_wds_GetMobileIP_t, 984
- unpack_wds_GetMobileIPProfile_t, 986
- unpack_wds_GetPacketStatistics_t, 988
- unpack_wds_GetPacketStatus_t, 990
- unpack_wds_GetSessionDuration_t, 991
- unpack_wds_GetSessionDurationV2_t, 992
- unpack_wds_GetSessionState_t, 992
- unpack_wds_RMSetTransferStatistics_t, 993
- unpack_wds_SetMobileIPProfile_t, 993
- unpack_wds_SLQSCreateProfile_t, 994
- unpack_wds_SLQSDeleteProfile_t, 994
- unpack_wds_SLQSDUNCAllInfoCallBack_ind_t, 996

- unpack_wds_SLQSGet3GPPConfigItem_t, 998
- unpack_wds_SLQSGetCurrDataSystemStat_t, 998
- unpack_wds_SLQSGetCurrentChannelRate_t, 1000
- unpack_wds_SLQSGetDataBearerTechnology_t, 1001
- unpack_wds_SLQSGetDUNCallInfo_t, 1004
- unpack_wds_SLQSGetProfileSettings_t, 1004
- unpack_wds_SLQSGetProfileSettingsV2_t, 1005
- unpack_wds_SLQSGetRuntimeSettings_t, 1008
- unpack_wds_SLQSModifyProfile_t, 1009
- unpack_wds_SLQSSetIPFamilyPreference_t, 1009
- unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1011
- unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- unpack_wds_SLQSSetDHCPv4ClientConfig_t, 1018
- unpack_wds_SLQSSetLoopback_t, 1019
- unpack_wds_SLQSSetStartDataSession_t, 1019
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 1022
- partNoStr
 - _litefw_FirmwareFileInfo, 36
- path
 - uim_fileInfo, 590
- pathLen
 - uim_fileInfo, 590
- pbAuto
 - unpack_pds_GetPortAutomaticTracking_t, 823
 - unpack_pds_GetServiceAutomaticTracking_t, 823
- pbEnabled
 - unpack_pds_GetXTRAAutomaticDownload_t, 824
- pbPlatform
 - unpack_dms_GetOfflineReason_t, 637
- pci
 - nas_cellParams, 182
 - nas_ltePCI, 235
 - nas_PhyCaAggPcellInfo, 262
 - nas_PhyCaAggScellIndType, 263
 - nas_PhyCaAggScellInfo, 266
 - nas_umtsLTENbrCell, 314
 - NASPhyCaAggPcellInfo, 330
 - NASPhyCaAggScellArray, 332
 - NASPhyCaAggScellIndType, 334
 - NASPhyCaAggScellInfo, 335
- PcoList
 - LibPackPCOIDList, 106
- pcsDigit
 - nas_PlmnID, 268
- pcsfFQDNAddress
 - wds_PCSCFFQDNAddressList, 1095
- PdpConnState
 - unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, 707
- PdpStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, 378
- pdpType
 - pack_wds_SetDefaultProfile_t, 517
- pdptype
 - unpack_wds_GetDefaultProfile_t, 979
 - unpack_wds_GetDefaultProfileV2_t, 982
- pds.h, 1303
 - pack_pds_ForceXTRADownload, 1306
 - pack_pds_GetPDSDefaults, 1306
 - pack_pds_GetPDSState, 1307
 - pack_pds_GetPortAutomaticTracking, 1307
 - pack_pds_GetServiceAutomaticTracking, 1307
 - pack_pds_GetXTRAAutomaticDownload, 1308
 - pack_pds_GetXTRANetwork, 1308
 - pack_pds_GetXTRAValidity, 1308
 - pack_pds_PDSInjectTimeReference, 1309
 - pack_pds_ResetPDSData, 1309
 - pack_pds_SLQSGetAGPSConfig, 1312
 - pack_pds_SLQSGetGPSSStateInfo, 1312
 - pack_pds_SLQSPDSDeterminePosition, 1313
 - pack_pds_SLQSPDSInjectAbsoluteTimeReference, 1313
 - pack_pds_SLQSPDSInjectPositionData, 1313
 - pack_pds_SLQSSetAGPSConfig, 1314
 - pack_pds_SLQSSetPositionMethodState, 1314
 - pack_pds_SetEventReportCallback, 1309
 - pack_pds_SetPDSDefaults, 1310
 - pack_pds_SetPDSState, 1310
 - pack_pds_SetPortAutomaticTracking, 1310
 - pack_pds_SetServiceAutomaticTracking, 1311
 - pack_pds_SetXTRAAutomaticDownload, 1311
 - pack_pds_SetXTRANetwork, 1311
 - pack_pds_StartPDSTrackingSessionExt, 1314
 - pack_pds_StopPDSTrackingSession, 1315
 - unpack_pds_ForceXTRADownload, 1315
 - unpack_pds_ForceXTRADownload_t, 1305
 - unpack_pds_GetPDSDefaults, 1315
 - unpack_pds_GetPDSState, 1316
 - unpack_pds_GetPortAutomaticTracking, 1316
 - unpack_pds_GetServiceAutomaticTracking, 1316
 - unpack_pds_GetXTRAAutomaticDownload, 1317
 - unpack_pds_GetXTRANetwork, 1317
 - unpack_pds_GetXTRAValidity, 1318
 - unpack_pds_PDSInjectTimeReference, 1318
 - unpack_pds_PDSInjectTimeReference_t, 1305
 - unpack_pds_ResetPDSData, 1318
 - unpack_pds_ResetPDSData_t, 1305
 - unpack_pds_SLQSGetAGPSConfig, 1322
 - unpack_pds_SLQSGetGPSSStateInfo, 1322
 - unpack_pds_SLQSPDSDeterminePosition, 1322
 - unpack_pds_SLQSPDSDeterminePosition_t, 1306
 - unpack_pds_SLQSPDSInjectAbsoluteTimeReference, 1323
 - unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 1306
 - unpack_pds_SLQSPDSInjectPositionData, 1323
 - unpack_pds_SLQSPDSInjectPositionData_t, 1306
 - unpack_pds_SLQSSetAGPSConfig, 1323

- unpack_pds_SLQSSetAGPSConfig_t, 1306
- unpack_pds_SLQSSetPositionMethodState, 1324
- unpack_pds_SLQSSetPositionMethodState_t, 1306
- unpack_pds_SetEventReport_Ind, 1319
- unpack_pds_SetEventReportCallback, 1319
- unpack_pds_SetEventReportCallback_t, 1305
- unpack_pds_SetPDSDDefaults, 1319
- unpack_pds_SetPDSDDefaults_t, 1305
- unpack_pds_SetPDSSState, 1320
- unpack_pds_SetPDSSState_t, 1305
- unpack_pds_SetPdsState_Ind, 1320
- unpack_pds_SetPortAutomaticTracking, 1320
- unpack_pds_SetPortAutomaticTracking_t, 1305
- unpack_pds_SetServiceAutomaticTracking, 1321
- unpack_pds_SetServiceAutomaticTracking_t, 1306
- unpack_pds_SetXTRAAutomaticDownload, 1321
- unpack_pds_SetXTRAAutomaticDownload_t, 1306
- unpack_pds_SetXTRANetwork, 1321
- unpack_pds_SetXTRANetwork_t, 1306
- unpack_pds_StartPDSTrackingSessionExt, 1324
- unpack_pds_StartPDSTrackingSessionExt_t, 1306
- unpack_pds_StopPDSTrackingSession, 1324
- unpack_pds_StopPDSTrackingSession_t, 1306
- peakRate
 - unpack_qos_tokenBucket_t, 855
- peakThroughputClass
 - LibPackGPRSRequestedQoS, 105
 - wds_GPRSQoS, 1092
- PeriodicUpdateTimer
 - unpack_dms_PSMCfGChange_ind_t, 644
- periodicUpdateTimer
 - dms_PSMPeriodicUpdateTimerTlv, 71
- PeriodicUpdateTimerInd
 - dms_PSMPeriodicUpdateTimerIndTlv, 70
- PersistentLpm
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, 653
- PersistentTechPref
 - unpack_nas_GetNetworkPreference_t, 750
- persoFeature
 - appStats, 41
 - uim_appStatus, 579
- persoRetries
 - appStats, 41
 - uim_appStatus, 579
- persoState
 - appStats, 41
 - uim_appStatus, 579
- persoUnblockRetries
 - appStats, 41
 - uim_appStatus, 579
- pfailureCause
 - unpack_voice_SLQSOriginateUSSD_t, 933
- PhCtxtURI
 - ims_PhCtxtURIInfo, 90
- phase
 - nas_rxInfo, 291
- PhyCaAggPcellInfo
 - NasGetLTECphyCaInfo, 325
- PhyCaAggScellArray
 - NasGetLTECphyCaInfo, 325
- PhyCaAggScellDIBw
 - NasGetLTECphyCaInfo, 325
- PhyCaAggScellIndType
 - NasGetLTECphyCaInfo, 325
- PhyCaAggScellIndex
 - NasGetLTECphyCaInfo, 325
- PhyCaAggScellInfo
 - NasGetLTECphyCaInfo, 325
- PhysicalLayer
 - nas_protocolSubtypeElement, 272
- PilotEnergy
 - nas_NetworkStatEVDO, 252
- PilotPN
 - nas_PilotSetParams, 267
- PilotStrength
 - nas_PilotSetParams, 267
- PilotType
 - nas_PilotSetParams, 267
- pin1Len
 - uim_encryptedPIN1, 586
- pin1Retries
 - appStats, 42
 - uim_appStatus, 579
- pin1State
 - appStats, 42
 - uim_appStatus, 579
- pin1Val
 - uim_encryptedPIN1, 586
- pin2Retries
 - appStats, 42
 - uim_appStatus, 579
- pin2State
 - appStats, 42
 - uim_appStatus, 579
- pinID
 - uim_changeUIMPIN, 584
 - uim_setPINProtection, 600
 - uim_unblockUIMPIN, 604
 - uim_verifyUIMPIN, 605
- pinLen
 - uim_changeUIMPIN, 584
 - uim_verifyUIMPIN, 605
- pinLength
 - uim_setPINProtection, 600
- pinOperation
 - uim_setPINProtection, 600
- pinProtection
 - pack_uim_SetPinProtection_t, 482
 - pack_uim_UnblockPin_t, 490
- pinVal
 - uim_changeUIMPIN, 584
 - uim_verifyUIMPIN, 605
- pinValue
 - uim_setPINProtection, 600
- pkgDate

- unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgDateLength
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgDesc
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- PkgDescLength
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, [903](#)
- pkgDescLength
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- PkgDescription
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, [903](#)
- pkgInstallTime
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgInstallTimeLength
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- PkgName
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, [903](#)
- pkgName
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- PkgNameLength
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, [903](#)
- pkgNameLength
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgSize
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgVendorName
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgVendorNameLength
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgVersionName
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgVersionNameLength
 - unpack_swioama_SLQSOMADMGetSessionInfo-Ext_t, [906](#)
- pkgver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [671](#)
- PktErrRate
 - unpack_qos_swiQosFlow_t, [854](#)
- plasmaIDString
 - pack_dms_SLQSSwiSetHostDevInfo_t, [362](#)
 - unpack_dms_SLQSSwiGetHostDevInfo_t, [674](#)
- plmn
 - nas_GERANInfo, [202](#)
 - nas_LTEInfoIntrafreq, [227](#)
 - nas_UMTSInfo, [312](#)
- PlmnLen
 - nas_QmisNasSlqsNasPCICellInfo, [276](#)
- polarityIncluded
 - voice_lineCtrlInfo, [1068](#)
- port
 - loc_IPv4Info, [157](#)
 - loc_IPv6Info, [158](#)
 - unpack_qos_Port_t, [837](#)
- posDataNmea
 - pack_pds_SetEventReportCallback_t, [434](#)
- posSrc
 - unpack_pds_SetEventReport_Ind_t, [827](#)
- PositionDataNMEA
 - unpack_pds_SetEventReport_Ind_t, [827](#)
- positionSrc
 - pack_loc_SLQSLOCInjectPosition_t, [388](#)
- PowerOffMode
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [653](#)
- pprofileid3gpp
 - pack_wds_SLQSStartDataSession_t, [532](#)
- pprofileid3gpp2
 - pack_wds_SLQSStartDataSession_t, [532](#)
- prDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- prDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- prPCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- prPCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, [1022](#)
- Precedence
 - unpack_qos_swiQosFilter_t, [850](#)
- precedenceClass
 - LibPackGPRSRequestedQoS, [105](#)
 - wds_GPRSQoS, [1092](#)
- prefNetwork
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [999](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [1017](#)
- preference
 - pack_pds_SetXTRANetwork_t, [437](#)
- Preferred
 - nas_QmiNas3GppNetworkInfo, [274](#)
- prefixLen
 - unpack_qos_IPv6Addr_t, [836](#)
- presentationInd
 - voice_ECTNum, [1064](#)
 - voice_remotePartyNum, [1076](#)
- priCSCFPort
 - ims_PCSCFPortInfo, [90](#)

- priChA
 - nas_CDMAChannel, [175](#)
- priChB
 - nas_CDMAChannel, [175](#)
- priSize
 - unpack_dms_GetFirmwareRevisions_t, [631](#)
- priVersionStr
 - _litefw_FirmwareFileInfo, [36](#)
- pridns
 - unpack_wds_GetDefaultProfile_t, [979](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- pridnsV6
 - unpack_wds_GetDefaultProfile_t, [979](#)
 - unpack_wds_GetDefaultProfileV2_t, [982](#)
- primaryDNS
 - pack_wds_SetDefaultProfile_t, [517](#)
- PrimaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- PrimaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- primaryHA
 - unpack_wds_GetMobileIPProfile_t, [986](#)
- privacyPref
 - pack_voice_SLQSVoiceSetPreferredPrivacy_t, [507](#)
- priver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [671](#)
- priversion_str
 - unpack_dms_GetFirmwareInfo_t, [628](#)
- Profile
 - pack_audio_SLQSGetAudioPathConfig_t, [343](#)
 - pack_audio_SLQSGetAudioVolTLBConfig_t, [344](#)
 - pack_audio_SLQSSetAudioPathConfig_t, [346](#)
 - pack_audio_SLQSSetAudioProfile_t, [347](#)
 - pack_audio_SLQSSetAudioVolTLBConfig_t, [348](#)
 - pack_swiaudio_SLQSGetM2MAudioVolume_t, [457](#)
 - pack_swiaudio_SLQSGetM2MAVMute_t, [458](#)
 - pack_swiaudio_SLQSGetM2MSpkrGain_t, [458](#)
 - pack_swiaudio_SLQSSetM2MAudioAVCFG_t, [459](#)
 - pack_swiaudio_SLQSSetM2MAudioProfile_t, [460](#)
 - pack_swiaudio_SLQSSetM2MAudioVolume_t, [461](#)
 - pack_swiaudio_SLQSSetM2MAVMute_t, [462](#)
 - pack_swiaudio_SLQSSetM2MSpkrGain_t, [462](#)
 - unpack_audio_SLQSGetAudioProfile_t, [609](#)
 - unpack_swiaudio_SLQSGetM2MAudioProfile_t, [880](#)
- ProfileID
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- ProfileId
 - pack_wds_SLQSGetProfileSettings_t, [525](#)
- profileId
 - wds_DHCPPProfileIdTlv, [1087](#)
 - wds_DHCPv4ProfileId, [1090](#)
 - wdsDhcpv4ProfileId, [1104](#)
- ProfileId3GPP2
 - unpack_qos_swiQosFlow_t, [854](#)
- ProfileIdTlv
 - unpack_wds_DHCPv4ClientLease_ind_t, [973](#)
- ProfileIndex
 - PackCreateProfileOut, [536](#)
- profileIndex
 - pack_wds_SLQSDestroyProfile_t, [522](#)
 - wds_ProfileIdentifier, [1096](#)
- profileList
 - unpack_wds_SLQSGet3GPPConfigItem_t, [998](#)
- ProfileName
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- ProfileType
 - pack_wds_SLQSGetProfileSettings_t, [525](#)
 - PackCreateProfileOut, [536](#)
 - unpack_wds_SLQSGetProfileSettings_t, [1004](#)
 - unpack_wds_SLQSGetProfileSettingsV2_t, [1005](#)
- profileType
 - pack_wds_SetDefaultProfile_t, [517](#)
 - pack_wds_SLQSDestroyProfile_t, [523](#)
 - wds_DHCPPProfileIdTlv, [1087](#)
 - wds_DHCPv4ProfileId, [1090](#)
 - wds_ProfileIdentifier, [1097](#)
 - wdsDhcpv4ProfileId, [1104](#)
- profiletype
 - pack_wds_GetDefaultProfile_t, [511](#)
 - pack_wds_GetDefaultProfileV2_t, [512](#)
- PromptFwDownload
 - pack_swiaavms_SLQSAVMSSetSettings_t, [465](#)
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, [467](#)
- PromptFwUpdate
 - pack_swiaavms_SLQSAVMSSetSettings_t, [465](#)
 - pack_swiaavms_SLQSAVMSSetSettings_v2_t, [467](#)
- Protocol
 - unpack_nas_GetCDMANetworkParameters_t, [745](#)
- ProvisionStatus
 - voice_CLIPResp, [1055](#)
 - voice_CLIRResp, [1056](#)
 - voice_CNAPResp, [1057](#)
 - voice_COLPResp, [1057](#)
 - voice_COLRResp, [1058](#)
- psAttachState
 - nas_servSystem, [296](#)
 - NASServingSystemInfo, [341](#)
- psBarStatus
 - nas_CallBarringSysInfo, [173](#)
 - nas_callBarStatus, [174](#)
- psState
 - nas_CommInfo, [186](#)
- psc
 - nas_UMTSInfo, [312](#)
 - nas_wcdmaCellInfo, [316](#)
 - nas_WCDMASysInfo, [321](#)
- pscValid
 - nas_WCDMASysInfo, [321](#)
- pscsfIPv4Addr
 - wds_PCSCFIPv4ServerAddressList, [1096](#)
- psetting
 - unpack_wds_GetAutoconnect_t, [973](#)
- psid
 - pack_wds_SLQSStopDataSession_t, [533](#)

- unpack_wds_SLQSSStartDataSession_t, 1019
- puk1Retries
 - appStats, 42
 - uim_appStatus, 579
- puk2Retries
 - appStats, 42
 - uim_appStatus, 579
- pukLen
 - uim_unblockUIMPIN, 604
- pukVal
 - uim_unblockUIMPIN, 604
- pukValue
 - pack_dms_UIMUnlockPIN_t, 368
- pwd
 - unpack_wds_GetDefaultProfileV2_t, 982
- pwdsize
 - unpack_wds_GetDefaultProfileV2_t, 982
- pwrDenialTime
 - voice_lineCtrlInfo, 1068
- QMI_LITE_WDS_CURRENT_CALL_DB_MASK
 - wds.h, 1527
- QMI_LITE_WDS_LAST_CALL_DB_MASK
 - wds.h, 1527
- QCI
 - LibPackQosClassID, 142
- QFlowState
 - unpack_qos_QosFlowInfo_t, 839
- QLIC
 - nas_DeviceConfigDetail, 193
- QMI pack/unpack (lite-qmi), 31
- qaGobiApiTableBandClasses.h, 1325
- qaGobiApiTableCallControlReturnReasons.h, 1328
- qaGobiApiTableCallEndReasons.h, 1329
- qaGobiApiTableCarrierCodes.h, 1345
- qaGobiApiTableCodingScheme.h, 1347
- qaGobiApiTableGpsCapabilityCodes.h, 1350
- qaGobiApiTablePowerModes.h, 1350
- qaGobiApiTableRadioInterfaces.h, 1351
- qaGobiApiTableRegionCodes.h, 1352
- qaGobiApiTableServiceOptions.h, 1352
- qaGobiApiTableSupServiceInfoClasses.h, 1355
- qaGobiApiTableSwiAudio.h, 1355
- qaGobiApiTableSwiOMADMSessionStatus.h, 1356
- qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 1357
- qaGobiApiTableVoiceCallEndReasons.h, 1358
- qm_wds_ds_profile_extended_err_codes
 - qmerrno.h, 1371
- qmTlvResult, 542
 - DeviceError, 542
 - DeviceResult, 542
 - TlvPresenceMask, 542
 - TlvResultCode, 542
- qmerrno.h
 - eQCWWAN_ERR_API_MUTEX_TIMEOUT, 1368
 - eQCWWAN_ERR_BUFFER_SZ, 1367
 - eQCWWAN_ERR_CANCEL_OP, 1368
 - eQCWWAN_ERR_DRIVER, 1368
 - eQCWWAN_ERR_ENUM_BEGIN, 1367
 - eQCWWAN_ERR_ENUM_END, 1368
 - eQCWWAN_ERR_FILE_COPY, 1367
 - eQCWWAN_ERR_FILE_OPEN, 1367
 - eQCWWAN_ERR_GENERAL, 1367
 - eQCWWAN_ERR_INTERNAL, 1367
 - eQCWWAN_ERR_INVALID_ARG, 1367
 - eQCWWAN_ERR_INVALID_DEVID, 1367
 - eQCWWAN_ERR_INVALID_FILE, 1367
 - eQCWWAN_ERR_INVALID_QMI_RSP, 1367
 - eQCWWAN_ERR_INVALID_XID, 1368
 - eQCWWAN_ERR_MALFORMED_QMI_RSP, 1367
 - eQCWWAN_ERR_MEMORY, 1367
 - eQCWWAN_ERR_MULTIPLE_DEVICES, 1368
 - eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPOR-
TED, 1368
 - eQCWWAN_ERR_NO_CANCELABLE_OP, 1368
 - eQCWWAN_ERR_NO_CONNECTION, 1367
 - eQCWWAN_ERR_NO_DEVICE, 1367
 - eQCWWAN_ERR_NO_SIGNAL, 1368
 - eQCWWAN_ERR_NONE, 1367
 - eQCWWAN_ERR_NULL_TLV, 1371
 - eQCWWAN_ERR_OFFLINE, 1368
 - eQCWWAN_ERR_PDU_GENERATION, 1368
 - eQCWWAN_ERR_QMI_ABORTED, 1368
 - eQCWWAN_ERR_QMI_ACCESS_DENIED, 1370
 - eQCWWAN_ERR_QMI_ACK_NOT_SENT, 1370
 - eQCWWAN_ERR_QMI_ARG_TOO_LONG, 1368
 - eQCWWAN_ERR_QMI_AUTHENTICATION_FAI-
LED, 1369
 - eQCWWAN_ERR_QMI_AUTHENTICATION_LO-
CK, 1369
 - eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPP-
ORTED, 1370
 - eQCWWAN_ERR_QMI_CALL_FAILED, 1368
 - eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 1371
 - eQCWWAN_ERR_QMI_CARD_CALL_CONTRO-
L_FAILED, 1370
 - eQCWWAN_ERR_QMI_CAT_END, 1371
 - eQCWWAN_ERR_QMI_CAT_START, 1371
 - eQCWWAN_ERR_QMI_CAUSE_CODE, 1369
 - eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUST-
ED, 1368
 - eQCWWAN_ERR_QMI_CONNECT, 1367
 - eQCWWAN_ERR_QMI_DEVICE_IN_USE, 1368
 - eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 1369
 - eQCWWAN_ERR_QMI_DEVICE_STORAGE_FU-
LL, 1369
 - eQCWWAN_ERR_QMI_DISABLED, 1369
 - eQCWWAN_ERR_QMI_ENCODING, 1369
 - eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAI-
LURE, 1371
 - eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 1371
 - eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 1370

- eQCWWAN_ERR_QMI_FDN_RESTRICT, [1370](#)
- eQCWWAN_ERR_QMI_FLOW_SUSPENDED, [1369](#)
- eQCWWAN_ERR_QMI_GENERAL, [1369](#)
- eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, [1370](#)
- eQCWWAN_ERR_QMI_IFACE, [1367](#)
- eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, [1370](#)
- eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, [1369](#)
- eQCWWAN_ERR_QMI_INCORRECT_PIN, [1368](#)
- eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, [1370](#)
- eQCWWAN_ERR_QMI_INJECT_TIMEOUT, [1370](#)
- eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, [1369](#)
- eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, [1369](#)
- eQCWWAN_ERR_QMI_INTERNAL, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_ARG, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, [1371](#)
- eQCWWAN_ERR_QMI_INVALID_HANDLE, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_ID, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_INDEX, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_OPERATION, [1370](#)
- eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PINID, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_QMI_CMD, [1370](#)
- eQCWWAN_ERR_QMI_INVALID_QOS_ID, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_TECH_PREF, [1368](#)
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, [1371](#)
- eQCWWAN_ERR_QMI_INVALID_TRANSITION, [1369](#)
- eQCWWAN_ERR_QMI_INVALID_TX_ID, [1368](#)
- eQCWWAN_ERR_QMI_MALFORMED_MSG, [1368](#)
- eQCWWAN_ERR_QMI_MAX, [1370](#)
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, [1369](#)
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, [1369](#)
- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, [1369](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, [1369](#)
- eQCWWAN_ERR_QMI_MISSING_ARG, [1368](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED, [1370](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED, [1370](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY, [1369](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, [1369](#)
- eQCWWAN_ERR_QMI_NO_EFFECT, [1368](#)
- eQCWWAN_ERR_QMI_NO_ENTRY, [1369](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE, [1368](#)
- eQCWWAN_ERR_QMI_NO_MEMORY, [1368](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, [1368](#)
- eQCWWAN_ERR_QMI_NO_RADIO, [1370](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, [1370](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS, [1368](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, [1369](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED, [1368](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED, [1370](#)
- eQCWWAN_ERR_QMI_OFFSET, [1368](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, [1368](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, [1368](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, [1370](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL, [1368](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED, [1369](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, [1369](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH, [1370](#)
- eQCWWAN_ERR_QMI_REQ, [1367](#)
- eQCWWAN_ERR_QMI_REQ_SCH, [1367](#)
- eQCWWAN_ERR_QMI_REQ_TO, [1367](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, [1369](#)

- eQCWWAN_ERR_QMI_RSP, [1367](#)
- eQCWWAN_ERR_QMI_RSP_TO, [1367](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER, [1370](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, [1370](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE, [1369](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID, [1369](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, [1369](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, [1370](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, [1369](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR, [1370](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, [1370](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE, [1370](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, [1368](#)
- eQCWWAN_ERR_QMI_UNKNOWN, [1369](#)
- eQCWWAN_ERR_QMI_WIDTH, [1371](#)
- eQCWWAN_ERR_RESET, [1368](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR, [1370](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, [1370](#)
- eQCWWAN_ERR_SWICM_END, [1371](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, [1370](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, [1370](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, [1370](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, [1370](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, [1370](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, [1370](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, [1370](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, [1370](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE, [1370](#)
- eQCWWAN_ERR_SWICM_START, [1370](#)
- eQCWWAN_ERR_SWICM_TIMEOUT, [1370](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN, [1370](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP, [1370](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN, [1370](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP, [1370](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, [1371](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, [1371](#)
- eQCWWAN_ERR_SWIDCS_END, [1371](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR, [1371](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR, [1371](#)
- eQCWWAN_ERR_SWIDCS_START, [1371](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, [1371](#)
- eQCWWAN_ERR_SWIIM_END, [1371](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, [1371](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_NOT_SIGNED, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_SECBOOT_INVALID_CERT_CHAIN, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_TOO_MANY_FILES, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, [1371](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, [1371](#)
- eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, [1371](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH, [1371](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR, [1371](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE, [1371](#)
- eQCWWAN_ERR_SWIIM_START, [1371](#)
- eQCWWAN_ERR_SWISM_END, [1371](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, [1371](#)
- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, [1371](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, [1371](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, [1371](#)
- eQCWWAN_ERR_SWISMS_START, [1371](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, [1372](#)

- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT-
_OF_PROFILES, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PR-
OFILE_FAMILY, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ON-
LY_FLAG_SET, [1372](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FL-
AG_NOT_SET, [1372](#)
- eWDS_ERR_PROFILE_REG_END, [1372](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_-
FAMILY, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL_HNDL, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL_IDENT, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL_OP, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL_PROFILE_NUM, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL_PROFILE_TYPE, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_IN-
VAL_SUBS_ID, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LE-
N_INVALID, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LI-
B_NOT_INITED, [1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL,
[1372](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_E-
ND, [1372](#)
- qmerrno.h, [1365](#)
- eQCWWANError, [1367](#)
- qm_wds_ds_profile_extended_err_codes, [1371](#)
- qmiSmsMessageList, [541](#)
- messageIndex, [541](#)
- messageTag, [541](#)
- qmiWDSDataBearerTechnology, [541](#)
- currentNetwork, [542](#)
- ratMask, [542](#)
- soMask, [542](#)
- qmuint16
- SwiDataTypes.h, [1433](#)
- qmulong
- SwiDataTypes.h, [1433](#)
- qos.h, [1372](#)
- pack_qos_BindDataPort, [1374](#)
- pack_qos_SLQSQosGetNetworkStatus, [1374](#)
- pack_qos_SLQSQosSmiReadApnExtraParams,
 [1375](#)
- pack_qos_SLQSQosSmiReadDataStats, [1376](#)
- pack_qos_SLQSSetQosEventCallback, [1376](#)
- unpack_qos_BindDataPort, [1377](#)
- unpack_qos_SLQSQosGetNetworkStatus, [1377](#)
- unpack_qos_SLQSQosSmiReadApnExtraParams,
 [1378](#)
- unpack_qos_SLQSQosSmiReadDataStats, [1378](#)
- unpack_qos_SLQSSetQosEventCallback, [1379](#)
- unpack_qos_SLQSSetQosEventCallback_ind,
 [1379](#)
- unpack_qos_SLQSSetQosNWStatusCallback_ind,
 [1380](#)
- unpack_qos_SLQSSetQosPriEventCallback_ind,
 [1380](#)
- unpack_qos_SLQSSetQosStatusCallback_ind,
 [1381](#)
- qos_BindDataPortMuxID_t, [542](#)
- MuxID, [543](#)
- qos_BindDataPortPeripheralEndPointID_t, [543](#)
- EndPointType, [543](#)
- IfaceID, [543](#)
- qos_BindDataPortSIODDataPort_t, [543](#)
- SIODDataPort, [544](#)
- qosDeliveryOrder
- LibPackUMTSQoS, [146](#)
- wds_UMTSMinQoS, [1102](#)
- qosFlow
- unpack_qos_SLQSQosSmiReadDataStats_t, [843](#)
- QosFlowInfo
- unpack_qos_SLQSSetQosEventCallback_ind_t,
 [844](#)
- RAN
- unpack_nas_GetServingNetwork_t, [754](#)
- RAT
- nas_QmiNas3GppNetworkRAT, [275](#)
- RATMask
- currNetworkInfo, [56](#)
- wds_currNetworkInfo, [1083](#)
- RBTTlv
- unpack_ims_SLQSVolPCfgCallBack_ind_t, [703](#)
- RFBandInfoElements, [544](#)
- activeBandClass, [544](#)
- activeChannel, [544](#)
- radioInterface, [544](#)
- unpack_nas_GetRFInfo_t, [751](#)
- RFTlv
- unpack_nas_SetEventReportInd_t, [759](#)
- RMAutoConnect
- pack_dms_SetCustFeature_t, [355](#)
- unpack_dms_GetCustFeature_t, [617](#)
- RPCause
- unpack_sms_SLQSWmsAsyncRawSendCallBack-
 _ind_t, [877](#)
- RPTlv
- NASQmiCbKnasSystemSelPrefInd, [338](#)
- RRTlv
- unpack_nas_SetEventReportInd_t, [759](#)
- RSRPThresListLen
- nas_RSRPThresh, [288](#)
- RSRQThresListLen
- nas_RSRQThresh, [289](#)
- RSSIThresListLen
- nas_RSSIThresh, [290](#)
- RTIDTlv

- unpack_ims_SLQSVolPCfgCallback_ind_t, 703
- RTTlv
 - unpack_ims_SLQSVolPCfgCallback_ind_t, 703
- RX_EC_IO
 - nas_NetworkStat1x, 250
- RX_PWR
 - nas_NetworkStat1x, 250
 - nas_NetworkStatEVDO, 252
- RXBOTlv
 - unpack_wds_SLQSDUNCallInfoCallback_ind_t, 996
- RXChan
 - nas_LTEInfo, 224
- rXDroppedCount
 - unpack_wds_GetPacketStatus_t, 990
- rXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, 990
- rXOkBytesCount
 - unpack_wds_GetPacketStatus_t, 990
- rXPacketErrors
 - unpack_wds_GetPacketStatus_t, 990
- rXPacketOverflows
 - unpack_wds_GetPacketStatus_t, 990
- rXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, 990
- radio
 - unpack_nas_GetSignalStrengths_t, 755
- radio_if
 - pack_nas_SLQSNasGetTxRxInfo_t, 408
- radiolf
 - nas_ecioListElement, 195
 - nas_errorRateListElement, 200
 - nas_rsrqInformation, 288
 - nas_rxSignalStrengthListElement, 293
 - unpack_nas_SLQSNasNetworkRejectCallback_ind_t, 800
- Radiolfaces
 - dms_devCaps, 57
 - unpack_dms_GetDeviceCap_t, 619
 - unpack_dms_GetDeviceCapabilities_t, 621
 - unpack_nas_GetServingNetwork_t, 754
- RadiolfacesSize
 - dms_devCaps, 57
 - unpack_dms_GetDeviceCap_t, 619
 - unpack_nas_GetServingNetwork_t, 754
- radiolfacesSize
 - unpack_dms_GetDeviceCapabilities_t, 621
- radioInterface
 - nas_RFBandInfoElements, 280
 - nas_RfBandInfoExtFormatElements, 281
 - nas_RfBandwidthInfoElements, 283
 - nas_RfDedicatedBandInfoElements, 285
 - nas_RFInfoTlv, 286
 - nas_roamIndList, 287
 - nas_servSystem, 296
 - nas_SignalStrengthTlv, 297
 - nas_timeInfo, 309
 - RFBandInfoElements, 544
- radioInterfaceList
 - NASServingSystemInfo, 341
- radioInterfaceNo
 - NASServingSystemInfo, 341
- radioInterfaceSize
 - nas_RFInfoTlv, 286
- randomizationWindow
 - dms_PSMRandomizationWindowTlv, 71
- range
 - unpack_qos_Port_t, 837
- rankIndicatorTlv
 - unpack_nas_SLQSSwiRandIndicatorCallback_ind_t, 812
- rat
 - nas_CSGID, 187
 - nas_MNRInfo, 246
- RatDMTlv
 - NASQmiCbkNasSystemSelPrefInd, 338
- RatDisabledMask
 - NASRatDisabledMaskTlv, 339
- ratDisabledMask
 - nas_RatDisabledMaskTlv, 279
- RatHandover
 - unpack_imsa_SLQSImsaRatStatusCallback_ind_t, 708
- RatHandoverStatus
 - imsa_RatHandoverStatusInfo, 100
- RatHandoverStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, 378
- ratMask
 - qmiWSDDataBearerTechnology, 542
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- rawHorConfidence
 - pack_loc_SLQSLOCInjectPosition_t, 388
- rawHorUncCircular
 - pack_loc_SLQSLOCInjectPosition_t, 388
- rawLen
 - uim_fileAttributes, 589
- rawValue
 - uim_fileAttributes, 589
- readTransparent
 - pack_uim_ReadTransparent_t, 481
- Reason
 - pack_voice_SLQSVoiceGetCallForwardingStatus_t, 499
 - pack_voice_SLQSVoiceSetCallBarringPassword_t, 505
- reason
 - pack_voice_SLQSVoiceGetCallBarring_t, 498
 - pack_voice_SLQSVoiceSetSUPSService_t, 509
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 847
 - voice_ccSUPSType, 1054
 - voice_redirNumInfo, 1074
- receiptAction
 - sms_routeEntry, 558
- reconfigReqd

- unpack_wds_SLQSSetPacketSrvStatusCallback_ -
t, [1011](#)
- recordCount
 - uim_fileAttributes, [589](#)
- recordSize
 - uim_fileAttributes, [589](#)
- RedirPartyNum
 - voice_arrRedirPartyNum, [1041](#)
- refData
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, [673](#)
- refID
 - pack_cat_CATSendTerminalResponse_t, [350](#)
- refString
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, [673](#)
- ReferencelD
 - cat_AIPhaIdentifierTlv, [49](#)
 - cat_EventIDDDataTlv, [51](#)
- refpn
 - nas_CDMAInfo, [177](#)
- refreshComplete
 - pack_uim_SLQSUIRefreshComplete_t, [487](#)
- refreshEvent
 - unpack_uim_SLQSUIRefreshCallback_Ind_t, [923](#)
- RefreshMode
 - cat_RefreshTlv, [52](#)
- RefreshStage
 - cat_RefreshTlv, [52](#)
- regAction
 - pack_nas_SLQSInitiateNetworkRegistration_t, [401](#)
- RegForeignNID
 - unpack_nas_GetCDMANetworkParameters_t, [745](#)
- RegForeignSID
 - unpack_nas_GetCDMANetworkParameters_t, [745](#)
- RegHomeSID
 - unpack_nas_GetCDMANetworkParameters_t, [746](#)
- regInd
 - unpack_sms_SLQSTransLayerInfoCallback_ind_t, [875](#)
- regPrd
 - nas_AddCDMASysInfo, [170](#)
- regRefresh
 - pack_uim_SLQSUIRefreshRegister_t, [488](#)
- regRejectInfoValid
 - nas_GSMSSysInfo, [208](#)
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [259](#)
 - nas_WCDMASysInfo, [321](#)
- regState
 - nas_servSystem, [296](#)
- RegStatusConfig
 - pack_imsa_SLQSRegisterIMSAIndication_t, [378](#)
- registerFlag
 - uim_registerRefresh, [598](#)
- RegistrationState
 - unpack_nas_GetServingNetwork_t, [754](#)
- registrationState
 - NASServingSystemInfo, [341](#)
- rejCause
 - nas_GSMSSysInfo, [208](#)
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [259](#)
 - nas_WCDMASysInfo, [321](#)
- rejectCause
 - nas_RejectReasonTlv, [279](#)
 - unpack_nas_SLQSNasNetworkRejectCallback_Ind_t, [800](#)
- rejectSrvDomain
 - nas_GSMSSysInfo, [209](#)
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [259](#)
 - nas_WCDMASysInfo, [321](#)
- releaseDate
 - _litefw_FirmwareFileInfo, [36](#)
- reliabilityClass
 - LibPackGPRSRequestedQoS, [105](#)
 - wds_GPRSQoS, [1092](#)
- remPartyNumber
 - voice_remotePartyNum, [1076](#)
- RemotePartyName
 - voice_getAllCallRmtPtyName, [1066](#)
- RemotePartyNum
 - voice_getAllCallRmtPtyNum, [1066](#)
- ReqMitigationLvl
 - unpack_tmd_SLQSTmdGetMitigationLvl_t, [911](#)
- resBerRatio
 - LibPackUMTSQoS, [146](#)
 - wds_UMTSMInQoS, [1102](#)
- ResCode
 - unpack_audio_SLQSGetAudioVoITLBConfig_t, [609](#)
 - unpack_audio_SLQSSetAudioVoITLBConfig_t, [610](#)
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, [673](#)
- resetDelay
 - pack_swidms_SLQSSwiDmsSetHWWatchdog_t, [468](#)
 - swidms_SwiDmsGetHWWatchdog, [573](#)
- resetInfoInd
 - pack_dms_SLQSDmsSwiIndicationRegister_t, [359](#)
- resultcode
 - unpack_swiaavms_SLQSAVMSEventReportInd_t, [883](#)
 - unpack_swiaavms_SLQSAVMSSetSettings_t, [886](#)
 - unpack_swiaavms_SLQSAVMSSetSettings_v2_t, [888](#)
 - unpack_swiaavms_SLQSAVMSSendSelection_t, [889](#)
 - unpack_swiaavms_SLQSAVMSSessionGetInfo_t, [890](#)
 - unpack_swiaavms_SLQSAvmsSetEventReport_t, [890](#)
 - unpack_swiaavms_SLQSAVMSSetSettings_t, [890](#)
 - unpack_swiaavms_SLQSAVMSSetSettings_v2_t, [891](#)
 - unpack_swiaavms_SLQSAVMSStartSession_t, [891](#)

- unpack_swiaavms_SLQSAVMSSStopSession_t, 892
- RetryCount
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 903
- revPolarity
 - voice_lineCtrlInfo, 1068
- revTunneling
 - unpack_wds_GetMobileIPProfile_t, 986
- ReverseMac
 - nas_protocolSubtypeElement, 272
- RfSarState
 - pack_sar_SLQSSetRfSarState_t, 447
- rfbandInfoList
 - unpack_nas_SLQSNasGetRFInfo_t, 795
- RingBkTmr
 - ims_RngBkTmrInfo, 91
- RingTmr
 - ims_RngTmrInfo, 91
- RmTrasnferStaticsReq
 - pack_wds_RMSetTransferStatistics_t, 515
- rmTrasnferStaticsReq, 544
 - bResetStatistics, 545
 - ulMask, 545
- rms.h, 1382
 - pack_rms_GetSMSWake, 1382
 - pack_rms_SetSMSWake, 1383
 - unpack_rms_GetSMSWake, 1383
 - unpack_rms_SetSMSWake, 1383
- RmtPtyNum
 - voice_arrRemotePartyNum, 1042
- roamIndicator
 - nas_roamIndList, 287
- RoamIndicatorVal
 - unpack_nas_SLQSGetServingSystem_t, 773
- roamOrigVoiceSO
 - voice_prefVoiceSO, 1073
- RoamPref
 - nas_RoamPrefTlv, 287
 - NASRoamPreferenceTlv, 339
- roamStatus
 - nas_NR5GSystemInfoTlv, 259
 - nas_sysInfoCommon, 304
- roamStatusValid
 - nas_NR5GSystemInfoTlv, 259
 - nas_sysInfoCommon, 304
- roamTimerValue
 - voice_roamTimer, 1077
- Roaming
 - nas_QmiNas3GppNetworkInfo, 274
 - unpack_nas_GetCDMANetworkParameters_t, 746
 - unpack_nas_GetServingNetwork_t, 754
- roaming
 - unpack_nas_SetRoamingIndicatorCallback_ind_t, 761
- RoamingIndicatorList
 - unpack_nas_SLQSGetServingSystem_t, 773
- routeList
 - sms_setRoutesReq, 562
- routeStorage
 - sms_routeEntry, 558
- RptMSCapability
 - pack_dms_SetIndicationRegister_t, 357
- rptPosData
 - pack_pds_SetEventReportCallback_t, 434
- rptRate
 - nas_LTESigRptCfg, 238
 - nas_LTESigRptConfig, 239
- rscp
 - nas_rxInfo, 291
 - nas_UMTSInfo, 312
 - tdscdmaSigInfoExt, 574
- rsrp
 - lteSSInfo, 166
 - nas_cellParams, 182
 - nas_QmisNasSlqsNasPCIIInfo, 277
 - nas_rxInfo, 291
 - nas_RxSigInfo, 292
 - nas_umtsLTENbrCell, 314
- rsrpRx0
 - nas_QmisNasSlqsNasPCIIInfo, 277
- rsrpRx1
 - nas_QmisNasSlqsNasPCIIInfo, 277
- rsrpIlevel
 - nas_lteRsrpinformation, 236
- rsrq
 - lteSSInfo, 167
 - nas_cellParams, 182
 - nas_QmisNasSlqsNasPCIIInfo, 278
 - nas_rsrqInformation, 288
 - nas_SccRxInfo, 294
 - nas_umtsLTENbrCell, 314
- rsrqDelta
 - nas_SLQSSignalStrengthsIndReq, 299
- rsrqInfo
 - nas_SLQSSignalStrengthsInformation, 300
 - unpack_nas_SLQSGetSignalStrength_t, 776
- rsrqRx0
 - nas_QmisNasSlqsNasPCIIInfo, 278
- rsrqRx1
 - nas_QmisNasSlqsNasPCIIInfo, 278
- rssI
 - cdmaSSInfo, 53
 - hdrSSInfo, 83
 - lteSSInfo, 167
 - nas_cellParams, 182
 - nas_gsmCellInfo, 204
 - tdscdmaSigInfoExt, 574
 - unpack_nas_GetSignalStrengths_t, 755
- rx_bytes
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- rx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- RxByteOKCnt
 - wds_RXBytesOKTlv, 1097

- rxChainIndex
 - nas_RxSigInfo, [292](#)
- RxDropConutTlv
 - unpack_RMTransferStatistics_ind_t, [858](#)
- rxLev
 - nas_GERANInfo, [202](#)
- rxOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, [1004](#)
- RxOkByteCountTlv
 - unpack_RMTransferStatistics_ind_t, [858](#)
- RxOkConutTlv
 - unpack_RMTransferStatistics_ind_t, [859](#)
- rxPower
 - nas_rxInfo, [291](#)
 - nas_RxSigInfo, [292](#)
- RxQFilter
 - unpack_qos_QosFlowInfo_t, [839](#)
- RxQFlowGranted
 - unpack_qos_QosFlowInfo_t, [839](#)
- rxSignalStrength
 - nas_rxSignalStrengthListElement, [293](#)
- rxSignalStrengthDelta
 - nas_SLQSSignalStrengthsIndReq, [299](#)
- rxSignalStrengthInfo
 - nas_SLQSSignalStrengthsInformation, [300](#)
- rxSignalStrengthList
 - unpack_nas_SLQSGetSignalStrength_t, [776](#)
- rxSignalStrengthListLen
 - unpack_nas_SLQSGetSignalStrength_t, [776](#)
- SCI
 - unpack_nas_GetCDMANetworkParameters_t, [746](#)
- SCM
 - unpack_nas_GetCDMANetworkParameters_t, [746](#)
- SCTlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- SDPTlv
 - NASQmiCbkNasSystemSelPrefInd, [338](#)
- SDTlv
 - unpack_ims_SLQSVolPCfgCallBack_ind_t, [703](#)
- SDU_HDR_LEN
 - common.h, [1117](#)
- SFTlv
 - unpack_ims_SLQSSMSCfgCallBack_ind_t, [700](#)
- SHORT
 - SwiDataTypes.h, [1433](#)
- SI
 - voice_calledPartyInfo, [1045](#)
 - voice_callFWExtInfo, [1050](#)
 - voice_callingPartyInfo, [1054](#)
 - voice_redirNumInfo, [1074](#)
- SINTlv
 - unpack_ims_SLQSSMSCfgCallBack_ind_t, [700](#)
- SIODDataPort
 - qos_BindDataPortSIODDataPort_t, [544](#)
- SIPLocalPort
 - ims_SIPPortInfo, [93](#)
- sIntraSearch
 - nas_LTEInfoIntraFreq, [227](#)
- SLQSSSTlv
 - unpack_nas_SetEventReportInd_t, [759](#)
- SMS_CONFIG_LEN
 - sms.h, [1389](#)
- SMS_MAX_SMS_ROUTES
 - sms.h, [1390](#)
- SMS_NUM_OF_SET
 - sms.h, [1390](#)
- SMSC_TYPE_LEN
 - sms.h, [1390](#)
- sMSCAddressInfo, [563](#)
 - data, [563](#)
 - length, [563](#)
- sMSCAddressTlv, [563](#)
 - SMSCInfo, [563](#)
 - TlvPresent, [564](#)
- SMSCInfo
 - sMSCAddressTlv, [563](#)
- SMSC_Tlv
 - unpack_sms_SetNewSMSCallback_ind_t, [863](#)
- sMSEtwmsMessageInfo, [564](#)
 - data, [564](#)
 - length, [564](#)
 - notificationType, [564](#)
- sMSEtwmsMessageTlv, [564](#)
 - EtwsMessageInfo, [565](#)
 - TlvPresent, [565](#)
- sMSEtwmsPlmnInfo, [565](#)
 - mobileCountryCode, [565](#)
 - mobileNetworkCode, [565](#)
- sMSMTMessageInfo, [566](#)
 - messageIndex, [566](#)
 - storageType, [566](#)
- sMSMessageModelInfo, [565](#)
 - messageMode, [566](#)
- sMSOnIMSInfo, [566](#)
 - smsOnIMS, [566](#)
- sMSOnIMSTlv, [567](#)
 - IMSInfo, [567](#)
 - TlvPresent, [567](#)
- SMSSupport
 - pack_dms_SetCustFeature_t, [355](#)
 - unpack_dms_GetCustFeature_t, [617](#)
- sMSTransferRouteMTMessageInfo, [567](#)
 - ackIndicator, [568](#)
 - data, [568](#)
 - format, [568](#)
 - length, [568](#)
 - transactionID, [568](#)
- SNR
 - nas_NetworkStatEVDO, [252](#)
- sNonIntraSearch
 - nas_LTEInfoIntraFreq, [227](#)
- SO
 - nas_NetworkStat1x, [250](#)
- SOMask
 - currNetworkInfo, [56](#)
 - wds_currNetworkInfo, [1083](#)

- SPC_SIZE
 - dms.h, [1130](#)
- SPTlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- sPhyCaAggPcellInfo
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [760](#)
- sPhyCaAggScellDIBw
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [760](#)
- sPhyCaAggScellIndType
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [760](#)
- sPhyCaAggScellIndex
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [760](#)
- sPhyCaAggScellInfo
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [760](#)
- SRTlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- SSInfo
 - unpack_nas_SetServingSystemCallback_ind_t, [762](#)
- sLQSSignalStrengthsInfo
 - nas_SLQSSignalStrengthsTlv, [301](#)
- SSTlv
 - unpack_nas_SetEventReportInd_t, [759](#)
- STTlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- SUPSIInformation
 - unpack_voice_SLQSVoiceSUPSCallback_ind_t, [966](#)
- SUPSType
 - pack_voice_SLQSVoiceManageCalls_t, [503](#)
- SWI_API
 - SwiDataTypes.h, [1433](#)
- SWIWWANCMAPI.h, [1461](#)
- samplesPerBatch
 - loc_accelAcceptReady, [148](#)
 - loc_accelTempAcceptReady, [149](#)
 - loc_gyroAcceptReady, [155](#)
 - loc_gyroTempAcceptReady, [155](#)
- sar.h, [1384](#)
 - pack_sar_SLQSGetRfSarState, [1384](#)
 - pack_sar_SLQSSetRfSarState, [1384](#)
 - unpack_sar_SLQSGetRfSarState, [1385](#)
 - unpack_sar_SLQSSetRfSarState, [1385](#)
 - unpack_sar_SLQSSetRfSarState_t, [1384](#)
- sbas_almanac_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- sbas_ephemeris_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- sbas_health_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- sbas_visible_sv_msk
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- scell_idx
 - nas_PhyCaAggScellIndex, [263](#)
 - NASPhyCaAggScellArray, [332](#)
 - NASPhyCaAggScellIndex, [333](#)
- scell_state
 - nas_PhyCaAggScellIndType, [263](#)
 - nas_PhyCaAggScellInfo, [266](#)
 - NASPhyCaAggScellArray, [332](#)
 - NASPhyCaAggScellIndType, [334](#)
 - NASPhyCaAggScellInfo, [335](#)
- scrAmrEnable
 - ims_EnabSCRAMRInfo, [88](#)
- scrAmrWbEnable
 - ims_EnabSCRAMRWBInfo, [88](#)
- screeningInd
 - voice_connectNumInfo, [1060](#)
- sduErrorRatio
 - LibPackUMTSQoS, [146](#)
 - wds_UMTSMInQoS, [1102](#)
- seDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [1022](#)
- seDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [1022](#)
- sePCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [1022](#)
- sePCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [1022](#)
- secActivate
 - uim_fileAttributes, [589](#)
- secActivateMask
 - uim_fileAttributes, [589](#)
- secChA
 - nas_CDMAChannel, [175](#)
- secChB
 - nas_CDMAChannel, [175](#)
- secDeactivate
 - uim_fileAttributes, [589](#)
- secDeactivateMask
 - uim_fileAttributes, [589](#)
- secIncrease
 - uim_fileAttributes, [589](#)
- secIncreaseMask
 - uim_fileAttributes, [589](#)
- SecProt
 - nas_protocolSubtypeElement, [272](#)
- secRead
 - uim_fileAttributes, [589](#)
- secReadMask
 - uim_fileAttributes, [589](#)
- secWrite
 - uim_fileAttributes, [589](#)
- secWriteMask
 - uim_fileAttributes, [589](#)
- secdns
 - unpack_wds_GetDefaultProfile_t, [979](#)

- unpack_wds_GetDefaultProfileV2_t, 982
- secdnsV6
 - unpack_wds_GetDefaultProfile_t, 979
 - unpack_wds_GetDefaultProfileV2_t, 982
- second
 - nas_timeInfo, 309
 - nas_UniversalTime, 315
- secondaryDNS
 - pack_wds_SetDefaultProfile_t, 517
- SecondaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- SecondaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- secondaryHA
 - unpack_wds_GetMobileIPProfile_t, 986
- SectorIDLen
 - nas_NetworkStatEVDO, 252
- secureBootEnabled
 - unpack_swidms_SLQSSwiDmsGetSecureInfo_t, 894
- selNetwork
 - nas_servSystem, 296
- selected
 - sms_BroadcastConfig, 550
 - sms_CDMABroadcastConfig, 551
- selectedNetwork
 - NASServingSystemInfo, 341
- selection
 - pack_swiaVms_SLQSAVMSSendSelection_t, 463
 - pack_swioMa_SLQSOMADMSendSelection_t, 474
 - pack_swioMa_SLQSOMADMSendSelectionExt_t, 475
- sendStatus
 - unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 877
- sensorData_t, 545
 - flags, 546
 - sensorDataLen, 546
 - timeOfFirstSample, 546
 - timeOffset, 546
 - xAxis, 546
 - yAxis, 546
 - zAxis, 546
- sensorDataLen
 - sensorData_t, 546
- ServerAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- serverAddrTypeMask
 - pack_loc_SLQSLOCGetServer_t, 383
- serverStatus
 - unpack_loc_GetServer_Ind_t, 724
 - unpack_loc_SetServer_Ind_t, 740
- serverType
 - pack_loc_SLQSLOCGetServer_t, 383
 - pack_loc_SLQSLOCSetServer_t, 392
 - unpack_loc_GetServer_Ind_t, 724
- serviceCategory
 - sms_CDMABroadcastConfig, 551
- serviceDomain
 - nas_RejectReasonTlv, 279
 - unpack_nas_SLQSNasNetworkRejectCallback_Ind_t, 800
- ServiceStatusConfig
 - pack_imsa_SLQSRegisterIMSALIndication_t, 378
- servingCellId
 - nas_LTEInfoIntraFreq, 227
- ServingSystem
 - unpack_nas_SLQSGetServingSystem_t, 773
- sessExp
 - ims_SessDurInfo, 92
- sessionControl
 - pack_pds_StartPDSTrackingSessionExt_t, 443
- sessionEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1011
- SessionId
 - pack_loc_Start_t, 394
 - pack_loc_Stop_t, 394
- sessionId
 - unpack_loc_PositionRpt_Ind_t, 736
- sessionInfo
 - pack_uim_ChangePin_t, 480
 - pack_uim_ReadTransparent_t, 481
 - pack_uim_SetPinProtection_t, 482
 - pack_uim_SLQSUIMAuthenticate_t, 483
 - pack_uim_SLQSUIMGetFileAttributes_t, 485
 - pack_uim_SLQSUIMRefreshComplete_t, 487
 - pack_uim_SLQSUIMRefreshGetLastEvent_t, 487
 - pack_uim_SLQSUIMRefreshOK_t, 488
 - pack_uim_SLQSUIMRefreshRegister_t, 488
 - pack_uim_UnblockPin_t, 490
 - pack_uim_VerifyPin_t, 491
- SessionInfoConfig
 - unpack_swioMa_SLQSOMADMAAlertCallback_ind_t, 900
- SessionInfoFota
 - unpack_swioMa_SLQSOMADMAAlertCallback_ind_t, 900
- SessionInfoNotification
 - unpack_swioMa_SLQSOMADMAAlertCallback_ind_t, 900
- sessionOperation
 - pack_pds_StartPDSTrackingSessionExt_t, 443
- sessionResponse
 - unpack_swiaVms_SLQSAVMSSStartSession_t, 892
- sessionServerOption
 - pack_pds_StartPDSTrackingSessionExt_t, 443
- SessionState
 - unpack_swioMa_SLQSOMADMGetSessionInfo_t, 903
- sessionState
 - unpack_swioMa_SLQSOMADMGetSessionInfoExt_t, 906
- SessionStatus
 - unpack_pds_SetEventReport_Ind_t, 827

- sessionStatus
 - unpack_loc_PositionRpt_Ind_t, [736](#)
 - unpack_omaDmNotificationsTlv_t, [820](#)
- SessionType
 - pack_swioama_SLQSOMADMGetSessionInfo_t, [474](#)
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, [903](#)
- sessionType
 - pack_pds_StartPDSTrackingSessionExt_t, [443](#)
 - pack_swiaavms_SLQSAVMSStartSession_t, [467](#)
 - pack_swiaavms_SLQSAVMSStopSession_t, [468](#)
 - pack_swioama_SLQSOMADMCancelSession_t, [473](#)
 - pack_swioama_SLQSOMADMCancelSessionExt_t, [473](#)
 - pack_swioama_SLQSOMADMStartSession_t, [477](#)
 - pack_swioama_SLQSOMADMStartSessionExt_t, [478](#)
 - uim_refreshevent, [597](#)
 - uim_sessionInformation, [599](#)
 - uim_UIMSessionInformation, [603](#)
 - unpack_omaDmFotaTlv_t, [819](#)
- set_fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
- set_fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
- set_function
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
- set_max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
- set_max_time
 - pack_swiloc_SwiLocSetAutoStart_t, [472](#)
- SetupEventList
 - cat_EventListTlv, [52](#)
- Severity
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, [903](#)
- severity
 - unpack_omaDmFotaTlv_t, [819](#)
- shortName
 - nas_PLMNNetworkNameData, [270](#)
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- shortNameCI
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- shortNameEn
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- shortNameLen
 - nas_PLMNNetworkNameData, [270](#)
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- shortNameSB
 - unpack_nas_SLQSGetPLMNName_t, [768](#)
- shortNameSpareBits
 - nas_PLMNNetworkNameData, [270](#)
- sid
 - nas_CDMAInfo, [177](#)
 - nas_sidNid, [296](#)
 - unpack_nas_GetHomeNetwork_t, [749](#)
- SidNid
 - nas_homeSIDNID, [215](#)
- SigCompEn
 - ims_SigCompEnInfo, [93](#)
- SigInd
 - LibPackUMTSReqQoSsigInd, [147](#)
- sigInfo
 - nas_SccRxInfo, [294](#)
- signal
 - voice_signalInfo, [1077](#)
- signalStrength
 - nas_SignalStrengthTlv, [297](#)
- signalStrengthReqMask
 - unpack_nas_SLQSGetSignalStrength_t, [776](#)
- signalType
 - voice_signalInfo, [1077](#)
- simBusy
 - uim_simBusyStatus, [600](#)
- simBusyLength
 - uim_simBusyStatus, [600](#)
- SimCapability
 - dms_devCaps, [57](#)
 - unpack_dms_GetDeviceCap_t, [619](#)
- simCapability
 - unpack_dms_GetDeviceCapabilities_t, [621](#)
- SimVoiceDataCap
 - dms_devSubsVoiceDataList, [64](#)
- sinr
 - hdrSSInfo, [83](#)
 - nas_SLQSSignalStrengthsInformation, [300](#)
 - tdscdmaSigInfoExt, [574](#)
 - unpack_nas_SLQSGetSignalStrength_t, [776](#)
- sinrDelta
 - nas_SLQSSignalStrengthsIndReq, [299](#)
- sinrThresholdList
 - nas_SLQSSignalStrengthsIndReq, [299](#)
- sinrThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, [299](#)
- sku_str
 - unpack_dms_GetFirmwareInfo_t, [628](#)
- skuStr
 - _litefw_FirmwareFileInfo, [36](#)
- slot
 - pack_uim_SLQSUIMPowerDown_t, [486](#)
 - pack_uim_SLQSUIMPowerUp_t, [486](#)
- slot_t, [547](#)
 - bICCID, [547](#)
 - bICCIDLength, [547](#)
 - bLogicalSlot, [547](#)
 - uPhyCardStatus, [547](#)
 - uPhySlotStatus, [547](#)
- slotInf, [548](#)
 - AppStatus, [549](#)
 - cardState, [549](#)
 - errorState, [549](#)
 - numApp, [549](#)
 - upinRetries, [549](#)
 - upinState, [549](#)

- upukRetries, [549](#)
- SlotInfo
 - uim_cardStatus, [583](#)
- slots_t, [549](#)
 - uimSlotStatus, [550](#)
- slotsstatusChange
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, [917](#)
- SlqsProfile3GPP
 - unpackWdsProfileParam, [1032](#)
 - unpackWdsProfileParamV2, [1033](#)
 - wds_profileInfo, [1097](#)
- SlqsProfile3GPP2
 - unpackWdsProfileParam, [1032](#)
 - unpackWdsProfileParamV2, [1033](#)
 - wds_profileInfo, [1097](#)
- sms.h
 - LITEQMI_QMI_CBK_PARAM_NOCHANGE, [1390](#)
 - LITEQMI_QMI_CBK_PARAM_RESET, [1390](#)
 - LITEQMI_QMI_CBK_PARAM_SET, [1390](#)
- sms.h, [1386](#)
 - eqmiCbKsetStatus, [1390](#)
 - MAX_MSE_TWS_MSG, [1389](#)
 - MAX_SMS_LIST_SIZE, [1389](#)
 - pack_sms_GetSMSCAddress, [1390](#)
 - pack_sms_SLQSDDeleteSMS, [1392](#)
 - pack_sms_SLQSGetIndicationRegister, [1392](#)
 - pack_sms_SLQSGetMessageWaiting, [1392](#)
 - pack_sms_SLQSGetSMS, [1393](#)
 - pack_sms_SLQSGetSMSList, [1393](#)
 - pack_sms_SLQSGetSmsBroadcastConfig, [1393](#)
 - pack_sms_SLQSGetTransLayerInfo, [1394](#)
 - pack_sms_SLQSGetTransNWRRegInfo, [1394](#)
 - pack_sms_SLQSMModifySMSStatus, [1394](#)
 - pack_sms_SLQSSendAsyncSMS, [1395](#)
 - pack_sms_SLQSSetIndicationRegister, [1395](#)
 - pack_sms_SLQSSetSmsBroadcastActivation, [1395](#)
 - pack_sms_SLQSSetSmsBroadcastConfig, [1396](#)
 - pack_sms_SLQSSetSmsStorage, [1396](#)
 - pack_sms_SLQSSmsGetMaxStorageSize, [1397](#)
 - pack_sms_SLQSSmsGetMessageProtocol, [1397](#)
 - pack_sms_SLQSSmsSetRoutes, [1397](#)
 - pack_sms_SLQSSwiGetSMSStorage, [1398](#)
 - pack_sms_SaveSMS, [1390](#)
 - pack_sms_SendSMS, [1391](#)
 - pack_sms_SetNewSMSCallback, [1391](#)
 - pack_sms_SetSMSCAddress, [1391](#)
 - SMS_CONFIG_LEN, [1389](#)
 - SMS_MAX_SMS_ROUTES, [1390](#)
 - SMS_NUM_OF_SET, [1390](#)
 - SMSC_TYPE_LEN, [1390](#)
 - unpack_sms_GetSMSCAddress, [1398](#)
 - unpack_sms_SLQSDDeleteSMS, [1400](#)
 - unpack_sms_SLQSGetIndicationRegister, [1400](#)
 - unpack_sms_SLQSGetMessageWaiting, [1401](#)
 - unpack_sms_SLQSGetSMS, [1401](#)
 - unpack_sms_SLQSGetSMSList, [1402](#)
 - unpack_sms_SLQSGetSmsBroadcastConfig, [1401](#)
 - unpack_sms_SLQSGetTransLayerInfo, [1402](#)
 - unpack_sms_SLQSGetTransNWRRegInfo, [1403](#)
 - unpack_sms_SLQSMModifySMSStatus, [1403](#)
 - unpack_sms_SLQSNWRRegInfoCallback_ind, [1403](#)
 - unpack_sms_SLQSSendAsyncSMS, [1404](#)
 - unpack_sms_SLQSSetIndicationRegister, [1404](#)
 - unpack_sms_SLQSSetSmsBroadcastActivation, [1404](#)
 - unpack_sms_SLQSSetSmsBroadcastConfig, [1405](#)
 - unpack_sms_SLQSSetSmsStorage, [1405](#)
 - unpack_sms_SLQSSmsGetMaxStorageSize, [1405](#)
 - unpack_sms_SLQSSmsGetMessageProtocol, [1406](#)
 - unpack_sms_SLQSSmsSetRoutes, [1406](#)
 - unpack_sms_SLQSSwiGetSMSStorage, [1406](#)
 - unpack_sms_SLQSTransLayerInfoCallback_ind, [1407](#)
 - unpack_sms_SLQSWmsAsyncRawSendCallBack_ind, [1407](#)
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind, [1408](#)
 - unpack_sms_SLQSWmsMessageWaitingCallBack_ind, [1408](#)
 - unpack_sms_SaveSMS, [1398](#)
 - unpack_sms_SendSMS, [1399](#)
 - unpack_sms_SetNewSMSCallback, [1399](#)
 - unpack_sms_SetNewSMSCallback_ind, [1399](#)
 - unpack_sms_SetSMSCAddress, [1400](#)
- sms_BroadcastConfig, [550](#)
 - fromServiceId, [550](#)
 - selected, [550](#)
 - toServiceId, [550](#)
- sms_CDMABroadcastConfig, [550](#)
 - language, [551](#)
 - selected, [551](#)
 - serviceCategory, [551](#)
- sms_getIndicationReg, [551](#)
 - pRegCallStatInfoEvt, [552](#)
 - pRegTransLayerInfoEvt, [552](#)
 - pRegTransNWRRegInfoEvt, [552](#)
- sms_getMsgWaitingInfo, [552](#)
 - msgWaitInfo, [552](#)
 - numInstances, [552](#)
- sms_getTransLayerInfo, [552](#)
 - pRegInd, [553](#)
 - pTransLayerInfo, [553](#)
- sms_getTransNWRRegInfo, [553](#)
 - pRegStatus, [553](#)
- sms_maxStorageSizeReq, [554](#)
 - pMessageMode, [554](#)
 - storageType, [554](#)
- sms_maxStorageSizeResp, [554](#)
 - freeSlots, [555](#)
 - maxStorageSize, [555](#)
- sms_messageWaitingInfoContent, [555](#)
 - activeInd, [555](#)

- msgCount, 555
- msgType, 555
- sms_msgProtocolResp, 556
 - msgProtocol, 556
- sms_qaQmi3GPP2BroadcastCfgInfo, 556
 - activated_ind, 557
 - CDMABroadcastConfig, 557
 - num_instances, 557
- sms_qaQmi3GPPBroadcastCfgInfo, 557
 - activated_ind, 557
 - broadcastConfig, 557
 - num_instances, 557
- sms_routeEntry, 557
 - messageClass, 558
 - messageType, 558
 - receiptAction, 558
 - routeStorage, 558
- sms_sendAsyncsmsParams, 558
 - messageFormat, 560
 - messageSize, 560
 - pFollowOnDC, 560
 - pForceOnDC, 560
 - pLinktimer, 560
 - pMessage, 560
 - pRetryMessage, 560
 - pRetryMessageId, 560
 - pServiceOption, 560
 - pSmsOnIms, 560
 - pUserData, 560
- sms_setIndicationReg, 561
 - pRegCallStatInfoEvt, 561
 - pRegTransLayerInfoEvt, 561
 - pRegTransNWRRegInfoEvt, 561
- sms_setRoutesReq, 561
 - numOfRoutes, 562
 - pTransferStatusReport, 562
 - routeList, 562
- sms_transLayerInfo, 562
 - TransCap, 562
 - TransType, 563
- smsFormat
 - ims_SMSFmtInfo, 94
- smsOnIMS
 - sMSOnIMSInfo, 566
- SmsRat
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 710
- SmsRatVal
 - imsa_SmsRatInfo, 100
- SmsService
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 710
- SmsServiceRat
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, 706
- SmsServiceStatus
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, 706
- smsStorage
 - pack_sms_SLQSSetSmsStorage_t, 455
- SmsSvcStatus
 - imsa_SmsSvcStatusInfo, 101
- smsolPNW
 - ims_SMSolPNwInfo, 95
- snr
 - loc_satelliteInfo, 162
 - lteSSInfo, 167
 - nas_SccRxInfo, 294
- snrlevel
 - nas_lteSnrinformation, 240
- soMask
 - qmiWSDDataBearerTechnology, 542
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- Source
 - unpack_swioima_SLQSOMADMGetSessionInfo_t, 903
- source
 - altSrcInfo_t, 39
 - unpack_dms_GetNetworkTime_t, 635
 - unpack_dms_GetNetworkTimeV2_t, 636
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 654
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 655
- sourceIPMask
 - LibPackTFTIDParams, 144
- SourceLength
 - unpack_swioima_SLQSOMADMGetSessionInfo_t, 903
- SourceRAT
 - imsa_RatHandoverStatusInfo, 100
- spc
 - pack_dms_ResetToFactoryDefaults_t, 352
 - pack_nas_SetACCOLC_t, 395
 - pack_wds_SetMobileIPProfile_t, 521
- spn
 - nas_serviceProviderName, 294
 - unpack_nas_SLQSGetPLMNName_t, 768
- spnEncoding
 - unpack_nas_SLQSGetPLMNName_t, 768
- spnLength
 - nas_serviceProviderName, 294
 - unpack_nas_SLQSGetPLMNName_t, 769
- srcPortRangeEnd
 - LibPackTFTIDParams, 144
- srcPortRangeStart
 - LibPackTFTIDParams, 144
- srvCapValid
 - nas_NR5GSystemInfoTlv, 259
- srvCapability
 - nas_detailSvcInfo, 192
 - nas_sysInfoCommon, 304
- srvCapabilityValid
 - nas_sysInfoCommon, 304
- srvDomain
 - nas_NR5GSystemInfoTlv, 259
 - nas_sysInfoCommon, 304
- SrvDomainPref

- nas_SrvDomainPrefTlv, 301
- NASServDomainPrefTlv, 339
- srvDomainValid
 - nas_NR5GSystemInfoTlv, 259
 - nas_sysInfoCommon, 304
- srvOption
 - voice_arrSvcOption, 1042
- srvStatus
 - nas_detailSvcInfo, 192
 - nas_GSMSrvStatusInfo, 206
 - nas_NR5GSerStatTlv, 255
 - nas_SrvStatusInfo, 302
- srvcapability
 - nas_NR5GSystemInfoTlv, 259
- srxlev
 - nas_cellParams, 182
 - nas_gsmCellInfo, 204
 - nas_umtsLTENbrCell, 314
 - nas_wcdmaCellInfo, 316
- stage
 - uim_refreshevent, 597
- State
 - nas_NetworkStat1x, 250
 - nas_NetworkStatEVDO, 252
- state
 - pack_loc_SLQSLOCSetCradleMountConfig_t, 391
 - unpack_dms_GetActivationState_t, 612
 - unpack_dms_SLQSUIMGetState_t, 678
 - unpack_omaDmConfigTlv_t, 817
 - unpack_omaDmFotaTlv_t, 819
 - unpack_qos_QosFlowInfoState_t, 839
- statmask
 - pack_wds_GetPacketStatus_t, 514
- StatsMask
 - wds_transferStatInd, 1098
- statsMask
 - wds_TrStatInd, 1099
- StatsPeriod
 - wds_transferStatInd, 1098
- statsPeriod
 - wds_TrStatInd, 1099
- Status
 - unpack_loc_GetOpMode_Ind_t, 723
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 903
- status
 - nas_lteEARFCN, 221
 - nas_ltePCI, 235
 - nas_wcdmaUARFCN, 322
 - pack_sms_SetNewSMSCallback_t, 449
 - unpack_loc_BestAvailPos_Ind_t, 717
 - unpack_loc_DeleteAssistData_Ind_t, 719
 - unpack_loc_InjectPositionCallback_Ind_t, 726
 - unpack_loc_InjectUTCTimeCallback_Ind_t, 729
 - unpack_loc_SetExtPowerConfig_Ind_t, 737
 - unpack_loc_SetOperationMode_Ind_t, 739
 - unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 845
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 847
 - unpack_swima_SLQSOMADMGetSessionInfo_Ext_t, 906
- storageIndex
 - FMSImageIdElement, 80
- storageType
 - pack_sms_SaveSMS_t, 448
 - pack_sms_SLQSDeleteSMS_t, 450
 - pack_sms_SLQSGetSMS_t, 451
 - pack_sms_SLQSGetSMSList_t, 452
 - pack_sms_SLQSModifySMSStatus_t, 453
 - sms_maxStorageSizeReq, 554
 - sMSMTMessageInfo, 566
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 878
- Streaming Download Protocol (lte-fw), 32
- String
 - unpack_dms_GetDeviceHardwareRev_t, 624
 - unpack_dms_GetDeviceMfr_t, 625
 - unpack_dms_GetFSN_t, 631
 - unpack_dms_UIMGetICCID_t, 681
- stringSize
 - unpack_dms_GetDeviceHardwareRev_t, 624
 - unpack_dms_GetDeviceMfr_t, 625
 - unpack_dms_UIMGetICCID_t, 681
- subAddr
 - voice_calledPartySubAdd, 1046
- subAddrLen
 - voice_calledPartySubAdd, 1046
- subAddrType
 - voice_calledPartySubAdd, 1046
- subnetMask
 - unpack_qos_IPv4Addr_t, 835
- SubnetMaskV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- 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, 494
- SubsVoiceDataCap
 - dms_devSubsVoiceDataList, 64

- SubsVoiceDataCapLen
 - dms_devSubsVoiceDataCaps, [64](#)
- SubsVoiceDataList
 - dms_devSubsVoiceDataCaps, [64](#)
- subscrTmr
 - ims_SubscrTmrInfo, [95](#)
- SupUSBComps
 - unpack_dms_GetUSBComp_t, [642](#)
- SuppOA
 - voice_CUGInfo, [1060](#)
- SuppPrefCUG
 - voice_CUGInfo, [1060](#)
- supportedLteBandLen
 - dms_LteBandsSupport, [65](#)
- svInfoMask
 - loc_satelliteInfo, [162](#)
- svListLen
 - loc_satelliteInfo, [162](#)
- svStatus
 - loc_satelliteInfo, [162](#)
- svc
 - pack_qmi_t, [444](#)
- SvcClass
 - voice_callFWExtInfo, [1050](#)
 - voice_callFWInfo, [1051](#)
- SvcStatus
 - voice_callFWExtInfo, [1050](#)
 - voice_callFWInfo, [1051](#)
- svcType
 - voice_ccSUPSType, [1055](#)
 - voice_SUPSInfo, [1078](#)
- sw1
 - uim_cardResult, [582](#)
- sw2
 - uim_cardResult, [582](#)
- swVerString
 - pack_dms_SLQSSwiSetHostDevInfo_t, [362](#)
 - unpack_dms_SLQSSwiGetHostDevInfo_t, [674](#)
- swi_uint256_clear_bit
 - switype_256bit.h, [1460](#)
- swi_uint256_get_bit
 - switype_256bit.h, [1460](#)
- swi_uint256_print_mask
 - switype_256bit.h, [1460](#)
- swi_uint256_set_bit
 - switype_256bit.h, [1460](#)
- swi_uint256_t, [568](#)
 - word, [568](#)
- SwiDataTypes.h, [1432](#)
 - BOOL, [1433](#)
 - BYTE, [1433](#)
 - CHAR, [1433](#)
 - FLOAT, [1433](#)
 - INT32, [1433](#)
 - INT8, [1433](#)
 - LPCSTR, [1433](#)
 - qmuint16, [1433](#)
 - qmulong, [1433](#)
- SHORT, [1433](#)
- SWI_API, [1433](#)
- ULONG, [1433](#)
- ULONGLONG, [1433](#)
- UNUSEDPARAM, [1433](#)
- USHORT, [1433](#)
- WORD, [1433](#)
- swiaudio.h, [1408](#)
 - pack_swiaudio_SLQSSetM2MAVMute, [1411](#)
 - pack_swiaudio_SLQSSetM2MAudioProfile, [1410](#)
 - pack_swiaudio_SLQSSetM2MAudioVolume, [1410](#)
 - pack_swiaudio_SLQSSetM2MSpkrGain, [1411](#)
 - pack_swiaudio_SLQSSetM2MAVMute, [1413](#)
 - pack_swiaudio_SLQSSetM2MAudioAVCFG, [1412](#)
 - pack_swiaudio_SLQSSetM2MAudioLPBK, [1412](#)
 - pack_swiaudio_SLQSSetM2MAudioNVDef, [1412](#)
 - pack_swiaudio_SLQSSetM2MAudioProfile, [1413](#)
 - pack_swiaudio_SLQSSetM2MAudioVolume, [1413](#)
 - pack_swiaudio_SLQSSetM2MSpkrGain, [1414](#)
 - unpack_swiaudio_SLQSSetM2MAVMute, [1415](#)
 - unpack_swiaudio_SLQSSetM2MAudioProfile, [1414](#)
 - unpack_swiaudio_SLQSSetM2MAudioVolume, [1415](#)
 - unpack_swiaudio_SLQSSetM2MSpkrGain, [1415](#)
 - unpack_swiaudio_SLQSSetM2MAVMute, [1417](#)
 - unpack_swiaudio_SLQSSetM2MAVMute_t, [1410](#)
 - unpack_swiaudio_SLQSSetM2MAudioAVCFG, [1416](#)
 - unpack_swiaudio_SLQSSetM2MAudioAVCFG_t, [1410](#)
 - unpack_swiaudio_SLQSSetM2MAudioLPBK, [1416](#)
 - unpack_swiaudio_SLQSSetM2MAudioLPBK_t, [1410](#)
 - unpack_swiaudio_SLQSSetM2MAudioNVDef, [1416](#)
 - unpack_swiaudio_SLQSSetM2MAudioNVDef_t, [1410](#)
 - unpack_swiaudio_SLQSSetM2MAudioProfile, [1417](#)
 - unpack_swiaudio_SLQSSetM2MAudioProfile_t, [1410](#)
 - unpack_swiaudio_SLQSSetM2MAudioVolume, [1417](#)
 - unpack_swiaudio_SLQSSetM2MAudioVolume_t, [1410](#)
 - unpack_swiaudio_SLQSSetM2MSpkrGain, [1418](#)
 - unpack_swiaudio_SLQSSetM2MSpkrGain_t, [1410](#)
- swiaudio_PCMparams, [568](#)
 - iFaceTab, [569](#)
 - iFaceTabLen, [569](#)
- swiavms.h, [1418](#)
 - pack_swiavms_SLQSAVMSGetSettings, [1420](#)
 - pack_swiavms_SLQSAVMSGetSettings_v2, [1421](#)
 - pack_swiavms_SLQSAVMSSendSelection, [1422](#)
 - pack_swiavms_SLQSAVMSSessionGetInfo, [1422](#)
 - pack_swiavms_SLQSAVMSSetSettings, [1423](#)
 - pack_swiavms_SLQSAVMSSetSettings_v2, [1424](#)

- pack_swiaavms_SLQSAVMSSetSettingsNoAuto-
RebootField, [1425](#)
- pack_swiaavms_SLQSAVMSStartSession, [1425](#)
- pack_swiaavms_SLQSAVMSStopSession, [1426](#)
- pack_swiaavms_SLQSAvmsSetEventReport, [1423](#)
- unpack_swiaavms_SLQSAVMSEventReportInd,
[1427](#)
- unpack_swiaavms_SLQSAVMSGetSettings, [1427](#)
- unpack_swiaavms_SLQSAVMSGetSettings_v2,
[1427](#)
- unpack_swiaavms_SLQSAVMSSendSelection,
[1428](#)
- unpack_swiaavms_SLQSAVMSSessionGetInfo,
[1428](#)
- unpack_swiaavms_SLQSAVMSSetSettings, [1429](#)
- unpack_swiaavms_SLQSAVMSSetSettings_v2,
[1430](#)
- unpack_swiaavms_SLQSAVMSStartSession, [1430](#)
- unpack_swiaavms_SLQSAVMSStopSession, [1431](#)
- unpack_swiaavms_SLQSAVMSStopSession_avc2,
[1431](#)
- unpack_swiaavms_SLQSAVMSStopSession_avc2-
_t, [1420](#)
- unpack_swiaavms_SLQSAvmsSetEventReport,
[1429](#)
- swidms.h, [1433](#)
 - pack_swidms_SLQSSwiDmsGetHWWatchdog,
[1435](#)
 - pack_swidms_SLQSSwiDmsGetMTU, [1435](#)
 - pack_swidms_SLQSSwiDmsGetSecureInfo, [1435](#)
 - pack_swidms_SLQSSwiDmsGetUsbComp, [1436](#)
 - pack_swidms_SLQSSwiDmsGetUsbNetNum,
[1436](#)
 - pack_swidms_SLQSSwiDmsSetHWWatchdog,
[1436](#)
 - pack_swidms_SLQSSwiDmsSetMTU, [1437](#)
 - pack_swidms_SLQSSwiDmsSetUsbComp, [1437](#)
 - pack_swidms_SLQSSwiDmsSetUsbNetNum,
[1437](#)
 - unpack_swidms_SLQSSwiDmsGetHWWatchdog,
[1438](#)
 - unpack_swidms_SLQSSwiDmsGetMTU, [1438](#)
 - unpack_swidms_SLQSSwiDmsGetSecureInfo,
[1438](#)
 - unpack_swidms_SLQSSwiDmsGetUsbComp,
[1439](#)
 - unpack_swidms_SLQSSwiDmsGetUsbNetNum,
[1439](#)
 - unpack_swidms_SLQSSwiDmsSetHWWatchdog,
[1440](#)
 - unpack_swidms_SLQSSwiDmsSetMTU, [1440](#)
 - unpack_swidms_SLQSSwiDmsSetUsbComp,
[1440](#)
 - unpack_swidms_SLQSSwiDmsSetUsbNetNum,
[1441](#)
 - unpack_swidms_SLQSSwiDmsSetUsbNetNum_t,
[1435](#)
- swidms_SwiDmsGetHWWatchdog, [572](#)
- count, [573](#)
- enable, [573](#)
- resetDelay, [573](#)
- timeout, [573](#)
- swidms_ehrpdMTUSizeTlv, [569](#)
 - ehrpdmTUSize, [569](#)
 - TlvPresent, [569](#)
- swidms_hrpdmTUSizeTlv, [569](#)
 - hrpdmTUSize, [570](#)
 - TlvPresent, [570](#)
- swidms_ifaceCfgTlv, [570](#)
 - CfgValue, [571](#)
 - CurrentCfgType, [571](#)
 - TlvPresent, [571](#)
- swidms_mtuSize3gppTlv, [571](#)
 - MTUSize3gpp, [572](#)
 - TlvPresent, [572](#)
- swidms_supportedIntBitmaskTlv, [572](#)
 - TlvPresent, [572](#)
 - ValidBitmasks, [572](#)
- swidms_usbMTUSizeTlv, [573](#)
 - TlvPresent, [573](#)
 - UsbMTUSize, [573](#)
- swiloc.h, [1441](#)
 - pack_swiloc_SwiLocGetAutoStart, [1442](#)
 - pack_swiloc_SwiLocSetAutoStart, [1442](#)
 - unpack_swiloc_SwiLocGetAutoStart, [1442](#)
 - unpack_swiloc_SwiLocSetAutoStart, [1443](#)
 - unpack_swiloc_SwiLocSetAutoStart_t, [1441](#)
- swioma.h, [1443](#)
 - pack_swioma_SLQSOMADMAAlertCallback, [1444](#)
 - pack_swioma_SLQSOMADMCancelSession, [1445](#)
 - pack_swioma_SLQSOMADMGetSessionInfo,
[1446](#)
 - pack_swioma_SLQSOMADMGetSettings, [1446](#)
 - pack_swioma_SLQSOMADMSendSelection, [1447](#)
 - pack_swioma_SLQSOMADMSetSettings, [1447](#)
 - pack_swioma_SLQSOMADMStartSession, [1448](#)
 - unpack_swioma_SLQSOMADMAAlertCallback,
[1449](#)
 - unpack_swioma_SLQSOMADMAAlertCallback_ind,
[1449](#)
 - unpack_swioma_SLQSOMADMAAlertCallback_t,
[1444](#)
 - unpack_swioma_SLQSOMADMCancelSession,
[1450](#)
 - unpack_swioma_SLQSOMADMCancelSession_t,
[1444](#)
 - unpack_swioma_SLQSOMADMGetSessionInfo,
[1450](#)
 - unpack_swioma_SLQSOMADMGetSettings, [1451](#)
 - unpack_swioma_SLQSOMADMSendSelection,
[1451](#)
 - unpack_swioma_SLQSOMADMSendSelection_t,
[1444](#)
 - unpack_swioma_SLQSOMADMSetSettings, [1452](#)
 - unpack_swioma_SLQSOMADMSetSettings_t,
[1444](#)

- unpack_swima_SLQSOMADMStartSession, 1452
- swiomaext.h, 1453
 - pack_swima_SLQSOMADMCancelSessionExt, 1454
 - pack_swima_SLQSOMADMGetSessionInfoExt, 1454
 - pack_swima_SLQSOMADMSendSelectionExt, 1455
 - pack_swima_SLQSOMADMSetSettingsExt, 1455
 - pack_swima_SLQSOMADMStartSessionExt, 1456
 - unpack_swima_SLQSOMADMCancelSession-Ext, 1457
 - unpack_swima_SLQSOMADMCancelSession-Ext_t, 1453
 - unpack_swima_SLQSOMADMGetSessionInfo-Ext, 1457
 - unpack_swima_SLQSOMADMSendSelectionExt, 1458
 - unpack_swima_SLQSOMADMSendSelectionExt_t, 1454
 - unpack_swima_SLQSOMADMSetSettingsExt, 1458
 - unpack_swima_SLQSOMADMSetSettingsExt_t, 1454
 - unpack_swima_SLQSOMADMStartSessionExt, 1459
 - unpack_swima_SLQSOMADMStartSessionExt_t, 1454
- switchOption
 - pack_voice_SLQSVoiceALSSetLineSwitching_t, 493
- switype_256bit.h, 1459
 - swi_uint256_clear_bit, 1460
 - swi_uint256_get_bit, 1460
 - swi_uint256_print_mask, 1460
 - swi_uint256_set_bit, 1460
- sysForbidden
 - nas_NR5GSystemInfoTlv, 259
- sysForbiddenValid
 - nas_NR5GSystemInfoTlv, 259
- sysInfoCDMA
 - nas_CDMA SysInfo, 181
- sysInfoGSM
 - nas_GSM SysInfo, 209
- sysInfoHDR
 - nas_HDR SysInfo, 215
- sysInfoLTE
 - nas_LTE SysInfo, 243
- sysInfoWCDMA
 - nas_WCDMA SysInfo, 321
- system
 - loc_satelliteInfo, 162
 - loc_SV, 164
- systemDiscontinuities
 - pack_pds_PDSInjectTimeReference_t, 433
- SystemID
 - unpack_nas_SLQSGetServingSystem_t, 773
- systemID
 - nas_CDMA SysInfo, 181
- systemMode
 - nas_CommInfo, 186
- systemTime
 - pack_pds_PDSInjectTimeReference_t, 433
- szAPN
 - PackSwiAvmsSetSettingsAPNInfo, 538
- szAlertMsg
 - UnpackSwiAvmsEventReportConfig, 1026
- szCarrier_str
 - _litew_FirmwareInfo_, 37
- szCarrierPriversion_str
 - _litew_FirmwareInfo_, 37
- szDescription
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1025
- szFwversion_str
 - _litew_FirmwareInfo_, 37
- szModelid_str
 - _litew_FirmwareInfo_, 37
- szName
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1025
- szPWD
 - PackSwiAvmsSetSettingsAPNInfo, 538
- szPackageid_str
 - _litew_FirmwareInfo_, 37
- szPartno_str
 - _litew_FirmwarePartNo_, 38
- szSku_str
 - _litew_FirmwareInfo_, 37
- szUname
 - PackSwiAvmsSetSettingsAPNInfo, 538
- szVersion
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1025
- t3396_apn
 - unpack_nas_SLQSNasTimerCallback_ind_t, 805
- t3396_plmn_id
 - unpack_nas_SLQSNasTimerCallback_ind_t, 805
- t3396_val
 - unpack_nas_SLQSNasTimerCallback_ind_t, 805
- TCPDStPort
 - unpack_qos_swiQosFilter_t, 850
- TCPSrcPort
 - unpack_qos_swiQosFilter_t, 850
- TDSCDMAECIOThreshListLen
 - nas_TDSCDMAECIOThresh, 305
- TDSCDMARSCPTThreshListLen
 - nas_TDSCDMARSCPTThresh, 305
- TDSCDMARSSIThreshListLen
 - nas_TDSCDMARSSIThresh, 306
- TDSCDMASINRThreshListLen
 - nas_TDSCDMASINRThresh, 307
- TLVPresent
 - dms_LteBandsSupport, 65

- TMD_MAX_DEV_LIST
 - tmd.h, [1461](#)
- TMTlv
 - unpack_ims_SLQSRegMgrCfgCallBack_ind_t, [695](#)
- TPCause
 - unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, [877](#)
- TRMessageTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [863](#)
- TT1Tlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- TT2Tlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- TTfTlv
 - unpack_ims_SLQSSIPCfgCallBack_ind_t, [699](#)
- TX_PWR
 - nas_NetworkStat1x, [250](#)
- TXBOTlv
 - unpack_wds_SLQSDUNCallInfoCallBack_ind_t, [996](#)
- TXChan
 - nas_LTEInfo, [224](#)
- tXDroppedCount
 - unpack_wds_GetPacketStatus_t, [990](#)
- tXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, [990](#)
- tXOkBytesCount
 - unpack_wds_GetPacketStatus_t, [990](#)
- tXPacketErrors
 - unpack_wds_GetPacketStatus_t, [990](#)
- tXPacketOverflows
 - unpack_wds_GetPacketStatus_t, [990](#)
- tXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, [990](#)
- Tables, [33](#)
- tac
 - nas_LTEInfoIntraFreq, [227](#)
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [259](#)
- tacValid
 - nas_LTESysInfo, [243](#)
 - nas_NR5GSystemInfoTlv, [259](#)
- TargetRAT
 - imsa_RatHandoverStatusInfo, [100](#)
- TdsBandCapability
 - unpack_dms_SLQSGetBandCapability_t, [661](#)
 - unpack_dms_SLQSGetBandCapabilityExt_t, [665](#)
- tdscdmaSigInfoExt, [574](#)
 - ecio, [574](#)
 - rscp, [574](#)
 - rssi, [574](#)
 - sinr, [574](#)
- techName
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [1011](#)
- Technology
 - nas_DeviceConfigDetail, [193](#)
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- TechnologyPref
 - pack_nas_SetNetworkPreference_t, [398](#)
- tempData_t, [574](#)
 - temperature, [575](#)
 - temperatureDataLen, [575](#)
 - timeOfFirstSample, [575](#)
 - timeOffset, [575](#)
 - timeSource, [575](#)
- TempStat
 - dms_TemperatureTlv, [72](#)
- TempTlv
 - unpack_dms_SwiEventReportCallBack_ind_t, [679](#)
- Temperature
 - dms_TemperatureTlv, [72](#)
- temperature
 - nas_CommInfo, [186](#)
 - tempData_t, [575](#)
- temperatureDataLen
 - tempData_t, [575](#)
- threshGsmHigh
 - nas_lteGsmCellInfo, [222](#)
- threshGsmLow
 - nas_lteGsmCellInfo, [222](#)
- threshServingLow
 - nas_LTEInfoIntraFreq, [227](#)
- threshXHigh
 - nas_infoInterFreq, [216](#)
- threshXLow
 - nas_infoInterFreq, [217](#)
- threshXhigh
 - nas_lteWcdmaCellInfo, [244](#)
- threshXlow
 - nas_lteWcdmaCellInfo, [244](#)
- ThrottleTimer
 - LibPackPDNThrottleTimer, [106](#)
- Time
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, [903](#)
- time
 - NASTimeInfoTlv, [341](#)
- Time_uncert_ms
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- timeBase
 - pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, [438](#)
- TimeLength
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, [903](#)
- timeMsec
 - pack_loc_SLQSLOCInjectUTCTime_t, [390](#)
 - pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, [438](#)
- timeOfFirstSample
 - sensorData_t, [546](#)
 - tempData_t, [575](#)
- timeOffset
 - sensorData_t, [546](#)
 - tempData_t, [575](#)

- timeSource
 - tempData_t, 575
- TimeStmp_gps_week
 - unpack_pds_SLQSGetGPSSStateInfo_t, 833
- TimeStmp_tow_ms
 - unpack_pds_SLQSGetGPSSStateInfo_t, 833
- timeSyncRefCounter
 - unpack_loc_EventTimeSyncCallback_Ind_t, 721
- timeTlv
 - NASQmiCbkNasSwiOTAMessageInd, 336
- timeUncMsec
 - pack_loc_SLQSLOCInjectUTCTime_t, 390
 - pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 439
- timeZone
 - nas_timeInfo, 309
- timeout
 - pack_pds_SetPDSDDefaults_t, 435
 - pack_qmi_t, 444
 - pack_swidms_SLQSSwiDmsSetHWWatchdog_t, 468
 - swidms_SwiDmsGetHWWatchdog, 573
- Timers
 - PackSwiAvmsSetSettingsConnectionRetryTimers, 538
 - PackSwiAVMSSettingsConnectionRetryTimers, 540
- timestamp
 - unpack_dms_GetNetworkTime_t, 635
 - unpack_dms_GetNetworkTimeV2_t, 636
- timestampAge
 - pack_loc_SLQSLOCInjectPosition_t, 388
- timestampUtc
 - pack_loc_SLQSLOCInjectPosition_t, 388
- timingAdvance
 - nas_GERANInfo, 202
- TlvPresenceMask
 - qmTlvResult, 542
- 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_SMSolPNwInfo, 95
 - ims_SubscrTmrInfo, 95
 - ims_TmrT1Info, 96
 - ims_TmrT2Info, 96
 - ims_TmrTfInfo, 97
 - imsa_IMSFailErrCodeTlv, 97
 - imsa_IMSRegStatusErrorCodeInfo, 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_IPv4Info, 157
 - loc_IPv6Info, 158
 - loc_urlAddr, 165
 - messageModeTlv, 167
 - nas_AcqOrderPrefTlv, 169
 - nas_BandPrefInfoTlv, 171
 - nas_BandPrefTlv, 173
 - nas_CiotAcqOrderPrefTlv, 184
 - nas_CiotLteOpModePrefTlv, 184
 - nas_CsgId, 187
 - nas_EdrxCiotLteMode, 196
 - nas_EdrxCycleLength, 196
 - nas_EdrxEnableType, 197
 - nas_EdrxPagingTimeWindow, 197
 - nas_EdrxRatType, 198
 - nas_EmerModeTlv, 199
 - nas_ForbiddenNetworks3GPP, 201
 - nas_GWAcqOrderPrefTlv, 209
 - nas_LTEBandPrefTlv, 220
 - nas_LteCiotOpModeTlv, 220

- nas_LteM1BandPrefTlv, [231](#)
- nas_LteNb1BandPrefTlv, [233](#)
- nas_LTEOperationMode, [233](#)
- nas_LteOpMode, [234](#)
- nas_LteOpModeTlv, [235](#)
- nas_ModePrefTlv, [246](#)
- nas_NetSelPrefTlv, [248](#)
- nas_NR5GCellStatusTlv, [254](#)
- nas_NR5GSerStatTlv, [255](#)
- nas_NR5GSystemInfoTlv, [259](#)
- nas_PhyCaAggPcellInfo, [262](#)
- nas_PhyCaAggScellIDIBw, [262](#)
- nas_PhyCaAggScellIndex, [263](#)
- nas_PhyCaAggScellIndType, [263](#)
- nas_PhyCaAggScellInfo, [266](#)
- nas_PlmnID, [268](#)
- nas_PRLPrefTlv, [271](#)
- nas_RankIndicatorTlv, [278](#)
- nas_RatDisabledMaskTlv, [279](#)
- nas_RejectReasonTlv, [279](#)
- nas_RfBandInfoExtFormat, [281](#)
- nas_RfBandwidthInfo, [283](#)
- nas_RfDedicatedBandInfo, [284](#)
- nas_RfInfoTlv, [286](#)
- nas_RoamPrefTlv, [287](#)
- nas_SccRxInfo, [294](#)
- nas_SignalStrengthTlv, [297](#)
- nas_SLQSSignalStrengthsTlv, [301](#)
- nas_SrvDomainPrefTlv, [301](#)
- nas_timeInfo, [309](#)
- NASAcqOrderPrefTlv, [322](#)
- NASBandPreferenceTlv, [322](#)
- NASCIotAcqOrderPrefTlv, [323](#)
- NASCIotLteOpModePrefTlv, [323](#)
- NASEmergencyModeTlv, [324](#)
- NASGWAcqOrderPrefTlv, [325](#)
- NASLTEBandPreferenceTlv, [326](#)
- NASLteM1BandPrefTlv, [326](#)
- NASLteNasReleaseInfoTlv, [327](#)
- NASLteNB1BandPrefTlv, [327](#)
- NASModePreferenceTlv, [328](#)
- NASNetSelPreferenceTlv, [328](#)
- NASNr5gBandPrefTlv, [329](#)
- NASOTAMessageTlv, [330](#)
- NASPhyCaAggPcellInfo, [330](#)
- NASPhyCaAggScellArray, [332](#)
- NASPhyCaAggScellIDIBw, [332](#)
- NASPhyCaAggScellIndex, [333](#)
- NASPhyCaAggScellIndType, [334](#)
- NASPhyCaAggScellInfo, [335](#)
- NASPRPreferenceTlv, [335](#)
- NASRatDisabledMaskTlv, [339](#)
- NASRoamPreferenceTlv, [339](#)
- NASServDomainPrefTlv, [339](#)
- NASTimeInfoTlv, [341](#)
- newMTMessageTlv, [342](#)
- sMSCAddressTlv, [564](#)
- sMSEtwsMessageTlv, [565](#)
- sMSOnIMSTlv, [567](#)
- swidms_ehrpdMTUSizeTlv, [569](#)
- swidms_hrpdmTUSizeTlv, [570](#)
- swidms_ifaceCfgTlv, [571](#)
- swidms_mtuSize3gppTlv, [572](#)
- swidms_supportedIntBitmaskTlv, [572](#)
- swidms_usbMTUSizeTlv, [573](#)
- transferRouteMessageTlv, [576](#)
- uim_GetSlotsInfoTlv, [590](#)
- uim_GetSlotsStatusTlv, [591](#)
- unpack_pds_SetPdsState_Ind_t, [828](#)
- unpack_uim_SLQSUIRefreshCallback_Ind_t, [923](#)
- UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
- UnpackSwiAvmsEventReportConfig, [1026](#)
- UnpackSwiAvmsEventReportConnectionRequest, [1027](#)
- UnpackSwiAvmsEventReportDataSessionStatus, [1027](#)
- UnpackSwiAvmsEventReportHTTPStatus, [1028](#)
- UnpackSwiAvmsEventReportNotification, [1030](#)
- UnpackSwiAvmsEventReportPackageID, [1030](#)
- UnpackSwiAvmsEventReportRegStatus, [1030](#)
- UnpackSwiAvmsEventReportSessionType, [1031](#)
- UnpackSwiAvmsEventReportWAMSParam-Change, [1032](#)
- wds_ChannelRateTlv, [1081](#)
- wds_ConnStatusTlv, [1082](#)
- wds_DataBearTechTlv, [1084](#)
- wds_DataULongLongTlv, [1085](#)
- wds_DataULongTlv, [1085](#)
- wds_DHCPLeaseOptTlv, [1086](#)
- wds_DHCPLeaseStateTlv, [1086](#)
- wds_DHCPProfileIDTlv, [1087](#)
- wds_DormStatTlv, [1091](#)
- wds_IPv4AdTlv, [1093](#)
- wds_LastMdmCallEndRsnTlv, [1094](#)
- wds_RXBytesOKTlv, [1097](#)
- wds_TXBytesOKTlv, [1099](#)
- tlvPresent
 - voice_USSDNotificationNetworkInfo, [1079](#)
- TlvResult
 - unpack_swidms_SLQSSwiDmsGetSecureInfo_t, [894](#)
- TlvResultCode
 - qmTlvResult, [542](#)
- Tlvresult
 - pack_dms_GetCustFeaturesV2_t, [351](#)
 - pack_dms_SetCustFeaturesV2_t, [356](#)
 - pack_dms_SetPower_t, [358](#)
 - pack_dms_SetUSBComp_t, [358](#)
 - pack_dms_UIMGetICCID_t, [365](#)
 - pack_fms_GetImagesPreference_t, [369](#)
 - pack_fms_GetStoredImages_t, [369](#)
 - pack_fms_SetImagesPreference_t, [370](#)
 - pack_loc_Delete_Assist_Data_t, [379](#)
 - pack_loc_EventRegister_t, [381](#)

- pack_loc_SetExtPowerState_t, 381
- pack_loc_SetOperationMode_t, 382
- pack_loc_SLQSLOCGetBestAvailPos_t, 382
- pack_loc_Start_t, 394
- pack_loc_Stop_t, 394
- pack_nas_SetNetworkPreference_t, 398
- pack_uim_ChangePin_t, 480
- pack_uim_ReadTransparent_t, 481
- pack_uim_SetPinProtection_t, 482
- pack_uim_UnblockPin_t, 490
- pack_uim_VerifyPin_t, 491
- unpack_cat_SetCATEventCallback_t, 611
- unpack_dms_GetBandCapability_t, 614
- unpack_dms_GetCrashAction_t, 615
- unpack_dms_GetCustFeature_t, 617
- unpack_dms_GetCustFeaturesV2_t, 618
- unpack_dms_GetDeviceCap_t, 619
- unpack_dms_GetDeviceCapabilitiesV2_t, 623
- unpack_dms_GetDeviceHardwareRev_t, 624
- unpack_dms_GetDeviceMfr_t, 625
- unpack_dms_GetDeviceSerialNumbers_t, 627
- unpack_dms_GetFirmwareInfo_t, 628
- unpack_dms_GetFirmwareRevision_t, 629
- unpack_dms_GetFirmwareRevisions_t, 631
- unpack_dms_GetFSN_t, 631
- unpack_dms_GetIMSI_t, 633
- unpack_dms_GetManufacturer_t, 633
- unpack_dms_GetModelID_t, 634
- unpack_dms_GetNetworkTime_t, 635
- unpack_dms_GetNetworkTimeV2_t, 636
- unpack_dms_GetOfflineReason_t, 637
- unpack_dms_GetPower_t, 638
- unpack_dms_GetPRLVersion_t, 639
- unpack_dms_GetUSBComp_t, 642
- unpack_dms_GetVoiceNumber_t, 643
- unpack_dms_PSMCfGChange_ind_t, 644
- unpack_dms_ResetToFactoryDefaults_t, 645
- unpack_dms_SetActivationStatusCallback_t, 645
- unpack_dms_SetCustFeature_t, 646
- unpack_dms_SetCustFeaturesV2_t, 647
- unpack_dms_SetEventReport_ind_t, 648
- unpack_dms_SetEventReport_t, 648
- unpack_dms_SetFirmwarePreference_t, 649
- unpack_dms_SetIndicationRegister_t, 649
- unpack_dms_SetPower_t, 650
- unpack_dms_SetUSBComp_t, 650
- unpack_dms_SLQSDmsSwiGetResetInfo_ind_t, 654
- unpack_dms_SLQSDmsSwiGetResetInfo_t, 655
- unpack_dms_SLQSDmsSwiIndicationRegister_t, 657
- unpack_dms_SLQSGetERIFile_t, 666
- unpack_dms_SLQSSetPowerSaveModeConfig_t, 668
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 668
- unpack_dms_SLQSSwiGetCrashInfo_t, 669
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, 670
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 670
- unpack_dms_SLQSSwiGetFwUpdateStatus_t, 673
- unpack_dms_SLQSSwiGetHostDevInfo_t, 674
- unpack_dms_SLQSSwiGetOSInfo_t, 675
- unpack_dms_SLQSSwiGetSerialNoExt_t, 676
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, 676
- unpack_dms_SLQSSwiSetHostDevInfo_t, 677
- unpack_dms_SLQSSwiSetOSInfo_t, 677
- unpack_dms_SLQSUIMGetState_t, 678
- unpack_dms_SwiSetEventReport_t, 679
- unpack_dms_SwiUimSelect_t, 680
- unpack_dms_UIMGetControlKeyStatus_t, 681
- unpack_dms_UIMGetICCID_t, 681
- unpack_dms_UIMGetPINStatus_t, 683
- unpack_dms_UIMSetControlKeyProtection_t, 684
- unpack_dms_UIMSetPINProtection_t, 685
- unpack_dms_UIMUnlockControlKey_t, 685
- unpack_fms_GetImagesPreference_t, 686
- unpack_fms_GetStoredImages_t, 687
- unpack_fms_SetImagesPreference_t, 687
- unpack_loc_BestAvailPos_ind_t, 717
- unpack_loc_Delete_Assist_Data_t, 718
- unpack_loc_DeleteAssistData_ind_t, 719
- unpack_loc_EngineState_ind_t, 720
- unpack_loc_EventNMEA_ind_t, 720
- unpack_loc_EventRegister_t, 721
- unpack_loc_GetServer_ind_t, 724
- unpack_loc_GnssSvInfo_ind_t, 725
- unpack_loc_PositionRpt_ind_t, 736
- unpack_loc_SetExtPowerConfig_ind_t, 737
- unpack_loc_SetExtPowerState_t, 738
- unpack_loc_SetOperationMode_ind_t, 739
- unpack_loc_SetOperationMode_t, 739
- unpack_loc_SetServer_ind_t, 740
- unpack_loc_SLQSLOCGetBestAvailPos_t, 741
- unpack_loc_SLQSLOCGetOpMode_t, 741
- unpack_loc_Start_t, 742
- unpack_loc_Stop_t, 742
- unpack_nas_GetNetworkPreference_t, 750
- unpack_nas_SetNetworkPreference_t, 761
- unpack_nas_SetServingSystemCallback_ind_t, 762
- unpack_nas_SlqsGetLTECphyCAInfo_t, 764
- unpack_nas_SLQSNasGetRFInfo_t, 795
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 805
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 806
- unpack_result_t, 856
- unpack_swidms_SLQSSwiDmsSetHWWatchdog_t, 896
- unpack_swidms_SLQSSwiDmsSetMTU_t, 896
- unpack_swidms_SLQSSwiDmsSetUsbComp_t, 897
- unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 909

- unpack_tmd_SLQSTmdGetMitigationDevList_t, 910
- unpack_tmd_SLQSTmdGetMitigationLvl_t, 911
- unpack_tmd_SLQSTmdRegNotMitigationLvl_t, 912
- unpack_uim_ChangePin_t, 913
- unpack_uim_GetCardStatus_t, 913
- unpack_uim_GetCardStatusV2_t, 914
- unpack_uim_ReadTransparent_t, 915
- unpack_uim_SetPinProtection_t, 916
- unpack_uim_UnblockPin_t, 925
- unpack_uim_UnblockPinV2_t, 926
- unpack_uim_VerifyPin_t, 926
- unpack_wds_SLQSCreateProfile_t, 994
- unpack_wds_SLQSGetProfileSettings_t, 1004
- unpack_wds_SLQSGetProfileSettingsV2_t, 1005
- unpack_wds_SLQSSetIPFamilyPreference_t, 1009
- tmd.h, 1461
 - pack_tmd_SLQSTmdDeRegNotMitigationLvl, 1461
 - pack_tmd_SLQSTmdGetMitigationDevList, 1462
 - pack_tmd_SLQSTmdGetMitigationLvl, 1462
 - pack_tmd_SLQSTmdRegNotMitigationLvl, 1463
 - TMD_MAX_DEV_LIST, 1461
 - unpack_tmd_SLQSTmdDeRegNotMitigationLvl, 1463
 - unpack_tmd_SLQSTmdGetMitigationDevList, 1463
 - unpack_tmd_SLQSTmdGetMitigationLvl, 1464
 - unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind, 1464
 - unpack_tmd_SLQSTmdRegNotMitigationLvl, 1464
- tmd_mitigationDevList, 575
 - maxMitigationLevel, 576
 - mitigationDevId, 576
 - mitigationDevIdLen, 576
- tmrSIPRegn
 - ims_SIPRegnTmrInfo, 94
- tmrT1
 - ims_TmrT1Info, 96
- tmrT2
 - ims_TmrT2Info, 96
- tmrTf
 - ims_TmrTfInfo, 97
- toServiceId
 - sms_BroadcastConfig, 550
- toggleMode
 - voice_lineCtrlInfo, 1068
- TokenBucket
 - unpack_qos_swiQosFlow_t, 854
- tokenRate
 - unpack_qos_tokenBucket_t, 855
- tosMask
 - LibPackTFTIDParams, 144
- total_rx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- total_rx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- total_tx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- total_tx_bytes_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- total_tx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 843
- TrackAreaCode
 - unpack_nas_SLQSGetServingSystem_t, 773
- TrackingStatus
 - unpack_pds_SetPdsState_Ind_t, 828
- TrafficClass
 - unpack_qos_swiQosFlow_t, 854
- trafficClass
 - LibPackUMTSQoS, 146
 - wds_UMTSMInQoS, 1102
- trafficPriority
 - LibPackUMTSQoS, 146
 - wds_UMTSMInQoS, 1102
- TranDstPort
 - unpack_qos_swiQosFilter_t, 850
- TranSrcPort
 - unpack_qos_swiQosFilter_t, 850
- TransCap
 - sms_transLayerInfo, 562
- TransType
 - sms_transLayerInfo, 563
- transactionID
 - sMSTransferRouteMTMessageInfo, 568
- transferDelay
 - LibPackUMTSQoS, 146
 - wds_UMTSMInQoS, 1102
- TransferRouteMTMessageInfo
 - transferRouteMessageTlv, 576
- transferRouteMessageTlv, 576
 - TlvPresent, 576
 - TransferRouteMTMessageInfo, 576
- transferStats
 - pack_wds_SLQSSetWdsEventCallback_t, 530
- trueSrvStatus
 - nas_GSMSrvStatusInfo, 206
 - nas_NR5GSerStatTlv, 255
- tx_bytes
 - unpack_QosFlowStat_t, 856
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- tx_bytes_drp
 - unpack_QosFlowStat_t, 856
- tx_pkt
 - unpack_QosFlowStat_t, 856
- tx_pkt_drp
 - unpack_QosFlowStat_t, 856
- tx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 1017
- TxByteOKCnt
 - wds_TXBytesOKTlv, 1099

- TxDropConutTlv
 - unpack_RMTransferStatistics_ind_t, [859](#)
- txOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, [1004](#)
- TxOkByteCountTlv
 - unpack_RMTransferStatistics_ind_t, [859](#)
- TxOkConutTlv
 - unpack_RMTransferStatistics_ind_t, [859](#)
- txPower
 - nas_txInfo, [310](#)
- TxQFilter
 - unpack_qos_QosFlowInfo_t, [839](#)
- TxQFlowGranted
 - unpack_qos_QosFlowInfo_t, [839](#)
- type
 - pack_wds_GetDefaultProfileNum_t, [511](#)
 - pack_wds_SetDefaultProfileNum_t, [518](#)
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, [654](#)
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, [655](#)
 - unpack_qmi_t, [834](#)
- typeSize
 - unpack_sms_GetSMSCAddress_t, [860](#)
- u16PRLVersion
 - unpack_dms_GetPRLVersion_t, [639](#)
- u8PRLPreference
 - unpack_dms_GetPRLVersion_t, [639](#)
- UDPDstPort
 - unpack_qos_swiQosFilter_t, [850](#)
- UDPSrcPort
 - unpack_qos_swiQosFilter_t, [850](#)
- ULONG
 - SwiDataTypes.h, [1433](#)
- ULONGLONG
 - SwiDataTypes.h, [1433](#)
- UMTSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, [1008](#)
- UMTSInstInfo
 - nas_UMTSInfo, [312](#)
- UMTSLTENbrCell
 - nas_WCDMAInfoLTENeighborCell, [317](#)
- UMTSReqQoS
 - LibPackUMTSReqQoSsigInd, [147](#)
- UNIQUE_ID_LEN
 - dms.h, [1130](#)
- UNUSEDPARAM
 - common.h, [1117](#)
 - SwiDataTypes.h, [1433](#)
- uPhyCardStatus
 - slot_t, [547](#)
- uPhySlotStatus
 - slot_t, [547](#)
- USBComp
 - pack_dms_SetUSBComp_t, [358](#)
 - unpack_dms_GetUSBComp_t, [642](#)
- USHORT
 - SwiDataTypes.h, [1433](#)
- USSDNotificationNetworkInfo
 - unpack_voice_USSDNotificationCallback_ind_t, [968](#)
- USSInformation
 - pack_voice_SLQSVoiceOrigUSSDNoWait_t, [504](#)
- UUSData
 - voice_UUSInfo, [1080](#)
- UUSDatalen
 - voice_UUSInfo, [1080](#)
- UUSDcs
 - voice_UUSInfo, [1080](#)
- UUSType
 - voice_UUSInfo, [1080](#)
- uarfcn
 - nas_lteWcdmaCellInfo, [244](#)
 - nas_UMTSInfo, [312](#)
 - nas_wcdmaUARFCN, [322](#)
- ueInIdle
 - nas_LTEInfoInterfreq, [225](#)
 - nas_LTEInfoIntrafreq, [227](#)
 - nas_LTEInfoNeighboringGSM, [228](#)
 - nas_LTEInfoNeighboringWCDMA, [229](#)
- uim.h, [1465](#)
 - MAX_ATR_LENGTH, [1468](#)
 - MAX_ICCID_LENGTH, [1468](#)
 - MAX_NO_OF_SLOTS, [1468](#)
 - MAX_PHY_SLOTS_INFO, [1468](#)
 - MAX_SLOTS_STATUS, [1468](#)
 - pack_uim_ChangePin, [1469](#)
 - pack_uim_GetCardStatus, [1469](#)
 - pack_uim_ReadTransparent, [1470](#)
 - pack_uim_SLQSUIMAuthenticate, [1470](#)
 - pack_uim_SLQSUIMDepersonalization, [1471](#)
 - pack_uim_SLQSUIMEventRegister, [1471](#)
 - pack_uim_SLQSUIMGetConfiguration, [1471](#)
 - pack_uim_SLQSUIMGetFileAttributes, [1472](#)
 - pack_uim_SLQSUIMGetSlotsStatus, [1472](#)
 - pack_uim_SLQSUIMPowerDown, [1473](#)
 - pack_uim_SLQSUIMPowerUp, [1473](#)
 - pack_uim_SLQSUIMRefreshComplete, [1473](#)
 - pack_uim_SLQSUIMRefreshGetLastEvent, [1474](#)
 - pack_uim_SLQSUIMRefreshOK, [1474](#)
 - pack_uim_SLQSUIMRefreshRegister, [1474](#)
 - pack_uim_SLQSUIMReset, [1475](#)
 - pack_uim_SLQSUIMSwitchSlot, [1475](#)
 - pack_uim_SetPinProtection, [1470](#)
 - pack_uim_UnblockPin, [1475](#)
 - pack_uim_VerifyPin, [1476](#)
 - unpack_uim_ChangePin, [1476](#)
 - unpack_uim_GetCardStatus, [1477](#)
 - unpack_uim_GetCardStatusV2, [1477](#)
 - unpack_uim_ReadTransparent, [1477](#)
 - unpack_uim_SLQSUIMAuthenticate, [1478](#)
 - unpack_uim_SLQSUIMDepersonalization, [1479](#)
 - unpack_uim_SLQSUIMEventRegister, [1479](#)
 - unpack_uim_SLQSUIMGetConfiguration, [1479](#)
 - unpack_uim_SLQSUIMGetFileAttributes, [1480](#)
 - unpack_uim_SLQSUIMGetSlotsStatus, [1480](#)
 - unpack_uim_SLQSUIMGetSlotsStatusV2, [1481](#)

- unpack_uim_SLQSUIMPowerDown, 1481
- unpack_uim_SLQSUIMPowerDown_t, 1469
- unpack_uim_SLQSUIMPowerUp, 1481
- unpack_uim_SLQSUIMPowerUp_t, 1469
- unpack_uim_SLQSUIMRefreshCallback_Ind, 1482
- unpack_uim_SLQSUIMRefreshComplete, 1482
- unpack_uim_SLQSUIMRefreshComplete_t, 1469
- unpack_uim_SLQSUIMRefreshGetLastEvent, 1482
- unpack_uim_SLQSUIMRefreshOK, 1483
- unpack_uim_SLQSUIMRefreshOK_t, 1469
- unpack_uim_SLQSUIMRefreshRegister, 1483
- unpack_uim_SLQSUIMRefreshRegister_t, 1469
- unpack_uim_SLQSUIMReset, 1483
- unpack_uim_SLQSUIMReset_t, 1469
- unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind, 1484
- unpack_uim_SLQSUIMSwitchSlot, 1484
- unpack_uim_SLQSUIMSwitchSlot_t, 1469
- unpack_uim_SetPinProtection, 1478
- unpack_uim_SetUimSlotStatusChangeCallback_ind, 1478
- unpack_uim_UnblockPin, 1484
- unpack_uim_UnblockPinV2, 1485
- unpack_uim_VerifyPin, 1485
- uim_GetSlotsInfoTlv, 590
 - NumberOfPhySlotInfo, 590
 - TlvPresent, 590
 - uimSlotInfo, 590
- uim_GetSlotsStatusTlv, 591
 - NumberOfPhySlot, 591
 - TlvPresent, 591
 - uimSlotStatus, 591
- uim_UIMSessionInformation, 602
 - aid, 603
 - aidLength, 603
 - sessionType, 603
- uim_appStatus, 576
 - aidLength, 579
 - aidVal, 579
 - appState, 579
 - appType, 579
 - persoFeature, 579
 - persoRetries, 579
 - persoState, 579
 - persoUnblockRetries, 579
 - pin1Retries, 579
 - pin1State, 579
 - pin2Retries, 579
 - pin2State, 579
 - puk1Retries, 579
 - puk2Retries, 579
 - univPin, 579
- uim_authenticateResult, 579
 - content, 580
 - contentLen, 580
- uim_authenticationData, 580
 - context, 581
 - data, 581
 - dataLen, 581
- uim_cardResult, 581
 - sw1, 582
 - sw2, 582
- uim_cardStatus, 582
 - index1xPri, 583
 - index1xSec, 583
 - indexGwPri, 583
 - indexGwSec, 583
 - numSlot, 583
 - SlotInfo, 583
- uim_changeUIMPIN, 583
 - oldPINLen, 584
 - oldPINVal, 584
 - pinID, 584
 - pinLen, 584
 - pinVal, 584
- uim_depersonalizationInformation, 584
 - ckLen, 585
 - ckVal, 585
 - feature, 585
 - operation, 585
- uim_encryptedPIN1, 585
 - pin1Len, 586
 - pin1Val, 586
- uim_fileAttributes, 586
 - fileID, 589
 - fileSize, 589
 - fileType, 589
 - rawLen, 589
 - rawValue, 589
 - recordCount, 589
 - recordSize, 589
 - secActivate, 589
 - secActivateMask, 589
 - secDeactivate, 589
 - secDeactivateMask, 589
 - secIncrease, 589
 - secIncreaseMask, 589
 - secRead, 589
 - secReadMask, 589
 - secWrite, 589
 - secWriteMask, 589
- uim_fileInfo, 589
 - fileID, 590
 - path, 590
 - pathLen, 590
- uim_hotSwapStatus, 591
 - hotSwap, 592
 - hotSwapLength, 592
- uim_personalizationStatus, 592
 - feature, 593
 - numFeatures, 593
 - unblockLeft, 593
 - verifyLeft, 593
- uim_physlotInfo, 593
 - atrValue, 594

- atrValueLen, [594](#)
- cardProtocol, [594](#)
- iseUICC, [594](#)
- numApp, [594](#)
- uim_physlotsInfo, [594](#)
 - uimSlotInfo, [594](#)
- uim_readResult, [594](#)
 - content, [595](#)
 - contentLen, [595](#)
- uim_readTransparentInfo, [595](#)
 - length, [595](#)
 - offset, [595](#)
- uim_refreshevent, [595](#)
 - aid, [596](#)
 - aidLength, [597](#)
 - arrfileInfo, [597](#)
 - mode, [597](#)
 - numOfFiles, [597](#)
 - sessionType, [597](#)
 - stage, [597](#)
- uim_registerRefresh, [597](#)
 - arrfileInfo, [597](#)
 - numFiles, [598](#)
 - registerFlag, [598](#)
 - voteForInit, [598](#)
- uim_remainingRetries, [598](#)
 - unblockLeft, [598](#)
 - verifyLeft, [598](#)
- uim_select
 - pack_dms_SwiUimSelect_t, [363](#)
- uim_sessionInformation, [598](#)
 - aid, [599](#)
 - aidLength, [599](#)
 - sessionType, [599](#)
- uim_setPINProtection, [599](#)
 - pinID, [600](#)
 - pinLength, [600](#)
 - pinOperation, [600](#)
 - pinValue, [600](#)
- uim_simBusyStatus, [600](#)
 - simBusy, [600](#)
 - simBusyLength, [600](#)
- uim_slotInfo, [601](#)
 - AppStatus, [602](#)
 - cardState, [602](#)
 - errorState, [602](#)
 - numApp, [602](#)
 - upinRetries, [602](#)
 - upinState, [602](#)
 - upukRetries, [602](#)
- uim_unblockUIMPIN, [603](#)
 - newPINLen, [604](#)
 - newPINVal, [604](#)
 - pinID, [604](#)
 - pukLen, [604](#)
 - pukVal, [604](#)
- uim_validCardStatus, [604](#)
 - validCard, [604](#)
 - validCardLength, [605](#)
- uim_verifyUIMPIN, [605](#)
 - pinID, [605](#)
 - pinLen, [605](#)
 - pinVal, [605](#)
- uimAutoSwitchActSlot
 - dms_UimAutoSwitchActSlotTlv, [72](#)
- uimSelect
 - unpack_dms_SLQSDmsSwiGetUimSelection_t, [656](#)
- uimSlotInfo
 - uim_GetSlotsInfoTlv, [590](#)
 - uim_physlotsInfo, [594](#)
- uimSlotStatus
 - slots_t, [550](#)
 - uim_GetSlotsStatusTlv, [591](#)
- UimStatusTlv
 - unpack_dms_SwiEventReportCallBack_ind_t, [679](#)
- ulData
 - wds_DataULongTlv, [1085](#)
- ulMask
 - rmTrasferStaticsReq, [545](#)
- ulMax
 - PackSwiAvmsSetSettingsPeriodInfo, [539](#)
- ulMin
 - PackSwiAvmsSetSettingsPeriodInfo, [539](#)
- ulPhysicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, [489](#)
- ulPkgDownloadComplete
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
- ulPkgDownloadSize
 - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1025](#)
- ulldata
 - wds_DataULongLongTlv, [1085](#)
- umtsEcio
 - nas_UMTSinstInfo, [313](#)
- umtsInst
 - nas_UMTSInfo, [312](#)
- umtsLTENbrCellLen
 - nas_WCDMAInfoLTENeighborCell, [317](#)
- umtsPsc
 - nas_UMTSinstInfo, [313](#)
- umtsRscp
 - nas_UMTSinstInfo, [313](#)
- umtsUarfcn
 - nas_UMTSinstInfo, [313](#)
- UnPackGetProfileSettingOut, [1022](#)
 - curProfile, [1023](#)
 - pExtErrCode, [1023](#)
- UnPackGetProfileSettingOutV2, [1023](#)
 - curProfile, [1023](#)
 - pExtErrCode, [1023](#)
- unblockLeft
 - uim_personalizationStatus, [593](#)
 - uim_remainingRetries, [598](#)
- unblockRetriesLeft

- unpack_dms_UIMGetControlKeyStatus_t, 681
- unpack_dms_UIMSetPINProtection_t, 685
- unpack_dms_UIMUnlockControlKey_t, 685
- uniqueID
 - image_info_t, 84
- univPin
 - appStats, 42
 - uim_appStatus, 579
- universalTime
 - unpack_nas_SLQSNasNetworkTimeCallback_ind_t, 801
- unpack_QosFlowStat_t, 855
 - bearerId, 856
 - tx_bytes, 856
 - tx_bytes_drp, 856
 - tx_pkt, 856
 - tx_pkt_drp, 856
- unpack_RMTransferStatistics_ind_t, 858
 - ParamPresenceMask, 858
 - RxDropConutTlv, 858
 - RxOkByteCountTlv, 858
 - RxOkConutTlv, 859
 - TxDropConutTlv, 859
 - TxOkByteCountTlv, 859
 - TxOkConutTlv, 859
- unpack_audio_SLQSGetAudioPathConfig
 - audio.h, 1109
- unpack_audio_SLQSGetAudioPathConfig_t, 605
 - pCodecSTGain, 607
 - pECMode, 607
 - pMICGainSelect, 607
 - pNSEnable, 607
 - pRXAGCList, 607
 - pRXAVCList, 607
 - pTXAGCList, 607
 - pTXAVCSwitch, 607
 - pTXGain, 607
 - ParamPresenceMask, 607
- unpack_audio_SLQSGetAudioProfile
 - audio.h, 1109
- unpack_audio_SLQSGetAudioProfile_t, 608
 - EarMute, 608
 - MicMute, 609
 - ParamPresenceMask, 609
 - Profile, 609
 - Volume, 609
- unpack_audio_SLQSGetAudioVolTLBConfig
 - audio.h, 1109
- unpack_audio_SLQSGetAudioVolTLBConfig_t, 609
 - ParamPresenceMask, 609
 - ResCode, 609
- unpack_audio_SLQSSetAudioPathConfig
 - audio.h, 1110
- unpack_audio_SLQSSetAudioProfile
 - audio.h, 1110
- unpack_audio_SLQSSetAudioVolTLBConfig
 - audio.h, 1110
- unpack_audio_SLQSSetAudioVolTLBConfig_t, 609
 - ParamPresenceMask, 610
 - ResCode, 610
- unpack_cat_CATSndEnvelopeCommand
 - cat.h, 1113
- unpack_cat_CATSndEnvelopeCommand_t
 - cat.h, 1112
- unpack_cat_CATSndTerminalResponse
 - cat.h, 1113
- unpack_cat_CATSndTerminalResponse_t
 - cat.h, 1112
- unpack_cat_SetCATEventCallback
 - cat.h, 1114
- unpack_cat_SetCATEventCallback_t, 610
 - errorMask, 611
 - ParamPresenceMask, 611
 - Tlvresult, 611
- unpack_cat_SetCatEventCallback_ind
 - cat.h, 1114
- unpack_cat_SetCatEventCallback_ind_t, 610
 - CCETlv, 610
 - event_Index, 610
 - ParamPresenceMask, 610
- unpack_dms_ActivateAutomatic
 - dms.h, 1155
- unpack_dms_GetActivationState
 - dms.h, 1156
- unpack_dms_GetActivationState_t, 611
 - ParamPresenceMask, 612
 - state, 612
- unpack_dms_GetBandCapability
 - dms.h, 1156
- unpack_dms_GetBandCapability_t, 612
 - BandCapability, 614
 - ParamPresenceMask, 614
 - Tlvresult, 614
- unpack_dms_GetCrashAction
 - dms.h, 1157
- unpack_dms_GetCrashAction_t, 614
 - DevCrashState, 614
 - ParamPresenceMask, 614
 - Tlvresult, 615
- unpack_dms_GetCustFeature
 - dms.h, 1157
- unpack_dms_GetCustFeature_t, 615
 - DHCPRelayEnabled, 617
 - DisableIMSI, 617
 - GPSPMP, 617
 - GPSSel, 617
 - GpsEnable, 617
 - IPFamSupport, 617
 - IsVoiceEnabled, 617
 - ParamPresenceMask, 617
 - RMAutoConnect, 617
 - SMSSupport, 617
 - Tlvresult, 617
- unpack_dms_GetCustFeaturesV2
 - dms.h, 1157
- unpack_dms_GetCustFeaturesV2_t, 617

- GetCustomFeatureV2, 618
- ParamPresenceMask, 618
- Tlvresult, 618
- unpack_dms_GetDeviceCap
 - dms.h, 1158
- unpack_dms_GetDeviceCap_t, 618
 - DataServiceCapability, 619
 - MaxRXChannelRate, 619
 - MaxTXChannelRate, 619
 - ParamPresenceMask, 619
 - Radiolfaces, 619
 - RadiolfacesSize, 619
 - SimCapability, 619
 - Tlvresult, 619
- unpack_dms_GetDeviceCapabilities
 - dms.h, 1158
- unpack_dms_GetDeviceCapabilities_t, 619
 - dataServiceCaCapability, 620
 - maxRxChannelRate, 621
 - maxTxChannelRate, 621
 - ParamPresenceMask, 621
 - Radiolfaces, 621
 - radiolfacesSize, 621
 - simCapability, 621
- unpack_dms_GetDeviceCapabilitiesV2
 - dms.h, 1158
- unpack_dms_GetDeviceCapabilitiesV2_t, 621
 - DevCaps, 623
 - pDevCurSubsCaps, 623
 - pDevExplicitCfgIndex, 623
 - pDevMaxActDataSubsCaps, 623
 - pDevMaxCfgListCaps, 623
 - pDevMaxSubsCaps, 623
 - pDevMultiSimCaps, 623
 - pDevMultiSimVoiceDataCaps, 623
 - pDevSrvCaps, 623
 - pDevSubsFeatureModeCaps, 623
 - pDevSubsVoiceDataCaps, 623
 - pDevVoiceCaps, 623
 - pDevVoiceDataCaps, 623
 - ParamPresenceMask, 623
 - Tlvresult, 623
- unpack_dms_GetDeviceHardwareRev
 - dms.h, 1159
- unpack_dms_GetDeviceHardwareRev_t, 623
 - ParamPresenceMask, 624
 - String, 624
 - stringSize, 624
 - Tlvresult, 624
- unpack_dms_GetDeviceMfr
 - dms.h, 1159
- unpack_dms_GetDeviceMfr_t, 624
 - ParamPresenceMask, 625
 - String, 625
 - stringSize, 625
 - Tlvresult, 625
- unpack_dms_GetDeviceSerialNumbers
 - dms.h, 1159
- unpack_dms_GetDeviceSerialNumbers_t, 625
 - ESNString, 627
 - esnSize, 626
 - IMEIString, 627
 - imeiSize, 627
 - imeiSvnSize, 627
 - ImeiSvnString, 627
 - MEIDString, 627
 - meidSize, 627
 - ParamPresenceMask, 627
 - Tlvresult, 627
- unpack_dms_GetFSN
 - dms.h, 1161
- unpack_dms_GetFSN_t, 631
 - ParamPresenceMask, 631
 - String, 631
 - Tlvresult, 631
- unpack_dms_GetFirmwareInfo
 - dms.h, 1160
- unpack_dms_GetFirmwareInfo_t, 627
 - appversion_str, 628
 - bootversion_str, 628
 - carrier_str, 628
 - cur_carr_name, 628
 - cur_carr_rev, 628
 - modelid_str, 628
 - packageid_str, 628
 - ParamPresenceMask, 628
 - priversion_str, 628
 - sku_str, 628
 - Tlvresult, 628
- unpack_dms_GetFirmwareRevision
 - dms.h, 1160
- unpack_dms_GetFirmwareRevision_t, 629
 - AMSSString, 629
 - amssSize, 629
 - PRISString, 629
 - ParamPresenceMask, 629
 - Tlvresult, 629
- unpack_dms_GetFirmwareRevisions
 - dms.h, 1160
- unpack_dms_GetFirmwareRevisions_t, 630
 - AMSSString, 631
 - amssSize, 631
 - bootSize, 631
 - BootString, 631
 - PRISString, 631
 - ParamPresenceMask, 631
 - priSize, 631
 - Tlvresult, 631
- unpack_dms_GetHardwareRevision
 - dms.h, 1161
- unpack_dms_GetHardwareRevision_t, 632
 - hwVer, 632
 - ParamPresenceMask, 632
- unpack_dms_GetIMSI
 - dms.h, 1161
- unpack_dms_GetIMSI_t, 632

- imsi, [632](#)
- ParamPresenceMask, [633](#)
- Tlvresult, [633](#)
- unpack_dms_GetManufacturer
 - dms.h, [1162](#)
- unpack_dms_GetManufacturer_t, [633](#)
 - manufacturer, [633](#)
 - ParamPresenceMask, [633](#)
 - Tlvresult, [633](#)
- unpack_dms_GetModelID
 - dms.h, [1162](#)
- unpack_dms_GetModelID_t, [633](#)
 - modelid, [634](#)
 - ParamPresenceMask, [634](#)
 - Tlvresult, [634](#)
- unpack_dms_GetNetworkTime
 - dms.h, [1163](#)
- unpack_dms_GetNetworkTime_t, [634](#)
 - ParamPresenceMask, [635](#)
 - source, [635](#)
 - timestamp, [635](#)
 - Tlvresult, [635](#)
- unpack_dms_GetNetworkTimeV2
 - dms.h, [1163](#)
- unpack_dms_GetNetworkTimeV2_t, [635](#)
 - pSysTime, [636](#)
 - pUsrTime, [636](#)
 - ParamPresenceMask, [636](#)
 - source, [636](#)
 - timestamp, [636](#)
 - Tlvresult, [636](#)
- unpack_dms_GetOfflineReason
 - dms.h, [1163](#)
- unpack_dms_GetOfflineReason_t, [636](#)
 - pReasonMask, [637](#)
 - ParamPresenceMask, [637](#)
 - pbPlatform, [637](#)
 - Tlvresult, [637](#)
- unpack_dms_GetPRLVersion
 - dms.h, [1164](#)
- unpack_dms_GetPRLVersion_t, [638](#)
 - ParamPresenceMask, [639](#)
 - Tlvresult, [639](#)
 - u16PRLVersion, [639](#)
 - u8PRLPreference, [639](#)
- unpack_dms_GetPower
 - dms.h, [1164](#)
- unpack_dms_GetPower_t, [637](#)
 - HardwareControlledMode, [638](#)
 - OfflineReason, [638](#)
 - OperationMode, [638](#)
 - ParamPresenceMask, [638](#)
 - Tlvresult, [638](#)
- unpack_dms_GetSerialNumbers
 - dms.h, [1164](#)
- unpack_dms_GetSerialNumbers_t, [639](#)
 - esn, [640](#)
 - imei_no, [640](#)
 - imeisv_svn, [640](#)
 - meid, [640](#)
 - ParamPresenceMask, [640](#)
- unpack_dms_GetUSBComp
 - dms.h, [1165](#)
- unpack_dms_GetUSBComp_t, [640](#)
 - NumSupUSBComps, [642](#)
 - ParamPresenceMask, [642](#)
 - SupUSBComps, [642](#)
 - Tlvresult, [642](#)
 - USBComp, [642](#)
- unpack_dms_GetVoiceNumber
 - dms.h, [1165](#)
- unpack_dms_GetVoiceNumber_t, [642](#)
 - MIN, [643](#)
 - minSize, [643](#)
 - ParamPresenceMask, [643](#)
 - Tlvresult, [643](#)
 - VoiceNumber, [643](#)
 - voiceNumberSize, [643](#)
- unpack_dms_PSMCfgChange_ind
 - dms.h, [1165](#)
- unpack_dms_PSMCfgChange_ind_t, [644](#)
 - ActiveTimer, [644](#)
 - EnableState, [644](#)
 - ParamPresenceMask, [644](#)
 - PeriodicUpdateTimer, [644](#)
 - Tlvresult, [644](#)
- unpack_dms_ResetToFactoryDefaults
 - dms.h, [1166](#)
- unpack_dms_ResetToFactoryDefaults_t, [644](#)
 - ParamPresenceMask, [645](#)
 - Tlvresult, [645](#)
- unpack_dms_SLQSDmsSwiGetPCInfo
 - dms.h, [1170](#)
- unpack_dms_SLQSDmsSwiGetPCInfo_t, [650](#)
 - has_LpmFlag, [652](#)
 - has_PersistentLpm, [652](#)
 - has_Wdisable, [652](#)
 - LpmFlag, [653](#)
 - opMode, [653](#)
 - ParamPresenceMask, [653](#)
 - PersistentLpm, [653](#)
 - PowerOffMode, [653](#)
 - Wdisable, [653](#)
- unpack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [1170](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind
 - dms.h, [1170](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, [653](#)
 - ParamPresenceMask, [654](#)
 - source, [654](#)
 - Tlvresult, [654](#)
 - type, [654](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_t, [654](#)
 - ParamPresenceMask, [655](#)
 - source, [655](#)
 - Tlvresult, [655](#)

- type, 655
- unpack_dms_SLQSDmsSwiGetUimSelection
 - dms.h, 1171
- unpack_dms_SLQSDmsSwiGetUimSelection_t, 655
 - pUimAutoSwitchActSlot, 656
 - ParamPresenceMask, 656
 - uimSelect, 656
- unpack_dms_SLQSDmsSwiIndicationRegister
 - dms.h, 1171
- unpack_dms_SLQSDmsSwiIndicationRegister_t, 656
 - ParamPresenceMask, 657
 - Tlvresult, 657
- unpack_dms_SLQSGetBandCapability
 - dms.h, 1171
- unpack_dms_SLQSGetBandCapability_t, 657
 - bandCapability, 661
 - LteBandCapability, 661
 - ParamPresenceMask, 661
 - TdsBandCapability, 661
- unpack_dms_SLQSGetBandCapabilityExt
 - dms.h, 1172
- unpack_dms_SLQSGetBandCapabilityExt_t, 661
 - bandCapability, 665
 - LteBandCapability, 665
 - LteBandsSupport, 665
 - ParamPresenceMask, 665
 - TdsBandCapability, 665
- unpack_dms_SLQSGetERIFile
 - dms.h, 1172
- unpack_dms_SLQSGetERIFile_t, 665
 - eriFile, 666
 - ParamPresenceMask, 666
 - Tlvresult, 666
- unpack_dms_SLQSGetPowerSaveModeConfig
 - dms.h, 1173
- unpack_dms_SLQSGetPowerSaveModeConfig_t, 666
 - pActiveTimer, 667
 - pDurationThreshold, 667
 - pEarlyWakeupTime, 667
 - pPeriodicUpdateTimer, 667
 - pPsmEnableState, 667
 - pRandomizationWindow, 667
 - ParamPresenceMask, 667
- unpack_dms_SLQSSetPowerSaveModeConfig
 - dms.h, 1173
- unpack_dms_SLQSSetPowerSaveModeConfig_t, 667
 - ParamPresenceMask, 668
 - Tlvresult, 668
- unpack_dms_SLQSSwiClearDyingGaspStatistics
 - dms.h, 1173
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 668
 - ParamPresenceMask, 668
 - Tlvresult, 668
- unpack_dms_SLQSSwiGetCrashInfo
 - dms.h, 1174
- unpack_dms_SLQSSwiGetCrashInfo_t, 668
 - crashInfoParam, 669
 - ParamPresenceMask, 669
 - Tlvresult, 669
- unpack_dms_SLQSSwiGetDyingGaspCfg
 - dms.h, 1174
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, 669
 - pGetDyingGaspCfg, 670
 - ParamPresenceMask, 670
 - Tlvresult, 670
- unpack_dms_SLQSSwiGetDyingGaspStatistics
 - dms.h, 1174
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 670
 - pGetDyingGaspStatistics, 670
 - ParamPresenceMask, 670
 - Tlvresult, 670
- unpack_dms_SLQSSwiGetFirmwareCurr
 - dms.h, 1175
- unpack_dms_SLQSSwiGetFirmwareCurr_t, 670
 - carrier, 671
 - fwvers, 671
 - numEntries, 671
 - pCurrImgInfo, 671
 - ParamPresenceMask, 671
 - pkgver, 671
 - priver, 671
- unpack_dms_SLQSSwiGetFwUpdateStatus
 - dms.h, 1175
- unpack_dms_SLQSSwiGetFwUpdateStatus_t, 671
 - imgType, 673
 - logString, 673
 - ParamPresenceMask, 673
 - refData, 673
 - refString, 673
 - ResCode, 673
 - Tlvresult, 673
- unpack_dms_SLQSSwiGetHostDevInfo
 - dms.h, 1175
- unpack_dms_SLQSSwiGetHostDevInfo_t, 673
 - hostID, 674
 - manString, 674
 - modelString, 674
 - ParamPresenceMask, 674
 - plasmaIDString, 674
 - swVerString, 674
 - Tlvresult, 674
- unpack_dms_SLQSSwiGetOSInfo
 - dms.h, 1176
- unpack_dms_SLQSSwiGetOSInfo_t, 674
 - nameString, 675
 - ParamPresenceMask, 675
 - Tlvresult, 675
 - versionString, 675
- unpack_dms_SLQSSwiGetSerialNoExt
 - dms.h, 1176
- unpack_dms_SLQSSwiGetSerialNoExt_t, 675
 - meidString, 676
 - ParamPresenceMask, 676
 - Tlvresult, 676
- unpack_dms_SLQSSwiSetDyingGaspCfg

- dms.h, 1176
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, 676
 - ParamPresenceMask, 676
 - Tlvresult, 676
- unpack_dms_SLQSSwiSetHostDevInfo
 - dms.h, 1177
- unpack_dms_SLQSSwiSetHostDevInfo_t, 676
 - ParamPresenceMask, 677
 - Tlvresult, 677
- unpack_dms_SLQSSwiSetOSInfo
 - dms.h, 1177
- unpack_dms_SLQSSwiSetOSInfo_t, 677
 - ParamPresenceMask, 677
 - Tlvresult, 677
- unpack_dms_SLQSUIMGetState
 - dms.h, 1177
- unpack_dms_SLQSUIMGetState_t, 677
 - ParamPresenceMask, 678
 - state, 678
 - Tlvresult, 678
- unpack_dms_SetActivationStatusCallback
 - dms.h, 1166
- unpack_dms_SetActivationStatusCallback_t, 645
 - ParamPresenceMask, 645
 - Tlvresult, 645
- unpack_dms_SetCrashAction
 - dms.h, 1166
- unpack_dms_SetCrashAction_t, 645
 - notused, 646
 - ParamPresenceMask, 646
- unpack_dms_SetCustFeature
 - dms.h, 1167
- unpack_dms_SetCustFeature_t, 646
 - ParamPresenceMask, 646
 - Tlvresult, 646
- unpack_dms_SetCustFeaturesV2
 - dms.h, 1167
- unpack_dms_SetCustFeaturesV2_t, 646
 - ParamPresenceMask, 647
 - Tlvresult, 647
- unpack_dms_SetEventReport
 - dms.h, 1167
- unpack_dms_SetEventReport_ind
 - dms.h, 1168
- unpack_dms_SetEventReport_ind_t, 647
 - ActivationStatusTlv, 647
 - OperatingModeTlv, 647
 - ParamPresenceMask, 648
 - Tlvresult, 648
- unpack_dms_SetEventReport_t, 648
 - ParamPresenceMask, 648
 - Tlvresult, 648
- unpack_dms_SetFirmwarePreference
 - dms.h, 1168
- unpack_dms_SetFirmwarePreference_t, 648
 - ParamPresenceMask, 649
 - Tlvresult, 649
- unpack_dms_SetIndicationRegister
 - dms.h, 1168
- unpack_dms_SetIndicationRegister_t, 649
 - ParamPresenceMask, 649
 - Tlvresult, 649
- unpack_dms_SetPower
 - dms.h, 1169
- unpack_dms_SetPower_t, 649
 - ParamPresenceMask, 650
 - Tlvresult, 650
- unpack_dms_SetUSBComp
 - dms.h, 1169
- unpack_dms_SetUSBComp_t, 650
 - ParamPresenceMask, 650
 - Tlvresult, 650
- unpack_dms_SwiEventReportCallBack_ind
 - dms.h, 1178
- unpack_dms_SwiEventReportCallBack_ind_t, 678
 - ParamPresenceMask, 679
 - TempTlv, 679
 - UimStatusTlv, 679
 - VoltTlv, 679
- unpack_dms_SwiSetEventReport
 - dms.h, 1178
- unpack_dms_SwiSetEventReport_t, 679
 - ParamPresenceMask, 679
 - Tlvresult, 679
- unpack_dms_SwiUimSelect
 - dms.h, 1178
- unpack_dms_SwiUimSelect_t, 679
 - ParamPresenceMask, 680
 - Tlvresult, 680
- unpack_dms_UIMChangePIN
 - dms.h, 1179
- unpack_dms_UIMGetControlKeyStatus
 - dms.h, 1179
- unpack_dms_UIMGetControlKeyStatus_t, 680
 - facilityState, 681
 - ParamPresenceMask, 681
 - Tlvresult, 681
 - unblockRetriesLeft, 681
 - verifyRetriesLeft, 681
- unpack_dms_UIMGetICCID
 - dms.h, 1179
- unpack_dms_UIMGetICCID_t, 681
 - ParamPresenceMask, 681
 - String, 681
 - stringSize, 681
 - Tlvresult, 681
- unpack_dms_UIMGetPINStatus
 - dms.h, 1180
- unpack_dms_UIMGetPINStatus_t, 682
 - p1Status, 683
 - p1UnblockRetriesLeft, 683
 - p1VerifyRetriesLeft, 683
 - p2Status, 683
 - p2UnblockRetriesLeft, 683
 - p2VerifyRetriesLeft, 683
 - ParamPresenceMask, 683

- Tlvresult, [683](#)
- unpack_dms_UIMSetControlKeyProtection
 - dms.h, [1180](#)
- unpack_dms_UIMSetControlKeyProtection_t, [683](#)
 - ParamPresenceMask, [684](#)
 - Tlvresult, [684](#)
 - verifyRetriesLeft, [684](#)
- unpack_dms_UIMSetPINProtection
 - dms.h, [1180](#)
- unpack_dms_UIMSetPINProtection_t, [684](#)
 - ParamPresenceMask, [685](#)
 - Tlvresult, [685](#)
 - unblockRetriesLeft, [685](#)
 - verifyRetriesLeft, [685](#)
- unpack_dms_UIMUnblockControlKey
 - dms.h, [1181](#)
- unpack_dms_UIMUnblockControlKey_t, [685](#)
 - ParamPresenceMask, [685](#)
 - Tlvresult, [685](#)
 - unblockRetriesLeft, [685](#)
- unpack_dms_UIMUnblockPIN
 - dms.h, [1181](#)
- unpack_dms_UIMVerifyPIN
 - dms.h, [1182](#)
- unpack_dms_ValidateSPC
 - dms.h, [1182](#)
- unpack_fms_GetImagesPreference
 - fms.h, [1185](#)
- unpack_fms_GetImagesPreference_t, [685](#)
 - ImageListSize, [686](#)
 - pImageList, [686](#)
 - ParamPresenceMask, [686](#)
 - Tlvresult, [686](#)
- unpack_fms_GetStoredImages
 - fms.h, [1185](#)
- unpack_fms_GetStoredImages_t, [686](#)
 - imageList, [687](#)
 - imagelistSize, [687](#)
 - ParamPresenceMask, [687](#)
 - Tlvresult, [687](#)
- unpack_fms_SetImagesPreference
 - fms.h, [1185](#)
- unpack_fms_SetImagesPreference_t, [687](#)
 - ImageTypes, [687](#)
 - ImageTypesSize, [687](#)
 - ParamPresenceMask, [687](#)
 - Tlvresult, [687](#)
- unpack_ims_SLQSGetIMSSMSConfig
 - ims.h, [1192](#)
- unpack_ims_SLQSGetIMSSMSConfig_t, [687](#)
 - pSettingResp, [688](#)
 - ParamPresenceMask, [688](#)
- unpack_ims_SLQSGetIMSUserConfig
 - ims.h, [1193](#)
- unpack_ims_SLQSGetIMSUserConfig_t, [688](#)
 - pIMSDomain, [689](#)
 - pSettingResp, [689](#)
 - ParamPresenceMask, [689](#)
- unpack_ims_SLQSGetIMSVoIPConfig
 - ims.h, [1193](#)
- unpack_ims_SLQSGetIMSVoIPConfig_t, [689](#)
 - pAmrMode, [691](#)
 - pRingTimer, [692](#)
 - pSettingResp, [692](#)
 - ParamPresenceMask, [692](#)
- unpack_ims_SLQSGetRegMgrConfig
 - ims.h, [1194](#)
- unpack_ims_SLQSGetRegMgrConfig_t, [692](#)
 - pIMSTestMode, [693](#)
 - pPCSCFPort, [693](#)
 - pSettingResp, [693](#)
 - ParamPresenceMask, [693](#)
- unpack_ims_SLQSGetSIPConfig
 - ims.h, [1194](#)
- unpack_ims_SLQSGetSIPConfig_t, [693](#)
 - pSIPLocalPort, [694](#)
 - pSettingResp, [694](#)
 - pSigCompEnabled, [694](#)
 - pSubscribeTimer, [694](#)
 - pTimerSIPReg, [694](#)
 - pTimerT1, [694](#)
 - pTimerT2, [694](#)
 - pTimerTf, [694](#)
 - ParamPresenceMask, [694](#)
- unpack_ims_SLQSImsConfigIndicationRegister
 - ims.h, [1194](#)
- unpack_ims_SLQSImsConfigIndicationRegister_t
 - ims.h, [1188](#)
- unpack_ims_SLQSRegMgrCfgCallBack_ind
 - ims.h, [1195](#)
- unpack_ims_SLQSRegMgrCfgCallBack_ind_t, [694](#)
 - PCTlv, [695](#)
 - PNTlv, [695](#)
 - ParamPresenceMask, [695](#)
 - TMTlv, [695](#)
- unpack_ims_SLQSSIPCfgCallBack_ind
 - ims.h, [1197](#)
- unpack_ims_SLQSSIPCfgCallBack_ind_t, [698](#)
 - ParamPresenceMask, [699](#)
 - SCTlv, [699](#)
 - SPTlv, [699](#)
 - SRTlv, [699](#)
 - STTlv, [699](#)
 - TT1Tlv, [699](#)
 - TT2Tlv, [699](#)
 - TTfTlv, [699](#)
- unpack_ims_SLQSSMSCfgCallBack_ind
 - ims.h, [1197](#)
- unpack_ims_SLQSSMSCfgCallBack_ind_t, [699](#)
 - ParamPresenceMask, [700](#)
 - SFTlv, [700](#)
- unpack_ims_SLQSSetIMSSMSConfig
 - ims.h, [1195](#)
- unpack_ims_SLQSSetIMSSMSConfig_t, [695](#)
 - pSettingResp, [696](#)
 - ParamPresenceMask, [696](#)

- unpack_ims_SLQSSetIMSUserConfig
 - ims.h, [1195](#)
- unpack_ims_SLQSSetIMSUserConfig_t, [696](#)
 - pSettingResp, [696](#)
 - ParamPresenceMask, [696](#)
- unpack_ims_SLQSSetIMSVoIPConfig
 - ims.h, [1196](#)
- unpack_ims_SLQSSetIMSVoIPConfig_t, [696](#)
 - pSettingResp, [697](#)
 - ParamPresenceMask, [697](#)
- unpack_ims_SLQSSetRegMgrConfig
 - ims.h, [1196](#)
- unpack_ims_SLQSSetRegMgrConfig_t, [697](#)
 - pSettingResp, [697](#)
 - ParamPresenceMask, [697](#)
- unpack_ims_SLQSSetSIPConfig
 - ims.h, [1196](#)
- unpack_ims_SLQSSetSIPConfig_t, [697](#)
 - pSettingResp, [698](#)
 - ParamPresenceMask, [698](#)
- unpack_ims_SLQSUserCfgCallBack_ind
 - ims.h, [1197](#)
- unpack_ims_SLQSUserCfgCallBack_ind_t, [700](#)
 - IDTlv, [701](#)
 - ParamPresenceMask, [701](#)
- unpack_ims_SLQSVoIPCfgCallBack_ind
 - ims.h, [1198](#)
- unpack_ims_SLQSVoIPCfgCallBack_ind_t, [701](#)
 - AMTlv, [702](#)
 - ParamPresenceMask, [703](#)
 - RTTlv, [703](#)
 - SDTlv, [703](#)
- unpack_imsa_SLQSGetIMSARegStatus
 - imsa.h, [1201](#)
- unpack_imsa_SLQSGetIMSARegStatus_t, [703](#)
 - ImRegErrCode, [704](#)
 - ImRegStatus, [704](#)
 - NewImRegStatus, [704](#)
 - ParamPresenceMask, [704](#)
- unpack_imsa_SLQSGetIMSAServiceStatus
 - imsa.h, [1201](#)
- unpack_imsa_SLQSGetIMSAServiceStatus_t, [704](#)
 - ParamPresenceMask, [706](#)
 - SmsServiceRat, [706](#)
 - SmsServiceStatus, [706](#)
 - UtServiceRat, [706](#)
 - UtServiceStatus, [706](#)
 - VoipServiceRat, [706](#)
 - VoipServiceStatus, [706](#)
 - VsServiceRat, [706](#)
 - VsServiceStatus, [706](#)
 - VtServiceRat, [706](#)
 - VtServiceStatus, [706](#)
- unpack_imsa_SLQSImsaPdpStatusCallBack_ind
 - imsa.h, [1202](#)
- unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, [706](#)
 - FailErrCode, [707](#)
 - ParamPresenceMask, [707](#)
 - PdpConnState, [707](#)
- unpack_imsa_SLQSImsaRatStatusCallBack_ind
 - imsa.h, [1202](#)
- unpack_imsa_SLQSImsaRatStatusCallBack_ind_t, [707](#)
 - ParamPresenceMask, [708](#)
 - RatHandover, [708](#)
- unpack_imsa_SLQSImsaRegStatusCallBack_ind
 - imsa.h, [1202](#)
- unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, [708](#)
 - IMSRegistration, [708](#)
 - ParamPresenceMask, [708](#)
- unpack_imsa_SLQSImsaSvcStatusCallBack_ind
 - imsa.h, [1203](#)
- unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, [709](#)
 - ParamPresenceMask, [710](#)
 - SmsRat, [710](#)
 - SmsService, [710](#)
 - UtRat, [710](#)
 - UtService, [710](#)
 - VoipRat, [710](#)
 - VoipService, [710](#)
 - VtRat, [710](#)
 - VtService, [710](#)
- unpack_imsa_SLQSRegisterIMSAIndication
 - imsa.h, [1203](#)
- unpack_imsa_SLQSRegisterIMSAIndication_t
 - imsa.h, [1199](#)
- unpack_loc_BestAvailPos_Ind
 - loc.h, [1229](#)
- unpack_loc_BestAvailPos_Ind_t, [710](#)
 - pAltitudeWrtEllipsoid, [716](#)
 - pAltitudeWrtMeanSeaLevel, [716](#)
 - pGpsTime, [716](#)
 - pHeading, [716](#)
 - pHeadingUnc, [716](#)
 - pHorCirConf, [716](#)
 - pHorEllpConf, [716](#)
 - pHorReliability, [716](#)
 - pHorUncCircular, [716](#)
 - pHorUncEllipseOrientAzimuth, [716](#)
 - pHorUncEllipseSemiMajor, [716](#)
 - pHorUncEllipseSemiMinor, [716](#)
 - pLatitude, [716](#)
 - pLongitude, [716](#)
 - pMagneticDeviation, [716](#)
 - pPrecisionDilution, [716](#)
 - pSensorDataUsage, [716](#)
 - pSpeedHorizontal, [716](#)
 - pSpeedUnc, [716](#)
 - pSpeedVertical, [716](#)
 - pSpeedVerticalUnc, [716](#)
 - pSvUsedforFix, [716](#)
 - pTechnologyMask, [716](#)
 - pTimeSrc, [716](#)
 - pTimeUnc, [716](#)
 - pTimestampUtc, [716](#)
 - pVertConfidence, [716](#)

- pVertReliability, 717
- pVertUnc, 717
- pXid, 717
- ParamPresenceMask, 716
- status, 717
- Tlvresult, 717
- unpack_loc_CradleMountCallback_Ind
 - loc.h, 1229
- unpack_loc_CradleMountCallback_Ind_t, 717
 - cradleMountConfigStatus, 717
 - ParamPresenceMask, 717
- unpack_loc_Delete_Assist_Data_t, 717
 - ParamPresenceMask, 718
 - Tlvresult, 718
- unpack_loc_DeleteAssistData
 - loc.h, 1230
- unpack_loc_DeleteAssistData_Ind
 - loc.h, 1230
- unpack_loc_DeleteAssistData_Ind_t, 718
 - ParamPresenceMask, 719
 - status, 719
 - Tlvresult, 719
- unpack_loc_DeleteAssistData_t
 - loc.h, 1223
- unpack_loc_EngineState_Ind
 - loc.h, 1230
- unpack_loc_EngineState_Ind_t, 719
 - engineState, 720
 - ParamPresenceMask, 720
 - Tlvresult, 720
- unpack_loc_EventNMEA_Ind
 - loc.h, 1231
- unpack_loc_EventNMEA_Ind_t, 720
 - NMEADData, 720
 - ParamPresenceMask, 720
 - Tlvresult, 720
- unpack_loc_EventRegister
 - loc.h, 1231
- unpack_loc_EventRegister_t, 721
 - ParamPresenceMask, 721
 - Tlvresult, 721
- unpack_loc_EventTimeSyncCallback_Ind
 - loc.h, 1231
- unpack_loc_EventTimeSyncCallback_Ind_t, 721
 - ParamPresenceMask, 721
 - timeSyncRefCounter, 721
- unpack_loc_GetOpMode_Ind
 - loc.h, 1232
- unpack_loc_GetOpMode_Ind_t, 721
 - pMode, 723
 - ParamPresenceMask, 723
 - Status, 723
- unpack_loc_GetServer_Ind
 - loc.h, 1232
- unpack_loc_GetServer_Ind_t, 723
 - pIPv4AddrInfo, 724
 - pIPv6AddrInfo, 724
 - pURL, 724
- ParamPresenceMask, 724
 - serverStatus, 724
 - serverType, 724
 - Tlvresult, 724
- unpack_loc_GnssSvInfo_Ind
 - loc.h, 1232
- unpack_loc_GnssSvInfo_Ind_t, 724
 - altitudeAssumed, 725
 - pSatelliteInfo, 725
 - ParamPresenceMask, 725
 - Tlvresult, 725
- unpack_loc_InjectPositionCallback_Ind
 - loc.h, 1233
- unpack_loc_InjectPositionCallback_Ind_t, 725
 - ParamPresenceMask, 726
 - status, 726
- unpack_loc_InjectSensorDataCallback_Ind
 - loc.h, 1233
- unpack_loc_InjectSensorDataCallback_Ind_t, 726
 - injectSensorDataStatus, 727
 - pAccelSamplesAccepted, 727
 - pAccelTempSamplesAccepted, 727
 - pGyroSamplesAccepted, 728
 - pGyroTempSamplesAccepted, 728
 - pOpaqueIdentifier, 728
 - ParamPresenceMask, 728
- unpack_loc_InjectTimeSyncDataCallback_Ind
 - loc.h, 1233
- unpack_loc_InjectTimeSyncDataCallback_Ind_t, 728
 - injectTimeSyncStatus, 728
 - ParamPresenceMask, 728
- unpack_loc_InjectUTCTimeCallback_Ind
 - loc.h, 1234
- unpack_loc_InjectUTCTimeCallback_Ind_t, 728
 - ParamPresenceMask, 729
 - status, 729
- unpack_loc_PositionRpt_Ind
 - loc.h, 1234
- unpack_loc_PositionRpt_Ind_t, 729
 - pAltitudeAssumed, 735
 - pAltitudeWrtEllipsoid, 735
 - pAltitudeWrtMeanSeaLevel, 735
 - pFixId, 735
 - pGpsTime, 735
 - pHeading, 735
 - pHeadingUnc, 735
 - pHorConfidence, 735
 - pHorReliability, 735
 - pHorUncCircular, 735
 - pHorUncEllipseOrientAzimuth, 735
 - pHorUncEllipseSemiMajor, 735
 - pHorUncEllipseSemiMinor, 735
 - pLatitude, 735
 - pLeapSeconds, 735
 - pLongitude, 735
 - pMagneticDeviation, 735
 - pPrecisionDilution, 735
 - pSensorDataUsage, 735

- pSpeedHorizontal, 735
- pSpeedUnc, 735
- pSpeedVertical, 735
- pSvUsedforFix, 735
- pTechnologyMask, 735
- pTimeSrc, 735
- pTimeUnc, 735
- pTimestampUtc, 735
- pVertConfidence, 736
- pVertReliability, 736
- pVertUnc, 736
- ParamPresenceMask, 735
- sessionId, 736
- sessionStatus, 736
- Tlvresult, 736
- unpack_loc_SLQSLOCGetBestAvailPos
 - loc.h, 1236
- unpack_loc_SLQSLOCGetBestAvailPos_t, 740
 - ParamPresenceMask, 741
 - Tlvresult, 741
- unpack_loc_SLQSLOCGetOpMode
 - loc.h, 1237
- unpack_loc_SLQSLOCGetOpMode_t, 741
 - ParamPresenceMask, 741
 - Tlvresult, 741
- unpack_loc_SLQSLOCGetServer
 - loc.h, 1237
- unpack_loc_SLQSLOCGetServer_t
 - loc.h, 1223
- unpack_loc_SLQSLOCInjectPosition
 - loc.h, 1237
- unpack_loc_SLQSLOCInjectPosition_t
 - loc.h, 1223
- unpack_loc_SLQSLOCInjectSensorData
 - loc.h, 1238
- unpack_loc_SLQSLOCInjectSensorData_t
 - loc.h, 1223
- unpack_loc_SLQSLOCInjectUTCTime
 - loc.h, 1238
- unpack_loc_SLQSLOCInjectUTCTime_t
 - loc.h, 1223
- unpack_loc_SLQSLOCSetCradleMountConfig
 - loc.h, 1238
- unpack_loc_SLQSLOCSetCradleMountConfig_t
 - loc.h, 1223
- unpack_loc_SLQSLOCSetServer
 - loc.h, 1239
- unpack_loc_SensorStreamingCallback_Ind
 - loc.h, 1234
- unpack_loc_SensorStreamingCallback_Ind_t, 736
 - pAccelAcceptReady, 736
 - pAccelTempAcceptReady, 736
 - pGyroAcceptReady, 737
 - pGyroTempAcceptReady, 737
 - ParamPresenceMask, 736
- unpack_loc_SetExtPowerConfig_Ind
 - loc.h, 1235
- unpack_loc_SetExtPowerConfig_Ind_t, 737
 - ParamPresenceMask, 737
 - status, 737
 - Tlvresult, 737
- unpack_loc_SetExtPowerState
 - loc.h, 1235
- unpack_loc_SetExtPowerState_t, 737
 - ParamPresenceMask, 738
 - Tlvresult, 738
- unpack_loc_SetOperationMode
 - loc.h, 1235
- unpack_loc_SetOperationMode_Ind
 - loc.h, 1236
- unpack_loc_SetOperationMode_Ind_t, 738
 - ParamPresenceMask, 739
 - status, 739
 - Tlvresult, 739
- unpack_loc_SetOperationMode_t, 739
 - ParamPresenceMask, 739
 - Tlvresult, 739
- unpack_loc_SetServer_Ind
 - loc.h, 1236
- unpack_loc_SetServer_Ind_t, 739
 - ParamPresenceMask, 740
 - serverStatus, 740
 - Tlvresult, 740
- unpack_loc_Start
 - loc.h, 1239
- unpack_loc_Start_t, 741
 - ParamPresenceMask, 742
 - Tlvresult, 742
- unpack_loc_Stop
 - loc.h, 1239
- unpack_loc_Stop_t, 742
 - ParamPresenceMask, 742
 - Tlvresult, 742
- unpack_nas_GetACCOLC
 - nas.h, 1275
- unpack_nas_GetACCOLC_t, 742
 - pAccolc, 743
 - ParamPresenceMask, 743
- unpack_nas_GetANAAAAAuthenticationStatus
 - nas.h, 1275
- unpack_nas_GetANAAAAAuthenticationStatus_t, 743
 - pAuthStatus, 743
 - ParamPresenceMask, 743
- unpack_nas_GetCDMANetworkParameters
 - nas.h, 1276
- unpack_nas_GetCDMANetworkParameters_t, 743
 - Application, 745
 - Broadcast, 745
 - CustomSCP, 745
 - ForceRev0, 745
 - ParamPresenceMask, 745
 - Protocol, 745
 - RegForeignNID, 745
 - RegForeignSID, 745
 - RegHomeSID, 746
 - Roaming, 746

- SCI, [746](#)
- SCM, [746](#)
- unpack_nas_GetHomeNetwork
 - nas.h, [1276](#)
- unpack_nas_GetHomeNetwork3GPP2
 - nas.h, [1277](#)
- unpack_nas_GetHomeNetwork3GPP2_t, [746](#)
 - nameSize, [748](#)
 - pMCC, [748](#)
 - pMNC, [748](#)
 - pNID, [748](#)
 - pName, [748](#)
 - pNw2DescDisp, [748](#)
 - pNw2DescEnc, [748](#)
 - pNw2DescLen, [748](#)
 - pNw2MCC, [748](#)
 - pNw2MNC, [748](#)
 - pNw2Name, [748](#)
 - pSID, [748](#)
 - ParamPresenceMask, [748](#)
- unpack_nas_GetHomeNetwork_t, [748](#)
 - mcc, [749](#)
 - mnc, [749](#)
 - name, [749](#)
 - nid, [749](#)
 - ParamPresenceMask, [749](#)
 - sid, [749](#)
- unpack_nas_GetNetworkPreference
 - nas.h, [1277](#)
- unpack_nas_GetNetworkPreference_t, [749](#)
 - ActiveTechPref, [750](#)
 - Duration, [750](#)
 - ParamPresenceMask, [750](#)
 - PersistentTechPref, [750](#)
 - Tlvresult, [750](#)
- unpack_nas_GetRFInfo
 - nas.h, [1277](#)
- unpack_nas_GetRFInfo_t, [750](#)
 - instancesSize, [751](#)
 - ParamPresenceMask, [751](#)
 - RFBandInfoElements, [751](#)
- unpack_nas_GetServingNetwork
 - nas.h, [1278](#)
- unpack_nas_GetServingNetwork_t, [751](#)
 - CSDomain, [753](#)
 - DataCaps, [753](#)
 - DataCapsLen, [753](#)
 - MCC, [753](#)
 - MNC, [753](#)
 - Name, [753](#)
 - nameSize, [754](#)
 - PSDomain, [754](#)
 - ParamPresenceMask, [754](#)
 - RAN, [754](#)
 - Radiolfaces, [754](#)
 - RadiolfacesSize, [754](#)
 - RegistrationState, [754](#)
 - Roaming, [754](#)
- unpack_nas_GetServingNetworkCapabilities
 - nas.h, [1278](#)
- unpack_nas_GetServingNetworkCapabilities_t, [754](#)
 - DataCaps, [755](#)
 - DataCapsLen, [755](#)
 - ParamPresenceMask, [755](#)
- unpack_nas_GetSignalStrengths
 - nas.h, [1278](#)
- unpack_nas_GetSignalStrengths_t, [755](#)
 - len, [755](#)
 - ParamPresenceMask, [755](#)
 - radio, [755](#)
 - rsi, [755](#)
- unpack_nas_InitiateDomainAttach
 - nas.h, [1279](#)
- unpack_nas_InitiateDomainAttach_t
 - nas.h, [1252](#)
- unpack_nas_PerformNetworkScan
 - nas.h, [1279](#)
- unpack_nas_PerformNetworkScan_t, [756](#)
 - p3GppNetworkInfoInstances, [757](#)
 - p3GppNetworkInstanceSize, [757](#)
 - pLteOpModeTlv, [757](#)
 - pPCIInfo, [757](#)
 - pPDSInstance, [757](#)
 - pPDSInstanceSize, [757](#)
 - pRATInstance, [757](#)
 - pRATInstanceSize, [757](#)
 - pScanResult, [757](#)
 - ParamPresenceMask, [757](#)
- unpack_nas_SLQSCfgSigInfo
 - nas.h, [1283](#)
- unpack_nas_SLQSCfgSigInfo_t
 - nas.h, [1253](#)
- unpack_nas_SLQSGetErrorRate
 - nas.h, [1283](#)
- unpack_nas_SLQSGetErrorRate_t, [762](#)
 - pCDMAFrameErrRate, [763](#)
 - pGSMBER, [763](#)
 - pHDRPackErrRate, [763](#)
 - pWCDMABER, [763](#)
 - ParamPresenceMask, [763](#)
- unpack_nas_SLQSGetNetworkTime
 - nas.h, [1284](#)
- unpack_nas_SLQSGetNetworkTime_t, [764](#)
 - p3GPP2TimeInfo, [764](#)
 - p3GPPTimeInfo, [764](#)
 - ParamPresenceMask, [764](#)
- unpack_nas_SLQSGetOperatorNameData
 - nas.h, [1284](#)
- unpack_nas_SLQSGetOperatorNameData_t, [764](#)
 - pNITZInformation, [765](#)
 - pOperatorNameString, [765](#)
 - pOperatorPLMNList, [765](#)
 - pPLMNNetworkName, [765](#)
 - pSvcProviderName, [765](#)
 - ParamPresenceMask, [765](#)
- unpack_nas_SLQSGetPLMNName

- nas.h, [1285](#)
- unpack_nas_SLQSGetPLMNName_t, [766](#)
 - longName, [768](#)
 - longNameCI, [768](#)
 - longNameEn, [768](#)
 - longNameLen, [768](#)
 - longNameSB, [768](#)
 - ParamPresenceMask, [768](#)
 - shortName, [768](#)
 - shortNameCI, [768](#)
 - shortNameEn, [768](#)
 - shortNameLen, [768](#)
 - shortNameSB, [768](#)
 - spn, [768](#)
 - spnEncoding, [768](#)
 - spnLength, [769](#)
- unpack_nas_SLQSGetServingSystem
 - nas.h, [1285](#)
- unpack_nas_SLQSGetServingSystem_t, [769](#)
 - BasestationID, [772](#)
 - BasestationLatitude, [772](#)
 - BasestationLongitude, [772](#)
 - CDMASystemInfoExt, [772](#)
 - CallBarStatus, [772](#)
 - CellID, [772](#)
 - ConcSvcInfo, [773](#)
 - CurrentPLMN, [773](#)
 - DTMInd, [773](#)
 - DataSrvCapabilities, [773](#)
 - DefaultRoamInd, [773](#)
 - DetailedSvcInfo, [773](#)
 - Gpp2TimeZone, [773](#)
 - GppNetworkDSTAdjustment, [773](#)
 - GppTimeZone, [773](#)
 - HdrPersonality, [773](#)
 - Lac, [773](#)
 - NetworkID, [773](#)
 - PRLInd, [773](#)
 - ParamPresenceMask, [773](#)
 - RoamIndicatorVal, [773](#)
 - RoamingIndicatorList, [773](#)
 - ServingSystem, [773](#)
 - SystemID, [773](#)
 - TrackAreaCode, [773](#)
- unpack_nas_SLQSGetSignalStrength
 - nas.h, [1285](#)
- unpack_nas_SLQSGetSignalStrength_t, [773](#)
 - ecioList, [775](#)
 - ecioListLen, [775](#)
 - errorRateList, [775](#)
 - errorRateListLen, [775](#)
 - lo, [775](#)
 - ltersrp, [776](#)
 - ltesnr, [776](#)
 - ParamPresenceMask, [776](#)
 - rsrqInfo, [776](#)
 - rxSignalStrengthList, [776](#)
 - rxSignalStrengthListLen, [776](#)
 - signalStrengthReqMask, [776](#)
 - sinr, [776](#)
- unpack_nas_SLQSGetSysInfo
 - nas.h, [1286](#)
- unpack_nas_SLQSGetSysInfo_t, [776](#)
 - pAddCDMASysInfo, [779](#)
 - pAddGSMSysInfo, [779](#)
 - pAddHDRSysInfo, [779](#)
 - pAddLTESysInfo, [779](#)
 - pCDMASrvStatusInfo, [779](#)
 - pCDMASysInfo, [779](#)
 - pCampedCiotLteOpMode, [779](#)
 - pGSMCallBarringSysInfo, [779](#)
 - pGSMCipherDomainSysInfo, [779](#)
 - pGSMSrvStatusInfo, [779](#)
 - pGSMSysInfo, [779](#)
 - pHDRSrvStatusInfo, [779](#)
 - pHDRSysInfo, [779](#)
 - pLTESrvStatusInfo, [779](#)
 - pLTESysInfo, [779](#)
 - pLTEVoiceSupportSysInfo, [779](#)
 - pWCDMASysInfo, [780](#)
 - ParamPresenceMask, [779](#)
- unpack_nas_SLQSGetSysSelectionPref
 - nas.h, [1286](#)
- unpack_nas_SLQSGetSysSelectionPref_t, [780](#)
 - pBandPref, [784](#)
 - pEmerMode, [784](#)
 - pGWAcqOrderPref, [784](#)
 - pLTEBandPref, [784](#)
 - pModePref, [784](#)
 - pNetSelPref, [784](#)
 - pPRLPref, [784](#)
 - pRoamPref, [784](#)
 - pSrvDomainPref, [784](#)
 - ParamPresenceMask, [784](#)
- unpack_nas_SLQSGetSysSelectionPrefExt
 - nas.h, [1287](#)
- unpack_nas_SLQSGetSysSelectionPrefExt_t, [784](#)
 - pAcqOrderPref, [786](#)
 - pBandPref, [786](#)
 - pCiotAcqOrderPref, [786](#)
 - pCiotLteOpModePref, [786](#)
 - pEmerMode, [787](#)
 - pGWAcqOrderPref, [787](#)
 - pLTEBandPref, [787](#)
 - pLteM1BandPref, [787](#)
 - pLteNb1BandPref, [787](#)
 - pModePref, [787](#)
 - pNetSelPref, [787](#)
 - pNr5gBandPref, [787](#)
 - pPRLPref, [787](#)
 - pRatDisabledMask, [787](#)
 - pRoamPref, [787](#)
 - pSrvDomainPref, [787](#)
- unpack_nas_SLQSIInitiateNetworkRegistration
 - nas.h, [1287](#)
- unpack_nas_SLQSIInitiateNetworkRegistration_t

- nas.h, [1253](#)
- unpack_nas_SLQSNASGetForbiddenNetworks
 - nas.h, [1290](#)
- unpack_nas_SLQSNASGetForbiddenNetworks_t, [793](#)
 - ParamPresenceMask, [794](#)
- unpack_nas_SLQSNASGetDRXParams
 - nas.h, [1289](#)
- unpack_nas_SLQSNASGetDRXParams_t, [791](#)
 - pCycleLen, [792](#)
 - pEdrxEnable, [792](#)
 - ParamPresenceMask, [792](#)
- unpack_nas_SLQSNASGetDRXParamsExt
 - nas.h, [1289](#)
- unpack_nas_SLQSNASGetDRXParamsExt_t, [792](#)
- unpack_nas_SLQSNASSetDRXParams
 - nas.h, [1293](#)
- unpack_nas_SLQSNASSetDRXParams_t
 - nas.h, [1253](#)
- unpack_nas_SLQSNASSwiGetChannelLock
 - nas.h, [1293](#)
- unpack_nas_SLQSNASSwiGetChannelLock_t, [803](#)
 - ParamPresenceMask, [803](#)
- unpack_nas_SLQSNASSwiSetChannelLock
 - nas.h, [1295](#)
- unpack_nas_SLQSNASSwiSetChannelLock_t
 - nas.h, [1253](#)
- unpack_nas_SLQSNasConfigSigInfo2
 - nas.h, [1287](#)
- unpack_nas_SLQSNasConfigSigInfo2_t
 - nas.h, [1253](#)
- unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind
 - nas.h, [1288](#)
- unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t, [787](#)
- unpack_nas_SLQSNasGet3GPP2Subscription
 - nas.h, [1288](#)
- unpack_nas_SLQSNasGet3GPP2Subscription_t, [788](#)
 - pDirNum, [789](#)
 - ParamPresenceMask, [789](#)
- unpack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [1288](#)
- unpack_nas_SLQSNasGetCellLocationInfo_t, [789](#)
 - pCDMAInfo, [791](#)
 - pLTEInfoInterfreq, [791](#)
 - pLTEInfoIntrafreq, [791](#)
 - pUMTSInfo, [791](#)
 - ParamPresenceMask, [790](#)
- unpack_nas_SLQSNasGetHDRColorCode
 - nas.h, [1290](#)
- unpack_nas_SLQSNasGetHDRColorCode_t, [794](#)
 - pColorCode, [794](#)
 - ParamPresenceMask, [794](#)
- unpack_nas_SLQSNasGetRFInfo
 - nas.h, [1290](#)
- unpack_nas_SLQSNasGetRFInfo_t, [794](#)
 - pLTEOperationMode, [795](#)
 - pRfBandInfoExtFormat, [795](#)
 - pRfBandwidthInfo, [795](#)
 - pRfDedicatedBandInfo, [795](#)
 - rffbandInfoList, [795](#)
 - Tlvresult, [795](#)
- unpack_nas_SLQSNasGetSigInfo
 - nas.h, [1291](#)
- unpack_nas_SLQSNasGetSigInfo_t, [795](#)
 - CDMASSInfo, [796](#)
 - GSMSSInfo, [796](#)
 - HDRSSInfo, [796](#)
 - LTSSInfo, [796](#)
 - ParamPresenceMask, [796](#)
- unpack_nas_SLQSNasGetTxRxInfo
 - nas.h, [1291](#)
- unpack_nas_SLQSNasGetTxRxInfo_t, [796](#)
 - pRXChain0Info, [797](#)
 - pRXChain1Info, [797](#)
 - pTXInfo, [797](#)
 - ParamPresenceMask, [797](#)
- unpack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [1291](#)
- unpack_nas_SLQSNasIndicationRegisterExt_t
 - nas.h, [1253](#)
- unpack_nas_SLQSNasNetworkRejectCallback_Ind
 - nas.h, [1292](#)
- unpack_nas_SLQSNasNetworkRejectCallback_Ind_t, [797](#)
 - pCsgId, [800](#)
 - pLteOpMode, [800](#)
 - pPlmnId, [800](#)
 - ParamPresenceMask, [800](#)
 - radiolf, [800](#)
 - rejectCause, [800](#)
 - serviceDomain, [800](#)
- unpack_nas_SLQSNasNetworkTimeCallBack_ind
 - nas.h, [1292](#)
- unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, [800](#)
 - pDayltSavAdj, [801](#)
 - pRadioInterface, [801](#)
 - pTimeZone, [801](#)
 - ParamPresenceMask, [801](#)
 - universalTime, [801](#)
- unpack_nas_SLQSNasSigInfoCallback_ind
 - nas.h, [1293](#)
- unpack_nas_SLQSNasSigInfoCallback_ind_t, [801](#)
 - pRscp, [802](#)
 - ParamPresenceMask, [802](#)
- unpack_nas_SLQSNasSwiIndicationRegister
 - nas.h, [1294](#)
- unpack_nas_SLQSNasSwiIndicationRegister_t
 - nas.h, [1253](#)
- unpack_nas_SLQSNasSwiModemStatus
 - nas.h, [1294](#)
- unpack_nas_SLQSNasSwiModemStatus_t, [803](#)
 - commonInfo, [804](#)
 - pLTEInfo, [804](#)
 - ParamPresenceMask, [804](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind

- nas.h, [1294](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, [804](#)
- Info, [805](#)
- Tlvresult, [805](#)
- unpack_nas_SLQSNasSysInfoCallback_ind nas.h, [1295](#)
- unpack_nas_SLQSNasTimerCallback_ind nas.h, [1295](#)
- unpack_nas_SLQSNasTimerCallback_ind_t, [805](#)
- ParamPresenceMask, [805](#)
- t3396_apn, [805](#)
- t3396_plmn_id, [805](#)
- t3396_val, [805](#)
- unpack_nas_SLQSSetBandPreference nas.h, [1296](#)
- unpack_nas_SLQSSetBandPreference_t nas.h, [1253](#)
- unpack_nas_SLQSSetSignalStrengthsCallback nas.h, [1296](#)
- unpack_nas_SLQSSetSignalStrengthsCallback_t nas.h, [1253](#)
- unpack_nas_SLQSSetSysSelectionPref nas.h, [1296](#)
- unpack_nas_SLQSSetSysSelectionPref_t nas.h, [1253](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind nas.h, [1297](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, [805](#)
- Info, [806](#)
- Tlvresult, [806](#)
- unpack_nas_SLQSSetSysSelectionPrefExt nas.h, [1297](#)
- unpack_nas_SLQSSetSysSelectionPrefExt_t nas.h, [1253](#)
- unpack_nas_SLQSSwiGetHDRPersonality nas.h, [1297](#)
- unpack_nas_SLQSSwiGetHDRPersonality_t, [806](#)
- pCurrentPersonality, [807](#)
- pPersonalityListLength, [807](#)
- pProtocolSubtypeElement, [807](#)
- ParamPresenceMask, [807](#)
- unpack_nas_SLQSSwiGetHDRProtSubtype nas.h, [1298](#)
- unpack_nas_SLQSSwiGetHDRProtSubtype_t, [807](#)
- pAppSubType, [808](#)
- pCurrentPrsnlty, [808](#)
- pPersonalityListLength, [808](#)
- ParamPresenceMask, [808](#)
- unpack_nas_SLQSSwiGetHRPDStats nas.h, [1298](#)
- unpack_nas_SLQSSwiGetHRPDStats_t, [808](#)
- pPilotSetData, [809](#)
- pUATI, [809](#)
- ParamPresenceMask, [809](#)
- unpack_nas_SLQSSwiGetLteCQI nas.h, [1298](#)
- unpack_nas_SLQSSwiGetLteCQI_t, [809](#)
- ParamPresenceMask, [810](#)
- ValidityCW0, [810](#)
- ValidityCW1, [810](#)
- unpack_nas_SLQSSwiGetLteScsRxInfo nas.h, [1299](#)
- unpack_nas_SLQSSwiGetLteScsRxInfo_t, [810](#)
- pScsRxInfo, [810](#)
- ParamPresenceMask, [810](#)
- unpack_nas_SLQSSwiHDRPersonalityCallback_Ind nas.h, [1299](#)
- unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t, [810](#)
- unpack_nas_SLQSSwiNetworkDebug nas.h, [1299](#)
- unpack_nas_SLQSSwiNetworkDebug_t, [811](#)
- pDataStatusDetail, [812](#)
- pDeviceConfigDetail, [812](#)
- pNetworkStat1x, [812](#)
- pNetworkStatEVDO, [812](#)
- pObjectVer, [812](#)
- ParamPresenceMask, [812](#)
- unpack_nas_SLQSSwiPSDetach nas.h, [1300](#)
- unpack_nas_SLQSSwiPSDetach_t nas.h, [1253](#)
- unpack_nas_SLQSSwiRandIndicatorCallback_Ind nas.h, [1300](#)
- unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t, [812](#)
- ParamPresenceMask, [812](#)
- rankIndicatorTlv, [812](#)
- unpack_nas_SLQSSysInfoCallback_ind_t, [812](#)
- pGSMSysInfo, [816](#)
- pHDRSysInfo, [816](#)
- pLTESysInfo, [816](#)
- pLteCiotOpModeTlv, [816](#)
- pNR5GCellStatus, [816](#)
- pSysInfoNoChange, [816](#)
- ParamPresenceMask, [816](#)
- unpack_nas_SetACCOLC nas.h, [1279](#)
- unpack_nas_SetACCOLC_t nas.h, [1253](#)
- unpack_nas_SetCDMANetworkParameters nas.h, [1280](#)
- unpack_nas_SetCDMANetworkParameters_t nas.h, [1253](#)
- unpack_nas_SetDataCapabilitiesCallback_ind nas.h, [1280](#)
- unpack_nas_SetDataCapabilitiesCallback_ind_t, [757](#)
- dataCaps, [758](#)
- dataCapsSize, [758](#)
- ParamPresenceMask, [758](#)
- unpack_nas_SetEventReportInd nas.h, [1280](#)
- unpack_nas_SetEventReportInd_t, [758](#)
- ParamPresenceMask, [759](#)

- RFTlv, [759](#)
- RRTlv, [759](#)
- SLQSSSTlv, [759](#)
- SSTlv, [759](#)
- unpack_nas_SetLURejectCallback
 - nas.h, [1281](#)
- unpack_nas_SetLURejectCallback_t
 - nas.h, [1253](#)
- unpack_nas_SetNasLTECphyCaIndCallback_ind
 - nas.h, [1281](#)
- unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [759](#)
 - ParamPresenceMask, [760](#)
- unpack_nas_SetNetworkPreference
 - nas.h, [1282](#)
- unpack_nas_SetNetworkPreference_t, [760](#)
 - ParamPresenceMask, [760](#)
 - Tlvresult, [761](#)
- unpack_nas_SetRFInfoCallback
 - nas.h, [1282](#)
- unpack_nas_SetRFInfoCallback_t
 - nas.h, [1253](#)
- unpack_nas_SetRoamingIndicatorCallback_ind
 - nas.h, [1282](#)
- unpack_nas_SetRoamingIndicatorCallback_ind_t, [761](#)
 - ParamPresenceMask, [761](#)
 - roaming, [761](#)
- unpack_nas_SetServingSystemCallback_ind
 - nas.h, [1283](#)
- unpack_nas_SetServingSystemCallback_ind_t, [761](#)
 - ParamPresenceMask, [762](#)
 - SSInfo, [762](#)
 - Tlvresult, [762](#)
- unpack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [1284](#)
- unpack_nas_SlqsGetLTECphyCAInfo_t, [763](#)
 - LTECphyCAInfo, [764](#)
 - ParamPresenceMask, [764](#)
 - Tlvresult, [764](#)
- unpack_omaDmConfigTlv_t, [816](#)
 - alertmsg, [817](#)
 - alertmsglength, [817](#)
 - state, [817](#)
 - userInputReq, [817](#)
 - userInputTimeout, [817](#)
- unpack_omaDmFotaTlv_t, [817](#)
 - description, [819](#)
 - descriptionlength, [819](#)
 - fwloadsize, [819](#)
 - fwloadComplete, [819](#)
 - namelength, [819](#)
 - package_name, [819](#)
 - sessionType, [819](#)
 - severity, [819](#)
 - state, [819](#)
 - updateCompleteStatus, [819](#)
 - userInputReq, [819](#)
 - userInputTimeout, [819](#)
 - version, [819](#)
 - versionlength, [819](#)
- unpack_omaDmNotificationsTlv_t, [819](#)
 - notification, [820](#)
 - sessionStatus, [820](#)
- unpack_pds_ForceXTRADownload
 - pds.h, [1315](#)
- unpack_pds_ForceXTRADownload_t
 - pds.h, [1305](#)
- unpack_pds_GetPDSDDefaults
 - pds.h, [1315](#)
- unpack_pds_GetPDSDDefaults_t, [820](#)
 - pAccuracy, [821](#)
 - pInterval, [821](#)
 - pOperation, [821](#)
 - pTimeout, [821](#)
 - ParamPresenceMask, [821](#)
- unpack_pds_GetPDSSState
 - pds.h, [1316](#)
- unpack_pds_GetPDSSState_t, [821](#)
 - pEnabledStatus, [822](#)
 - pTrackingStatus, [822](#)
 - ParamPresenceMask, [822](#)
- unpack_pds_GetPortAutomaticTracking
 - pds.h, [1316](#)
- unpack_pds_GetPortAutomaticTracking_t, [822](#)
 - ParamPresenceMask, [822](#)
 - pbAuto, [823](#)
- unpack_pds_GetServiceAutomaticTracking
 - pds.h, [1316](#)
- unpack_pds_GetServiceAutomaticTracking_t, [823](#)
 - ParamPresenceMask, [823](#)
 - pbAuto, [823](#)
- unpack_pds_GetXTRAAutomaticDownload
 - pds.h, [1317](#)
- unpack_pds_GetXTRAAutomaticDownload_t, [823](#)
 - pInterval, [824](#)
 - ParamPresenceMask, [824](#)
 - pbEnabled, [824](#)
- unpack_pds_GetXTRANetwork
 - pds.h, [1317](#)
- unpack_pds_GetXTRANetwork_t, [824](#)
 - pPreference, [824](#)
 - ParamPresenceMask, [824](#)
- unpack_pds_GetXTRAValidity
 - pds.h, [1318](#)
- unpack_pds_GetXTRAValidity_t, [824](#)
 - pDuration, [825](#)
 - pGPSWeek, [825](#)
 - pGPSWeekOffset, [825](#)
 - ParamPresenceMask, [825](#)
- unpack_pds_PDSInjectTimeReference
 - pds.h, [1318](#)
- unpack_pds_PDSInjectTimeReference_t
 - pds.h, [1305](#)
- unpack_pds_ResetPDSDData
 - pds.h, [1318](#)
- unpack_pds_ResetPDSDData_t
 - pds.h, [1305](#)

- unpack_pds_SLQSGetAGPSConfig
 - pds.h, [1322](#)
- unpack_pds_SLQSGetAGPSConfig_t, [828](#)
 - pServerAddress, [829](#)
 - pServerPort, [829](#)
 - pServerURL, [829](#)
 - ParamPresenceMask, [829](#)
- unpack_pds_SLQSGetGPSStateInfo
 - pds.h, [1322](#)
- unpack_pds_SLQSGetGPSStateInfo_t, [829](#)
 - Altitude, [833](#)
 - EngineState, [833](#)
 - HorizontalUncertainty, [833](#)
 - lono_valid, [833](#)
 - Latitude, [833](#)
 - Longitude, [833](#)
 - ParamPresenceMask, [833](#)
 - Time_uncert_ms, [833](#)
 - ValidMask, [833](#)
 - VerticalUncertainty, [833](#)
- unpack_pds_SLQSPDSDeterminePosition
 - pds.h, [1322](#)
- unpack_pds_SLQSPDSDeterminePosition_t
 - pds.h, [1306](#)
- unpack_pds_SLQSPDSInjectAbsoluteTimeReference
 - pds.h, [1323](#)
- unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t
 - pds.h, [1306](#)
- unpack_pds_SLQSPDSInjectPositionData
 - pds.h, [1323](#)
- unpack_pds_SLQSPDSInjectPositionData_t
 - pds.h, [1306](#)
- unpack_pds_SLQSSetAGPSConfig
 - pds.h, [1323](#)
- unpack_pds_SLQSSetAGPSConfig_t
 - pds.h, [1306](#)
- unpack_pds_SLQSSetPositionMethodState
 - pds.h, [1324](#)
- unpack_pds_SLQSSetPositionMethodState_t
 - pds.h, [1306](#)
- unpack_pds_SetEventReport_Ind
 - pds.h, [1319](#)
- unpack_pds_SetEventReport_Ind_t, [825](#)
 - dLatitude, [827](#)
 - dLongitude, [827](#)
 - has_PositionDataNMEA, [827](#)
 - has_SessionStatus, [827](#)
 - has_dLatitude, [827](#)
 - has_dLongitude, [827](#)
 - has_posSrc, [827](#)
 - ParamPresenceMask, [827](#)
 - posSrc, [827](#)
 - PositionDataNMEA, [827](#)
 - SessionStatus, [827](#)
- unpack_pds_SetEventReportCallback
 - pds.h, [1319](#)
- unpack_pds_SetEventReportCallback_t
 - pds.h, [1305](#)
- unpack_pds_SetPDSDefaults
 - pds.h, [1319](#)
- unpack_pds_SetPDSDefaults_t
 - pds.h, [1305](#)
- unpack_pds_SetPDSState
 - pds.h, [1320](#)
- unpack_pds_SetPDSState_t
 - pds.h, [1305](#)
- unpack_pds_SetPdsState_Ind
 - pds.h, [1320](#)
- unpack_pds_SetPdsState_Ind_t, [827](#)
 - EnabledStatus, [828](#)
 - ParamPresenceMask, [828](#)
 - TlvPresent, [828](#)
 - TrackingStatus, [828](#)
- unpack_pds_SetPortAutomaticTracking
 - pds.h, [1320](#)
- unpack_pds_SetPortAutomaticTracking_t
 - pds.h, [1305](#)
- unpack_pds_SetServiceAutomaticTracking
 - pds.h, [1321](#)
- unpack_pds_SetServiceAutomaticTracking_t
 - pds.h, [1306](#)
- unpack_pds_SetXTRAAutomaticDownload
 - pds.h, [1321](#)
- unpack_pds_SetXTRAAutomaticDownload_t
 - pds.h, [1306](#)
- unpack_pds_SetXTRANetwork
 - pds.h, [1321](#)
- unpack_pds_SetXTRANetwork_t
 - pds.h, [1306](#)
- unpack_pds_StartPDSTrackingSessionExt
 - pds.h, [1324](#)
- unpack_pds_StartPDSTrackingSessionExt_t
 - pds.h, [1306](#)
- unpack_pds_StopPDSTrackingSession
 - pds.h, [1324](#)
- unpack_pds_StopPDSTrackingSession_t
 - pds.h, [1306](#)
- unpack_qmi_t, [834](#)
 - msgid, [834](#)
 - type, [834](#)
 - xid, [834](#)
- unpack_qos_BindDataPort
 - qos.h, [1377](#)
- unpack_qos_BindDataPort_t, [834](#)
 - ParamPresenceMask, [834](#)
- unpack_qos_IPv4Addr_t, [835](#)
 - addr, [835](#)
 - subnetMask, [835](#)
- unpack_qos_IPv6Addr_t, [835](#)
 - addr, [836](#)
 - prefixLen, [836](#)
- unpack_qos_IPv6TrafCls_t, [836](#)
 - mask, [836](#)
 - val, [836](#)
- unpack_qos_Port_t, [837](#)
 - port, [837](#)

- range, 837
- unpack_qos_QosFlowInfo_t, 837
 - BearerID, 838
 - is_RxQFlowGranted_Available, 838
 - is_TxQFlowGranted_Available, 838
 - NumRxFilters, 838
 - NumTxFilters, 839
 - QFlowState, 839
 - RxQFilter, 839
 - RxQFlowGranted, 839
 - TxQFilter, 839
 - TxQFlowGranted, 839
- unpack_qos_QosFlowInfoState_t, 839
 - id, 839
 - isNewFlow, 839
 - state, 839
- unpack_qos_SLQSQosGetNetworkStatus
 - qos.h, 1377
- unpack_qos_SLQSQosGetNetworkStatus_t, 839
 - NWQoSStatus, 840
 - ParamPresenceMask, 840
- unpack_qos_SLQSQosSwiReadApnExtraParams
 - qos.h, 1378
- unpack_qos_SLQSQosSwiReadApnExtraParams_t, 840
 - ambr_dl, 841
 - ambr_dl_ext, 841
 - ambr_dl_ext2, 841
 - ambr_ul, 841
 - ambr_ul_ext, 841
 - ambr_ul_ext2, 841
 - apnId, 841
 - ParamPresenceMask, 841
- unpack_qos_SLQSQosSwiReadDataStats
 - qos.h, 1378
- unpack_qos_SLQSQosSwiReadDataStats_t, 842
 - apnId, 843
 - numQosFlow, 843
 - ParamPresenceMask, 843
 - qosFlow, 843
 - total_rx_bytes, 843
 - total_rx_pkt, 843
 - total_tx_bytes, 843
 - total_tx_pkt, 843
- unpack_qos_SLQSSetQosEventCallback
 - qos.h, 1379
- unpack_qos_SLQSSetQosEventCallback_ind
 - qos.h, 1379
- unpack_qos_SLQSSetQosEventCallback_ind_t, 843
 - NumFlows, 844
 - ParamPresenceMask, 844
 - QosFlowInfo, 844
- unpack_qos_SLQSSetQosEventCallback_t, 844
 - ParamPresenceMask, 844
- unpack_qos_SLQSSetQosNWStatusCallback_ind
 - qos.h, 1380
- unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 844
 - status, 845
- unpack_qos_SLQSSetQosPriEventCallback_ind
 - qos.h, 1380
- unpack_qos_SLQSSetQosPriEventCallback_ind_t, 845
 - event, 845
 - ParamPresenceMask, 845
- unpack_qos_SLQSSetQosStatusCallback_ind
 - qos.h, 1381
- unpack_qos_SLQSSetQosStatusCallback_ind_t, 845
 - event, 847
 - id, 847
 - ParamPresenceMask, 847
 - reason, 847
 - status, 847
- unpack_qos_Tos_t, 855
 - mask, 855
 - val, 855
- unpack_qos_dataRate_t, 835
 - dataRateMax, 835
 - guaranteedRate, 835
- unpack_qos_pktErrRate_t, 836
 - exponent, 837
 - multiplier, 837
- unpack_qos_swiQosFilter_t, 847
 - EspSpi, 849
 - IPv4DstAddr, 849
 - IPv4SrcAddr, 849
 - IPv4Tos, 849
 - IPv6DstAddr, 849
 - IPv6Label, 849
 - IPv6SrcAddr, 849
 - IPv6TrafCls, 849
 - Id, 849
 - index, 849
 - is_EspSpi_Available, 849
 - is_IPv4DstAddr_Available, 849
 - is_IPv4SrcAddr_Available, 849
 - is_IPv4Tos_Available, 849
 - is_IPv6DstAddr_Available, 849
 - is_IPv6Label_Available, 849
 - is_IPv6SrcAddr_Available, 849
 - is_IPv6TrafCls_Available, 849
 - is_Id_Available, 849
 - is_NxtHdrProto_Available, 850
 - is_Precedence_Available, 850
 - is_TCPDstPort_Available, 850
 - is_TCPSrcPort_Available, 850
 - is_TranDstPort_Available, 850
 - is_TranSrcPort_Available, 850
 - is_UDPdstPort_Available, 850
 - is_UDPSrcPort_Available, 850
 - NxtHdrProto, 850
 - Precedence, 850
 - TCPDstPort, 850
 - TCPSrcPort, 850
 - TranDstPort, 850
 - TranSrcPort, 850
 - UDPDstPort, 850

- UDPSrcPort, [850](#)
- version, [850](#)
- unpack_qos_swiQosFlow_t, [850](#)
 - DataRate, [853](#)
 - index, [853](#)
 - is_DataRate_Available, [853](#)
 - is_Jitter_Available, [853](#)
 - is_Latency_Available, [853](#)
 - is_LteQci_Available, [853](#)
 - is_MaxAllowedPktSz_Available, [853](#)
 - is_MinPolicedPktSz_Available, [853](#)
 - is_PktErrRate_Available, [853](#)
 - is_ProfileId3GPP2_Available, [853](#)
 - is-TokenBucket_Available, [853](#)
 - is_TrafficClass_Available, [853](#)
 - is_val_3GPP2Pri_Available, [853](#)
 - is_val_3GPPImCn_Available, [853](#)
 - is_val_3GPPSigInd_Available, [854](#)
 - Jitter, [854](#)
 - Latency, [854](#)
 - LteQci, [854](#)
 - MaxAllowedPktSz, [854](#)
 - MinPolicedPktSz, [854](#)
 - PktErrRate, [854](#)
 - ProfileId3GPP2, [854](#)
 - TokenBucket, [854](#)
 - TrafficClass, [854](#)
 - val_3GPP2Pri, [854](#)
 - val_3GPPImCn, [854](#)
 - val_3GPPResResidualBER, [854](#)
 - val_3GPPSigInd, [854](#)
 - val_3GPPTraHdIPri, [854](#)
- unpack_qos_tokenBucket_t, [854](#)
 - bucketSz, [855](#)
 - peakRate, [855](#)
 - tokenRate, [855](#)
- unpack_result_code_only
 - common.h, [1120](#)
- unpack_result_t, [856](#)
 - ParamPresenceMask, [856](#)
 - Tlvresult, [856](#)
- unpack_rms_GetSMSWake
 - rms.h, [1383](#)
- unpack_rms_GetSMSWake_t, [857](#)
 - enabled, [857](#)
 - ParamPresenceMask, [857](#)
 - wake_mask, [857](#)
- unpack_rms_SetSMSWake
 - rms.h, [1383](#)
- unpack_rms_SetSMSWake_t, [857](#)
 - ParamPresenceMask, [858](#)
- unpack_sar_SLQSSetRfSarState
 - sar.h, [1385](#)
- unpack_sar_SLQSGetRfSarState_t, [859](#)
 - pSarRFState, [859](#)
 - ParamPresenceMask, [859](#)
- unpack_sar_SLQSSetRfSarState
 - sar.h, [1385](#)
- unpack_sar_SLQSSetRfSarState_t
 - sar.h, [1384](#)
- unpack_sms_GetSMSCAddress
 - sms.h, [1398](#)
- unpack_sms_GetSMSCAddress_t, [859](#)
 - addressSize, [860](#)
 - pSMSCAddress, [860](#)
 - pSMSCType, [860](#)
 - ParamPresenceMask, [860](#)
 - typeSize, [860](#)
- unpack_sms_SLQSDeleteSMS
 - sms.h, [1400](#)
- unpack_sms_SLQSDeleteSMS_t, [864](#)
 - ParamPresenceMask, [864](#)
- unpack_sms_SLQSGetIndicationRegister
 - sms.h, [1400](#)
- unpack_sms_SLQSGetIndicationRegister_t, [864](#)
 - pGetIndicationRegInfo, [865](#)
 - ParamPresenceMask, [865](#)
- unpack_sms_SLQSGetMessageWaiting
 - sms.h, [1401](#)
- unpack_sms_SLQSGetMessageWaiting_t, [865](#)
 - pGetMsgWaitingInfoResp, [865](#)
 - ParamPresenceMask, [865](#)
- unpack_sms_SLQSGetSMS
 - sms.h, [1401](#)
- unpack_sms_SLQSGetSMS_t, [865](#)
 - message, [866](#)
 - messageFormat, [866](#)
 - messageSize, [866](#)
 - messageTag, [866](#)
 - ParamPresenceMask, [866](#)
- unpack_sms_SLQSGetSMSList
 - sms.h, [1402](#)
- unpack_sms_SLQSGetSMSList_t, [867](#)
 - messageList, [868](#)
 - messageListSize, [868](#)
 - ParamPresenceMask, [868](#)
- unpack_sms_SLQSGetSmsBroadcastConfig
 - sms.h, [1401](#)
- unpack_sms_SLQSGetSmsBroadcastConfig_t, [866](#)
 - pBroadcastConfig, [867](#)
 - pCDMABroadcastConfig, [867](#)
 - ParamPresenceMask, [867](#)
- unpack_sms_SLQSGetTransLayerInfo
 - sms.h, [1402](#)
- unpack_sms_SLQSGetTransLayerInfo_t, [868](#)
 - pGetTransLayerInfo, [868](#)
 - ParamPresenceMask, [868](#)
- unpack_sms_SLQSGetTransNWRegInfo
 - sms.h, [1403](#)
- unpack_sms_SLQSGetTransNWRegInfo_t, [868](#)
 - ParamPresenceMask, [869](#)
- unpack_sms_SLQSModifySMSStatus
 - sms.h, [1403](#)
- unpack_sms_SLQSModifySMSStatus_t, [869](#)
 - ParamPresenceMask, [869](#)
- unpack_sms_SLQSNWRegInfoCallback_ind

- sms.h, [1403](#)
- unpack_sms_SLQSNWRegInfoCallback_ind_t, [869](#)
 - NWRegStat, [870](#)
 - ParamPresenceMask, [870](#)
- unpack_sms_SLQSSendAsyncSMS
 - sms.h, [1404](#)
- unpack_sms_SLQSSendAsyncSMS_t, [870](#)
 - ParamPresenceMask, [870](#)
- unpack_sms_SLQSSetIndicationRegister
 - sms.h, [1404](#)
- unpack_sms_SLQSSetIndicationRegister_t, [870](#)
 - ParamPresenceMask, [871](#)
- unpack_sms_SLQSSetSmsBroadcastActivation
 - sms.h, [1404](#)
- unpack_sms_SLQSSetSmsBroadcastActivation_t, [871](#)
 - ParamPresenceMask, [871](#)
- unpack_sms_SLQSSetSmsBroadcastConfig
 - sms.h, [1405](#)
- unpack_sms_SLQSSetSmsBroadcastConfig_t, [871](#)
 - ParamPresenceMask, [872](#)
- unpack_sms_SLQSSetSmsStorage
 - sms.h, [1405](#)
- unpack_sms_SLQSSetSmsStorage_t, [872](#)
 - ParamPresenceMask, [872](#)
- unpack_sms_SLQSSmsGetMaxStorageSize
 - sms.h, [1405](#)
- unpack_sms_SLQSSmsGetMaxStorageSize_t, [872](#)
 - pMaxStorageSizeResp, [873](#)
 - ParamPresenceMask, [872](#)
- unpack_sms_SLQSSmsGetMessageProtocol
 - sms.h, [1406](#)
- unpack_sms_SLQSSmsGetMessageProtocol_t, [873](#)
 - pMessageProtocol, [873](#)
 - ParamPresenceMask, [873](#)
- unpack_sms_SLQSSmsSetRoutes
 - sms.h, [1406](#)
- unpack_sms_SLQSSmsSetRoutes_t, [873](#)
 - ParamPresenceMask, [873](#)
- unpack_sms_SLQSSwiGetSMSStorage
 - sms.h, [1406](#)
- unpack_sms_SLQSSwiGetSMSStorage_t, [874](#)
 - pSmsStorage, [874](#)
 - ParamPresenceMask, [874](#)
- unpack_sms_SLQSTransLayerInfoCallback_ind
 - sms.h, [1407](#)
- unpack_sms_SLQSTransLayerInfoCallback_ind_t, [874](#)
 - pTransLayerInfo, [875](#)
 - ParamPresenceMask, [875](#)
 - regInd, [875](#)
- unpack_sms_SLQSWmsAsyncRawSendCallBack_ind
 - sms.h, [1407](#)
- unpack_sms_SLQSWmsAsyncRawSendCallBack_ind-
 - _t, [875](#)
 - causeCode, [877](#)
 - errorClass, [877](#)
 - messageID, [877](#)
 - RPCause, [877](#)
 - sendStatus, [877](#)
 - TPCause, [877](#)
 - userData, [877](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind
 - sms.h, [1408](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind_t,
 - [877](#)
 - messageMode, [878](#)
 - ParamPresenceMask, [878](#)
 - storageType, [878](#)
- unpack_sms_SLQSWmsMessageWaitingCallBack_ind
 - sms.h, [1408](#)
- unpack_sms_SLQSWmsMessageWaitingCallBack_ind-
 - _t, [878](#)
 - msgWaitInfo, [878](#)
 - numInstances, [878](#)
 - ParamPresenceMask, [878](#)
- unpack_sms_SaveSMS
 - sms.h, [1398](#)
- unpack_sms_SaveSMS_t, [860](#)
 - pMessageIndex, [861](#)
 - ParamPresenceMask, [861](#)
- unpack_sms_SendSMS
 - sms.h, [1399](#)
- unpack_sms_SendSMS_t, [861](#)
 - messageFailureCode, [861](#)
 - messageID, [861](#)
 - ParamPresenceMask, [861](#)
- unpack_sms_SetNewSMSCallback
 - sms.h, [1399](#)
- unpack_sms_SetNewSMSCallback_ind
 - sms.h, [1399](#)
- unpack_sms_SetNewSMSCallback_ind_t, [862](#)
 - ETWSTIv, [863](#)
 - IMSTIv, [863](#)
 - MMTIv, [863](#)
 - NewMMTIv, [863](#)
 - ParamPresenceMask, [863](#)
 - SMSCTIv, [863](#)
 - TRMessageTIv, [863](#)
- unpack_sms_SetNewSMSCallback_t, [863](#)
 - ParamPresenceMask, [863](#)
- unpack_sms_SetSMSCAddress
 - sms.h, [1400](#)
- unpack_sms_SetSMSCAddress_t, [863](#)
 - ParamPresenceMask, [864](#)
- unpack_swiaudio_SLQSGetM2MAVMMute
 - swiaudio.h, [1415](#)
- unpack_swiaudio_SLQSGetM2MAVMMute_t, [880](#)
 - CwtMute, [881](#)
 - EarMute, [881](#)
 - MicMute, [881](#)
 - ParamPresenceMask, [881](#)
- unpack_swiaudio_SLQSGetM2MAudioProfile
 - swiaudio.h, [1414](#)
- unpack_swiaudio_SLQSGetM2MAudioProfile_t, [878](#)
 - CwtMute, [879](#)
 - EarMute, [879](#)
 - Generator, [879](#)

- MicMute, [879](#)
- ParamPresenceMask, [880](#)
- Profile, [880](#)
- Volume, [880](#)
- unpack_swiaudio_SLQSSetM2MAudioVolume
 - swiaudio.h, [1415](#)
- unpack_swiaudio_SLQSSetM2MAudioVolume_t, [880](#)
 - Level, [880](#)
 - ParamPresenceMask, [880](#)
- unpack_swiaudio_SLQSSetM2MSpkrGain
 - swiaudio.h, [1415](#)
- unpack_swiaudio_SLQSSetM2MSpkrGain_t, [881](#)
 - ParamPresenceMask, [882](#)
 - Value, [882](#)
- unpack_swiaudio_SLQSSetM2MAVMute
 - swiaudio.h, [1417](#)
- unpack_swiaudio_SLQSSetM2MAVMute_t
 - swiaudio.h, [1410](#)
- unpack_swiaudio_SLQSSetM2MAudioAVCFG
 - swiaudio.h, [1416](#)
- unpack_swiaudio_SLQSSetM2MAudioAVCFG_t
 - swiaudio.h, [1410](#)
- unpack_swiaudio_SLQSSetM2MAudioLPBK
 - swiaudio.h, [1416](#)
- unpack_swiaudio_SLQSSetM2MAudioLPBK_t
 - swiaudio.h, [1410](#)
- unpack_swiaudio_SLQSSetM2MAudioNVDef
 - swiaudio.h, [1416](#)
- unpack_swiaudio_SLQSSetM2MAudioNVDef_t
 - swiaudio.h, [1410](#)
- unpack_swiaudio_SLQSSetM2MAudioProfile
 - swiaudio.h, [1417](#)
- unpack_swiaudio_SLQSSetM2MAudioProfile_t
 - swiaudio.h, [1410](#)
- unpack_swiaudio_SLQSSetM2MAudioVolume
 - swiaudio.h, [1417](#)
- unpack_swiaudio_SLQSSetM2MAudioVolume_t
 - swiaudio.h, [1410](#)
- unpack_swiaudio_SLQSSetM2MSpkrGain
 - swiaudio.h, [1418](#)
- unpack_swiaudio_SLQSSetM2MSpkrGain_t
 - swiaudio.h, [1410](#)
- unpack_swiaavms_SLQSAVMSSEventReportInd
 - swiaavms.h, [1427](#)
- unpack_swiaavms_SLQSAVMSSEventReportInd_t, [882](#)
 - pConfig, [883](#)
 - pConnectionRequest, [883](#)
 - pDataSessionStatus, [883](#)
 - pNotification, [883](#)
 - pPackageID, [883](#)
 - pRegStatus, [883](#)
 - pSessionType, [883](#)
 - ParamPresenceMask, [883](#)
 - resultcode, [883](#)
- unpack_swiaavms_SLQSAVMSGetSettings
 - swiaavms.h, [1427](#)
- unpack_swiaavms_SLQSAVMSGetSettings_t, [883](#)
 - AutoReboot, [885](#)
 - AutoReboot, [885](#)
 - FwAutoSDM, [885](#)
 - FwPromptUpdate, [886](#)
 - FwPromptdownload, [885](#)
 - pAPNInfo, [886](#)
 - pConnectionRetryTimers, [886](#)
 - pNotificationStore, [886](#)
 - pPeroidsInfo, [886](#)
 - pPollingTimer, [886](#)
 - ParamPresenceMask, [886](#)
 - resultcode, [886](#)
- unpack_swiaavms_SLQSAVMSGetSettings_v2
 - swiaavms.h, [1427](#)
- unpack_swiaavms_SLQSAVMSGetSettings_v2_t, [886](#)
 - AutoConnect, [888](#)
 - FwPromptUpdate, [888](#)
 - FwPromptdownload, [888](#)
 - pAutoReboot, [888](#)
 - pNotificationStore, [888](#)
 - pPeroidsInfo, [888](#)
 - pPollingTimer, [888](#)
 - ParamPresenceMask, [888](#)
 - resultcode, [888](#)
- unpack_swiaavms_SLQSAVMSSEndSelection
 - swiaavms.h, [1428](#)
- unpack_swiaavms_SLQSAVMSSEndSelection_t, [888](#)
 - ParamPresenceMask, [889](#)
 - resultcode, [889](#)
- unpack_swiaavms_SLQSAVMSSessionGetInfo
 - swiaavms.h, [1428](#)
- unpack_swiaavms_SLQSAVMSSessionGetInfo_t, [889](#)
 - pConfig, [889](#)
 - pNotification, [890](#)
 - pPackageID, [890](#)
 - ParamPresenceMask, [889](#)
 - resultcode, [890](#)
- unpack_swiaavms_SLQSAVMSSetSettings
 - swiaavms.h, [1429](#)
- unpack_swiaavms_SLQSAVMSSetSettings_t, [890](#)
 - ParamPresenceMask, [890](#)
 - resultcode, [890](#)
- unpack_swiaavms_SLQSAVMSSetSettings_v2
 - swiaavms.h, [1430](#)
- unpack_swiaavms_SLQSAVMSSetSettings_v2_t, [891](#)
 - ParamPresenceMask, [891](#)
 - resultcode, [891](#)
- unpack_swiaavms_SLQSAVMSStartSession
 - swiaavms.h, [1430](#)
- unpack_swiaavms_SLQSAVMSStartSession_t, [891](#)
 - ParamPresenceMask, [891](#)
 - resultcode, [891](#)
 - sessionResponse, [892](#)
- unpack_swiaavms_SLQSAVMSStopSession
 - swiaavms.h, [1431](#)
- unpack_swiaavms_SLQSAVMSStopSession_avc2
 - swiaavms.h, [1431](#)
- unpack_swiaavms_SLQSAVMSStopSession_avc2_t
 - swiaavms.h, [1420](#)

- unpack_swiavms_SLQSAVMSStopSession_t, 892
 - ParamPresenceMask, 892
 - resultcode, 892
- unpack_swiavms_SLQSAvmsSetEventReport
 - swiavms.h, 1429
- unpack_swiavms_SLQSAvmsSetEventReport_t, 890
 - ParamPresenceMask, 890
 - resultcode, 890
- unpack_swidms_SLQSSwiDmsGetHWWatchdog
 - swidms.h, 1438
- unpack_swidms_SLQSSwiDmsGetHWWatchdog_t, 892
 - pHWWatchdog, 892
 - ParamPresenceMask, 892
- unpack_swidms_SLQSSwiDmsGetMTU
 - swidms.h, 1438
- unpack_swidms_SLQSSwiDmsGetMTU_t, 893
 - pMTUSize3gpp, 893
 - ParamPresenceMask, 893
- unpack_swidms_SLQSSwiDmsGetSecureInfo
 - swidms.h, 1438
- unpack_swidms_SLQSSwiDmsGetSecureInfo_t, 893
 - jtagAccessAllowed, 894
 - memoryDumpAllowed, 894
 - ParamPresenceMask, 894
 - secureBootEnabled, 894
 - TlvResult, 894
- unpack_swidms_SLQSSwiDmsGetUsbComp
 - swidms.h, 1439
- unpack_swidms_SLQSSwiDmsGetUsbComp_t, 894
 - pInterfaceCfg, 895
 - pSupportedBitmasks, 895
 - ParamPresenceMask, 895
- unpack_swidms_SLQSSwiDmsGetUsbNetNum
 - swidms.h, 1439
- unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 895
 - ParamPresenceMask, 895
 - usbNetNum, 895
- unpack_swidms_SLQSSwiDmsSetHWWatchdog
 - swidms.h, 1440
- unpack_swidms_SLQSSwiDmsSetHWWatchdog_t, 896
 - ParamPresenceMask, 896
 - Tlvresult, 896
- unpack_swidms_SLQSSwiDmsSetMTU
 - swidms.h, 1440
- unpack_swidms_SLQSSwiDmsSetMTU_t, 896
 - ParamPresenceMask, 896
 - Tlvresult, 896
- unpack_swidms_SLQSSwiDmsSetUsbComp
 - swidms.h, 1440
- unpack_swidms_SLQSSwiDmsSetUsbComp_t, 897
 - ParamPresenceMask, 897
 - Tlvresult, 897
- unpack_swidms_SLQSSwiDmsSetUsbNetNum
 - swidms.h, 1441
- unpack_swidms_SLQSSwiDmsSetUsbNetNum_t
 - swidms.h, 1435
- unpack_swiloc_SwiLocGetAutoStart
 - swiloc.h, 1442
- unpack_swiloc_SwiLocGetAutoStart_t, 897
 - fix_rate, 899
 - fix_rate_reported, 899
 - fix_type, 899
 - fix_type_reported, 899
 - function, 899
 - function_reported, 899
 - max_dist, 899
 - max_dist_reported, 899
 - max_time, 899
 - max_time_reported, 899
 - ParamPresenceMask, 899
- unpack_swiloc_SwiLocSetAutoStart
 - swiloc.h, 1443
- unpack_swiloc_SwiLocSetAutoStart_t
 - swiloc.h, 1441
- unpack_swioma_SLQSOMADMAAlertCallback
 - swioma.h, 1449
- unpack_swioma_SLQSOMADMAAlertCallback_ind
 - swioma.h, 1449
- unpack_swioma_SLQSOMADMAAlertCallback_ind_t, 899
 - eventType, 900
- unpack_swioma_SLQSOMADMAAlertCallback_t
 - swioma.h, 1444
- unpack_swioma_SLQSOMADMCancelSession
 - swioma.h, 1450
- unpack_swioma_SLQSOMADMCancelSession_t
 - swioma.h, 1444
- unpack_swioma_SLQSOMADMCancelSessionExt
 - swiomaext.h, 1457
- unpack_swioma_SLQSOMADMCancelSessionExt_t
 - swiomaext.h, 1453
- unpack_swioma_SLQSOMADMGetSessionInfo
 - swioma.h, 1450
- unpack_swioma_SLQSOMADMGetSessionInfo_t, 900
 - Date, 902
 - DateLength, 902
 - ParamPresenceMask, 903
 - PkgDescLength, 903
 - PkgDescription, 903
 - PkgName, 903
 - PkgNameLength, 903
 - RetryCount, 903
 - SessionState, 903
 - SessionType, 903
 - Severity, 903
 - Source, 903
 - SourceLength, 903
 - Status, 903
 - Time, 903
 - TimeLength, 903
 - UpdateCompleteStatus, 903
- unpack_swioma_SLQSOMADMGetSessionInfoExt
 - swiomaext.h, 1457
- unpack_swioma_SLQSOMADMGetSessionInfoExt_t, 903
 - fumoState, 906

- hfaStatus, 906
- pkgDate, 906
- pkgDesc, 906
- pkgName, 906
- pkgSize, 906
- sessionState, 906
- status, 906
- unpack_swisma_SLQSOMADMGetSettings
 - swisma.h, 1451
- unpack_swisma_SLQSOMADMGetSettings_t, 907
 - Autosdm, 908
 - FOTAdownload, 908
 - FwAutoCheck, 908
 - ParamPresenceMask, 908
- unpack_swisma_SLQSOMADMSendSelection
 - swisma.h, 1451
- unpack_swisma_SLQSOMADMSendSelection_t
 - swisma.h, 1444
- unpack_swisma_SLQSOMADMSendSelectionExt
 - swismaext.h, 1458
- unpack_swisma_SLQSOMADMSendSelectionExt_t
 - swismaext.h, 1454
- unpack_swisma_SLQSOMADMSetSettings
 - swisma.h, 1452
- unpack_swisma_SLQSOMADMSetSettings_t
 - swisma.h, 1444
- unpack_swisma_SLQSOMADMSetSettingsExt
 - swismaext.h, 1458
- unpack_swisma_SLQSOMADMSetSettingsExt_t
 - swismaext.h, 1454
- unpack_swisma_SLQSOMADMStartSession
 - swisma.h, 1452
- unpack_swisma_SLQSOMADMStartSession_t, 908
 - FwAvailability, 909
 - ParamPresenceMask, 909
- unpack_swisma_SLQSOMADMStartSessionExt
 - swismaext.h, 1459
- unpack_swisma_SLQSOMADMStartSessionExt_t
 - swismaext.h, 1454
- unpack_tmd_SLQSTmdDeRegNotMitigationLvl
 - tmd.h, 1463
- unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 909
 - ParamPresenceMask, 909
 - Tlvresult, 909
- unpack_tmd_SLQSTmdGetMitigationDevList
 - tmd.h, 1463
- unpack_tmd_SLQSTmdGetMitigationDevList_t, 909
 - MitigationDevList, 910
 - MitigationDevListLen, 910
 - ParamPresenceMask, 910
 - Tlvresult, 910
- unpack_tmd_SLQSTmdGetMitigationLvl
 - tmd.h, 1464
- unpack_tmd_SLQSTmdGetMitigationLvl_t, 910
 - CurrentmitigationLvl, 911
 - ParamPresenceMask, 911
 - ReqMitigationLvl, 911
 - Tlvresult, 911
- unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind
 - tmd.h, 1464
- unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, 911
 - deviceId, 911
 - deviceIdLen, 911
 - lvl, 911
 - ParamPresenceMask, 911
- unpack_tmd_SLQSTmdRegNotMitigationLvl
 - tmd.h, 1464
- unpack_tmd_SLQSTmdRegNotMitigationLvl_t, 911
 - ParamPresenceMask, 912
 - Tlvresult, 912
- unpack_uim_ChangePin
 - uim.h, 1476
- unpack_uim_ChangePin_t, 912
 - pEncryptedPIN1, 913
 - pIndicationToken, 913
 - pRemainingRetries, 913
 - ParamPresenceMask, 913
 - Tlvresult, 913
- unpack_uim_GetCardStatus
 - uim.h, 1477
- unpack_uim_GetCardStatus_t, 913
 - pCardStatus, 913
 - pHotSwapStatus, 913
 - ParamPresenceMask, 913
 - Tlvresult, 913
- unpack_uim_GetCardStatusV2
 - uim.h, 1477
- unpack_uim_GetCardStatusV2_t, 914
 - pCardStatus, 914
 - pHotSwapStatus, 914
 - pSimBusyStatus, 914
 - pValidCardStatus, 914
 - ParamPresenceMask, 914
 - Tlvresult, 914
- unpack_uim_ReadTransparent
 - uim.h, 1477
- unpack_uim_ReadTransparent_t, 915
 - pCardResult, 915
 - pEncryptedData, 915
 - pIndicationToken, 915
 - pReadResult, 915
 - ParamPresenceMask, 915
 - Tlvresult, 915
- unpack_uim_SLQSUIMAuthenticate
 - uim.h, 1478
- unpack_uim_SLQSUIMAuthenticate_t, 917
 - pAuthenticateResult, 918
 - pCardResult, 918
 - pIndicationToken, 918
 - ParamPresenceMask, 918
- unpack_uim_SLQSUIMDepersonalization
 - uim.h, 1479
- unpack_uim_SLQSUIMDepersonalization_t, 918
 - pRemainingRetries, 918
 - ParamPresenceMask, 918

- unpack_uim_SLQSUIEventRegister
 - uim.h, [1479](#)
- unpack_uim_SLQSUIEventRegister_t, [918](#)
 - eventMask, [919](#)
 - ParamPresenceMask, [919](#)
- unpack_uim_SLQSUIGetConfiguration
 - uim.h, [1479](#)
- unpack_uim_SLQSUIGetConfiguration_t, [919](#)
 - pAutoSelection, [920](#)
 - pHaltSubscription, [920](#)
 - pPersonalizationStatus, [920](#)
 - ParamPresenceMask, [920](#)
- unpack_uim_SLQSUIGetFileAttributes
 - uim.h, [1480](#)
- unpack_uim_SLQSUIGetFileAttributes_t, [920](#)
 - pCardResult, [921](#)
 - pFileAttributes, [921](#)
 - pIndicationToken, [921](#)
 - ParamPresenceMask, [921](#)
- unpack_uim_SLQSUIGetSlotsStatus
 - uim.h, [1480](#)
- unpack_uim_SLQSUIGetSlotsStatus_t, [921](#)
 - pNumberOfPhySlot, [921](#)
 - pUimSlotsStatus, [921](#)
 - ParamPresenceMask, [921](#)
- unpack_uim_SLQSUIGetSlotsStatusV2
 - uim.h, [1481](#)
- unpack_uim_SLQSUIGetSlotsStatusV2_t, [921](#)
 - ParamPresenceMask, [922](#)
- unpack_uim_SLQSUIPowerDown
 - uim.h, [1481](#)
- unpack_uim_SLQSUIPowerDown_t
 - uim.h, [1469](#)
- unpack_uim_SLQSUIPowerUp
 - uim.h, [1481](#)
- unpack_uim_SLQSUIPowerUp_t
 - uim.h, [1469](#)
- unpack_uim_SLQSUIRefreshCallback_Ind
 - uim.h, [1482](#)
- unpack_uim_SLQSUIRefreshCallback_Ind_t, [922](#)
 - ParamPresenceMask, [923](#)
 - refreshEvent, [923](#)
 - TlvPresent, [923](#)
- unpack_uim_SLQSUIRefreshComplete
 - uim.h, [1482](#)
- unpack_uim_SLQSUIRefreshComplete_t
 - uim.h, [1469](#)
- unpack_uim_SLQSUIRefreshGetLastEvent
 - uim.h, [1482](#)
- unpack_uim_SLQSUIRefreshGetLastEvent_t, [923](#)
 - pRefreshEvent, [923](#)
 - ParamPresenceMask, [923](#)
- unpack_uim_SLQSUIRefreshOK
 - uim.h, [1483](#)
- unpack_uim_SLQSUIRefreshOK_t
 - uim.h, [1469](#)
- unpack_uim_SLQSUIRefreshRegister
 - uim.h, [1483](#)
- unpack_uim_SLQSUIRefreshRegister_t
 - uim.h, [1469](#)
- unpack_uim_SLQSUIReset
 - uim.h, [1483](#)
- unpack_uim_SLQSUIReset_t
 - uim.h, [1469](#)
- unpack_uim_SLQSUISetStatusChangeCallBack_ind
 - uim.h, [1484](#)
- unpack_uim_SLQSUISetStatusChangeCallBack_ind_t, [923](#)
 - uim.h, [1484](#)
- unpack_uim_SLQSUISwitchSlot
 - uim.h, [1484](#)
- unpack_uim_SLQSUISwitchSlot_t
 - uim.h, [1469](#)
- unpack_uim_SetPinProtection
 - uim.h, [1478](#)
- unpack_uim_SetPinProtection_t, [916](#)
 - pEncryptedPIN1, [916](#)
 - pIndicationToken, [916](#)
 - pRemainingRetries, [916](#)
 - ParamPresenceMask, [916](#)
 - Tlvresult, [916](#)
- unpack_uim_SetUimSlotStatusChangeCallback_ind
 - uim.h, [1478](#)
- unpack_uim_SetUimSlotStatusChangeCallback_ind_t, [916](#)
 - bNumberOfPhySlots, [917](#)
 - ParamPresenceMask, [917](#)
 - slotsstatusChange, [917](#)
- unpack_uim_UnblockPin
 - uim.h, [1484](#)
- unpack_uim_UnblockPin_t, [924](#)
 - pEncryptedPIN1, [924](#)
 - pIndicationToken, [925](#)
 - pRemainingRetries, [925](#)
 - ParamPresenceMask, [924](#)
 - Tlvresult, [925](#)
- unpack_uim_UnblockPinV2
 - uim.h, [1485](#)
- unpack_uim_UnblockPinV2_t, [925](#)
 - pCardResult, [925](#)
 - pEncryptedPIN1, [925](#)
 - pIndicationToken, [926](#)
 - pRemainingRetries, [926](#)
 - ParamPresenceMask, [925](#)
 - Tlvresult, [926](#)
- unpack_uim_VerifyPin
 - uim.h, [1485](#)
- unpack_uim_VerifyPin_t, [926](#)
 - pEncryptedPIN1, [926](#)
 - pIndicationToken, [926](#)
 - pRemainingRetries, [926](#)
 - ParamPresenceMask, [926](#)
 - Tlvresult, [926](#)
- unpack_valid_nas_GetCDMANetworkParameters
 - nas.h, [1300](#)
- unpack_valid_nas_SLQSGetServingSystem
 - nas.h, [1301](#)

- unpack_valid_nas_SLQSGetSignalStrength
 - nas.h, [1302](#)
- unpack_valid_nas_SLQSNasGetSigInfo
 - nas.h, [1302](#)
- unpack_voice_AnswerUSSD
 - voice.h, [1504](#)
- unpack_voice_AnswerUSSD_t
 - voice.h, [1491](#)
- unpack_voice_CancelUSSD
 - voice.h, [1504](#)
- unpack_voice_CancelUSSD_t
 - voice.h, [1491](#)
- unpack_voice_DTMFEventCallback_ind
 - voice.h, [1504](#)
- unpack_voice_DTMFEventCallback_ind_t, [929](#)
 - DTMFInformation, [929](#)
 - pOffLength, [929](#)
 - pOnLength, [929](#)
 - ParamPresenceMask, [929](#)
- unpack_voice_OTASPStatusCallback_ind
 - voice.h, [1505](#)
- unpack_voice_OTASPStatusCallback_ind_t, [930](#)
 - callID, [931](#)
 - OTASPStatus, [931](#)
 - ParamPresenceMask, [931](#)
- unpack_voice_OriginateUSSD
 - voice.h, [1505](#)
- unpack_voice_OriginateUSSD_t
 - voice.h, [1491](#)
- unpack_voice_SLQSOriginateUSSD
 - voice.h, [1505](#)
- unpack_voice_SLQSOriginateUSSD_t, [931](#)
 - pAlphaIDInfo, [933](#)
 - pCCSuppsType, [933](#)
 - pCallID, [933](#)
 - pCCResultType, [933](#)
 - pUSSDInfo, [933](#)
 - ParamPresenceMask, [933](#)
 - pfailureCause, [933](#)
- unpack_voice_SLQSVoiceALSSelectLine
 - voice.h, [1506](#)
- unpack_voice_SLQSVoiceALSSelectLine_t
 - voice.h, [1491](#)
- unpack_voice_SLQSVoiceALSSetLineSwitching
 - voice.h, [1506](#)
- unpack_voice_SLQSVoiceALSSetLineSwitching_t
 - voice.h, [1491](#)
- unpack_voice_SLQSVoiceAnswerCall
 - voice.h, [1507](#)
- unpack_voice_SLQSVoiceAnswerCall_t, [933](#)
 - pCallID, [934](#)
 - ParamPresenceMask, [934](#)
- unpack_voice_SLQSVoiceBindSubscription
 - voice.h, [1507](#)
- unpack_voice_SLQSVoiceBindSubscription_t
 - voice.h, [1491](#)
- unpack_voice_SLQSVoiceBurstDTMF
 - voice.h, [1507](#)
- unpack_voice_SLQSVoiceBurstDTMF_t, [934](#)
 - pCallID, [934](#)
 - ParamPresenceMask, [934](#)
- unpack_voice_SLQSVoiceDialCall
 - voice.h, [1508](#)
- unpack_voice_SLQSVoiceDialCall_t, [934](#)
 - pAlphaIDInfo, [935](#)
 - pCCResultType, [935](#)
 - pCCSUPSType, [935](#)
 - pCallID, [935](#)
 - ParamPresenceMask, [935](#)
- unpack_voice_SLQSVoiceEndCall
 - voice.h, [1508](#)
- unpack_voice_SLQSVoiceEndCall_t, [935](#)
 - pCallID, [936](#)
 - ParamPresenceMask, [936](#)
- unpack_voice_SLQSVoiceGetAllCallInfo
 - voice.h, [1508](#)
- unpack_voice_SLQSVoiceGetAllCallInfo_t, [936](#)
 - pArrAlertingPattern, [939](#)
 - pArrAlertingType, [939](#)
 - pArrAlphaID, [939](#)
 - pArrCallEndReason, [939](#)
 - pArrCallInfo, [939](#)
 - pArrCalledPartyNum, [939](#)
 - pArrConnectPartyNum, [939](#)
 - pArrDiagInfo, [939](#)
 - pArrRedirPartyNum, [939](#)
 - pArrRemotePartyName, [939](#)
 - pArrRemotePartyNum, [939](#)
 - pArrSvcOption, [939](#)
 - pArrUUSInfo, [939](#)
 - pVoicePrivacy, [939](#)
 - ParamPresenceMask, [939](#)
- unpack_voice_SLQSVoiceGetCLIP
 - voice.h, [1510](#)
- unpack_voice_SLQSVoiceGetCLIP_t, [947](#)
 - pAlphaIDInfo, [948](#)
 - pCCResType, [948](#)
 - pCallID, [948](#)
 - pFailCause, [949](#)
 - ParamPresenceMask, [948](#)
- unpack_voice_SLQSVoiceGetCLIR
 - voice.h, [1510](#)
- unpack_voice_SLQSVoiceGetCLIR_t, [949](#)
 - pAlphaIDInfo, [950](#)
 - pCCResType, [950](#)
 - pCallID, [950](#)
 - pFailCause, [950](#)
 - ParamPresenceMask, [950](#)
- unpack_voice_SLQSVoiceGetCNAP
 - voice.h, [1511](#)
- unpack_voice_SLQSVoiceGetCNAP_t, [950](#)
 - pAlphaIDInfo, [951](#)
 - pCCResType, [951](#)
 - pCallID, [951](#)
 - pFailCause, [951](#)
 - ParamPresenceMask, [951](#)

- unpack_voice_SLQSVoiceGetCOLP
 - voice.h, [1511](#)
- unpack_voice_SLQSVoiceGetCOLP_t, [952](#)
 - pAlphaIDInfo, [953](#)
 - pCCResType, [953](#)
 - pCallID, [953](#)
 - pFailCause, [953](#)
 - ParamPresenceMask, [953](#)
- unpack_voice_SLQSVoiceGetCOLR
 - voice.h, [1511](#)
- unpack_voice_SLQSVoiceGetCOLR_t, [953](#)
 - pAlphaIDInfo, [954](#)
 - pCCResType, [954](#)
 - pCallID, [954](#)
 - pFailCause, [954](#)
 - ParamPresenceMask, [954](#)
- unpack_voice_SLQSVoiceGetCallBarring
 - voice.h, [1509](#)
- unpack_voice_SLQSVoiceGetCallBarring_t, [940](#)
 - pAlphaIDInfo, [941](#)
 - pCCResType, [941](#)
 - pCallID, [941](#)
 - pFailCause, [941](#)
 - pSvcClass, [941](#)
 - ParamPresenceMask, [941](#)
- unpack_voice_SLQSVoiceGetCallForwardingStatus
 - voice.h, [1509](#)
- unpack_voice_SLQSVoiceGetCallForwardingStatus_t, [941](#)
 - pAlphaIDInfo, [942](#)
 - pCCResType, [942](#)
 - pCallID, [942](#)
 - pFailCause, [942](#)
 - ParamPresenceMask, [942](#)
- unpack_voice_SLQSVoiceGetCallInfo
 - voice.h, [1509](#)
- unpack_voice_SLQSVoiceGetCallInfo_t, [943](#)
 - pAlertType, [945](#)
 - pAlertingPattern, [945](#)
 - pAlphaIDInfo, [945](#)
 - pCallInfo, [945](#)
 - pConnectNumInfo, [945](#)
 - pDiagInfo, [945](#)
 - pOTASPStatus, [945](#)
 - pRemotePartyName, [945](#)
 - pRemotePartyNum, [945](#)
 - pSrvOpt, [945](#)
 - pUUSInfo, [946](#)
 - pVoicePrivacy, [946](#)
 - ParamPresenceMask, [945](#)
- unpack_voice_SLQSVoiceGetCallWaiting
 - voice.h, [1510](#)
- unpack_voice_SLQSVoiceGetCallWaiting_t, [946](#)
 - pAlphaIDInfo, [947](#)
 - pCCResType, [947](#)
 - pCallID, [947](#)
 - pFailCause, [947](#)
 - pSvcClass, [947](#)
 - ParamPresenceMask, [947](#)
- unpack_voice_SLQSVoiceGetConfig
 - voice.h, [1512](#)
- unpack_voice_SLQSVoiceGetConfig_t, [954](#)
 - pAirTimerCnt, [956](#)
 - pAutoAnswerStat, [956](#)
 - pCurAMRConfig, [956](#)
 - pCurPrefVoiceSO, [956](#)
 - pCurVoiceDomainPref, [956](#)
 - pCurVoicePrivacyPref, [956](#)
 - pCurrTTYMode, [956](#)
 - pRoamTimerCnt, [956](#)
 - ParamPresenceMask, [956](#)
- unpack_voice_SLQSVoiceIndicationRegister
 - voice.h, [1512](#)
- unpack_voice_SLQSVoiceIndicationRegister_t
 - voice.h, [1491](#)
- unpack_voice_SLQSVoiceManageCalls
 - voice.h, [1512](#)
- unpack_voice_SLQSVoiceManageCalls_t, [956](#)
 - pFailCause, [957](#)
 - ParamPresenceMask, [957](#)
- unpack_voice_SLQSVoiceOrigUSSDNoWait
 - voice.h, [1513](#)
- unpack_voice_SLQSVoiceOrigUSSDNoWait_t
 - voice.h, [1491](#)
- unpack_voice_SLQSVoiceSUPSCallback_ind
 - voice.h, [1515](#)
- unpack_voice_SLQSVoiceSUPSCallback_ind_t, [963](#)
 - pDataSrc, [966](#)
 - pFailCause, [966](#)
 - pReason, [966](#)
 - pSvcClass, [966](#)
 - ParamPresenceMask, [965](#)
- unpack_voice_SLQSVoiceSendFlash
 - voice.h, [1513](#)
- unpack_voice_SLQSVoiceSendFlash_t, [957](#)
 - pCallID, [957](#)
 - ParamPresenceMask, [957](#)
- unpack_voice_SLQSVoiceSetCallBarringPassword
 - voice.h, [1513](#)
- unpack_voice_SLQSVoiceSetCallBarringPassword_t, [958](#)
 - pAlphaIDInfo, [959](#)
 - pCCResType, [959](#)
 - pCallID, [959](#)
 - pFailCause, [959](#)
 - ParamPresenceMask, [959](#)
- unpack_voice_SLQSVoiceSetConfig
 - voice.h, [1514](#)
- unpack_voice_SLQSVoiceSetConfig_t, [959](#)
 - pAirTimerStatus, [960](#)
 - pAutoAnsStatus, [960](#)
 - pPrefVoiceSOStatus, [960](#)
 - pRoamTimerStatus, [960](#)
 - pTTYConfigStatus, [960](#)
 - pVoiceDomainPrefStatus, [960](#)
 - ParamPresenceMask, [960](#)

- unpack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, [1514](#)
- unpack_voice_SLQSVoiceSetPreferredPrivacy_t
voice.h, [1491](#)
- unpack_voice_SLQSVoiceSetSUPSService
voice.h, [1514](#)
- unpack_voice_SLQSVoiceSetSUPSService_t, [961](#)
pCallID, [962](#)
pFailCause, [962](#)
ParamPresenceMask, [962](#)
- unpack_voice_SLQSVoiceStartContDTMF
voice.h, [1515](#)
- unpack_voice_SLQSVoiceStartContDTMF_t, [962](#)
pCallID, [962](#)
ParamPresenceMask, [962](#)
- unpack_voice_SLQSVoiceStopContDTMF
voice.h, [1515](#)
- unpack_voice_SLQSVoiceStopContDTMF_t, [962](#)
callID, [963](#)
ParamPresenceMask, [963](#)
- unpack_voice_SUPSNotificationCallback_ind
voice.h, [1516](#)
- unpack_voice_SUPSNotificationCallback_ind_t, [966](#)
callID, [968](#)
notifType, [968](#)
pCUGIndex, [968](#)
pECTNum, [968](#)
ParamPresenceMask, [968](#)
- unpack_voice_USSDNotificationCallback_ind
voice.h, [1516](#)
- unpack_voice_USSDNotificationCallback_ind_t, [968](#)
notification_Type, [968](#)
ParamPresenceMask, [968](#)
- unpack_voice_VoiceInfoRecCallback_ind
voice.h, [1516](#)
- unpack_voice_VoiceInfoRecCallback_ind_t, [968](#)
callID, [971](#)
pCLIRCause, [971](#)
pCallWaitInd, [971](#)
pCalledPartyInfo, [971](#)
pCallerIDInfo, [971](#)
pCallerNameInfo, [971](#)
pCallingPartyInfo, [971](#)
pConnectNumInfo, [971](#)
pDisplInfo, [971](#)
pExtDisplInfo, [971](#)
pExtDisplRecInfo, [971](#)
pLineCtrlInfo, [971](#)
pNSSAudioCtrl, [971](#)
pNSSRelease, [971](#)
pRedirNumInfo, [971](#)
pSignalInfo, [971](#)
ParamPresenceMask, [971](#)
- unpack_voice_allCallStatusCallback_ind
voice.h, [1503](#)
- unpack_voice_allCallStatusCallback_ind_t, [927](#)
arrCallInformation, [928](#)
pArrAlertingPattern, [928](#)
pArrAlertingType, [928](#)
pArrAlphaID, [928](#)
pArrCallEndReason, [928](#)
pArrCalledPartyNum, [928](#)
pArrConnectPartyNum, [928](#)
pArrDiagInfo, [928](#)
pArrRedirPartyNum, [928](#)
pArrRemotePartyName, [928](#)
pArrRemotePartyNum, [928](#)
pArrSvcOption, [928](#)
ParamPresenceMask, [928](#)
- unpack_voice_voicePrivacyChangeCallback_ind
voice.h, [1517](#)
- unpack_voice_voicePrivacyChangeCallback_ind_t, [971](#)
callID, [972](#)
ParamPresenceMask, [972](#)
voicePrivacy, [972](#)
- unpack_wds_DHCPv4ClientLease_ind
wds.h, [1548](#)
- unpack_wds_DHCPv4ClientLease_ind_t, [972](#)
DHCPv4LeaseOptTlv, [973](#)
DHCPv4LeaseStateTlv, [973](#)
IPv4AddrTlv, [973](#)
ParamPresenceMask, [973](#)
ProfileIdTlv, [973](#)
- unpack_wds_DHCPv4ClientLeaseChange
wds.h, [1548](#)
- unpack_wds_DHCPv4ClientLeaseChange_t
wds.h, [1526](#)
- unpack_wds_GetAutoconnect
wds.h, [1549](#)
- unpack_wds_GetAutoconnect_t, [973](#)
ParamPresenceMask, [973](#)
psetting, [973](#)
- unpack_wds_GetByteTotals
wds.h, [1549](#)
- unpack_wds_GetByteTotals_t, [973](#)
pRXTotalBytes, [974](#)
pTXTotalBytes, [974](#)
ParamPresenceMask, [974](#)
- unpack_wds_GetConnectionRate
wds.h, [1549](#)
- unpack_wds_GetConnectionRate_t, [974](#)
currentChannelRXRate, [975](#)
currentChannelTXRate, [975](#)
maxChannelRXRate, [975](#)
maxChannelTXRate, [975](#)
ParamPresenceMask, [975](#)
- unpack_wds_GetDataBearerTechnology
wds.h, [1550](#)
- unpack_wds_GetDataBearerTechnology_t, [975](#)
pDataBearer, [976](#)
ParamPresenceMask, [976](#)
- unpack_wds_GetDefaultProfile
wds.h, [1550](#)
- unpack_wds_GetDefaultProfile_t, [976](#)
apnname, [978](#)
apnsize, [978](#)

- auth, [978](#)
- ipaddr, [978](#)
- ipaddrv6, [979](#)
- name, [979](#)
- namesize, [979](#)
- ParamPresenceMask, [979](#)
- pdptype, [979](#)
- pridns, [979](#)
- pridnsv6, [979](#)
- secdns, [979](#)
- secdnsv6, [979](#)
- username, [979](#)
- usersize, [979](#)
- unpack_wds_GetDefaultProfileNum
 - wds.h, [1550](#)
- unpack_wds_GetDefaultProfileNum_t, [979](#)
 - index, [979](#)
 - ParamPresenceMask, [979](#)
- unpack_wds_GetDefaultProfileV2
 - wds.h, [1551](#)
- unpack_wds_GetDefaultProfileV2_t, [979](#)
 - apnname, [982](#)
 - apnsize, [982](#)
 - auth, [982](#)
 - ipaddr, [982](#)
 - ipaddrv6, [982](#)
 - name, [982](#)
 - namesize, [982](#)
 - ParamPresenceMask, [982](#)
 - pdptype, [982](#)
 - pridns, [982](#)
 - pridnsv6, [982](#)
 - pwd, [982](#)
 - pwdsize, [982](#)
 - secdns, [982](#)
 - secdnsv6, [982](#)
 - username, [982](#)
 - usersize, [982](#)
- unpack_wds_GetDormancyState
 - wds.h, [1551](#)
- unpack_wds_GetDormancyState_t, [982](#)
 - dormancyState, [983](#)
 - ParamPresenceMask, [983](#)
- unpack_wds_GetLastMobileIPError
 - wds.h, [1552](#)
- unpack_wds_GetLastMobileIPError_t, [983](#)
 - error, [983](#)
 - ParamPresenceMask, [983](#)
- unpack_wds_GetMobileIP
 - wds.h, [1552](#)
- unpack_wds_GetMobileIP_t, [984](#)
 - mipMode, [984](#)
 - ParamPresenceMask, [984](#)
- unpack_wds_GetMobileIPProfile
 - wds.h, [1552](#)
- unpack_wds_GetMobileIPProfile_t, [984](#)
 - AAASPI, [986](#)
 - AAASState, [986](#)
 - address, [986](#)
 - enabled, [986](#)
 - HASPI, [986](#)
 - HASState, [986](#)
 - NAI, [986](#)
 - naiSize, [986](#)
 - ParamPresenceMask, [986](#)
 - primaryHA, [986](#)
 - revTunneling, [986](#)
 - secondaryHA, [986](#)
- unpack_wds_GetPacketStatistics
 - wds.h, [1553](#)
- unpack_wds_GetPacketStatistics_t, [986](#)
 - pRXDroppedCount, [988](#)
 - pRXOKBytesLastCall, [988](#)
 - pRXOkBytesCount, [988](#)
 - pRXPacketErrors, [988](#)
 - pRXPacketOverflows, [988](#)
 - pRXPacketSuccesses, [988](#)
 - pTXDroppedCount, [988](#)
 - pTXOKBytesLastCall, [988](#)
 - pTXOkBytesCount, [988](#)
 - pTXPacketErrors, [988](#)
 - pTXPacketOverflows, [988](#)
 - pTXPacketSuccesses, [988](#)
 - ParamPresenceMask, [988](#)
- unpack_wds_GetPacketStatus
 - wds.h, [1553](#)
- unpack_wds_GetPacketStatus_t, [988](#)
 - ParamPresenceMask, [990](#)
 - rXDroppedCount, [990](#)
 - rXOKBytesLastCall, [990](#)
 - rXOkBytesCount, [990](#)
 - rXPacketErrors, [990](#)
 - rXPacketOverflows, [990](#)
 - rXPacketSuccesses, [990](#)
 - tXDroppedCount, [990](#)
 - tXOKBytesLastCall, [990](#)
 - tXOkBytesCount, [990](#)
 - tXPacketErrors, [990](#)
 - tXPacketOverflows, [990](#)
 - tXPacketSuccesses, [990](#)
- unpack_wds_GetSessionDuration
 - wds.h, [1553](#)
- unpack_wds_GetSessionDuration_t, [990](#)
 - callDuration, [991](#)
 - ParamPresenceMask, [991](#)
- unpack_wds_GetSessionDurationV2
 - wds.h, [1554](#)
- unpack_wds_GetSessionDurationV2_t, [991](#)
 - callDuration, [992](#)
 - pCallActiveDuration, [992](#)
 - pLastCallActiveDuration, [992](#)
 - pLastCallDuration, [992](#)
 - ParamPresenceMask, [992](#)
- unpack_wds_GetSessionState
 - wds.h, [1554](#)
- unpack_wds_GetSessionState_t, [992](#)

- connectionStatus, 992
- ParamPresenceMask, 992
- unpack_wds_RMSetTransferStatistics
 - wds.h, 1554
- unpack_wds_RMSetTransferStatistics_t, 992
 - ParamPresenceMask, 993
- unpack_wds_RMTransferStatistics_ind
 - wds.h, 1555
- unpack_wds_RMTransferStatistics_ind_t
 - wds.h, 1526
- unpack_wds_SLQSCreateProfile
 - wds.h, 1557
- unpack_wds_SLQSCreateProfile_t, 993
 - pCreateProfileOut, 994
 - pProfileID, 994
 - ParamPresenceMask, 994
 - Tlvresult, 994
- unpack_wds_SLQSDUNCallInfoCallBack_ind
 - wds.h, 1558
- unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 994
- unpack_wds_SLQSDDeleteProfile
 - wds.h, 1558
- unpack_wds_SLQSDDeleteProfile_t, 994
 - extendedErrorCode, 994
 - ParamPresenceMask, 994
- unpack_wds_SLQSGet3GPPConfigItem
 - wds.h, 1558
- unpack_wds_SLQSGet3GPPConfigItem_t, 996
 - _3gppRelease, 997
 - defaultPDNEnabled, 997
 - LTEAttachProfile, 997
 - ParamPresenceMask, 998
 - profileList, 998
- unpack_wds_SLQSGetCurrDataSystemStat
 - wds.h, 1559
- unpack_wds_SLQSGetCurrDataSystemStat_t, 998
 - currNetworkInfo, 998
 - networkInfoLen, 998
 - ParamPresenceMask, 998
 - prefNetwork, 999
- unpack_wds_SLQSGetCurrentChannelRate
 - wds.h, 1559
- unpack_wds_SLQSGetCurrentChannelRate_t, 999
 - max_channel_rx_rate, 1000
 - max_channel_tx_rate, 1000
 - ParamPresenceMask, 1000
- unpack_wds_SLQSGetDUNCallInfo
 - wds.h, 1560
- unpack_wds_SLQSGetDUNCallInfo_t, 1001
 - callEndReason, 1003
 - channelRate, 1003
 - connectionStatus, 1003
 - dataBearerTech, 1003
 - dormancyStatus, 1003
 - lastCallDataBearerTech, 1003
 - mdmCallDurationActive, 1004
 - ParamPresenceMask, 1004
 - rxOKBytesCount, 1004
 - txOKBytesCount, 1004
- unpack_wds_SLQSGetDataBearerTechnology
 - wds.h, 1559
- unpack_wds_SLQSGetDataBearerTechnology_t, 1000
 - curDataBearerTechnology, 1000
 - dataBearerMask, 1000
 - lastCallDataBearerTechnology, 1001
 - ParamPresenceMask, 1001
- unpack_wds_SLQSGetProfileSettings
 - wds.h, 1560
- unpack_wds_SLQSGetProfileSettings_t, 1004
 - pProfileSettings, 1004
 - ParamPresenceMask, 1004
 - ProfileType, 1004
 - Tlvresult, 1004
- unpack_wds_SLQSGetProfileSettingsV2
 - wds.h, 1560
- unpack_wds_SLQSGetProfileSettingsV2_t, 1004
 - pProfileSettings, 1005
 - ParamPresenceMask, 1005
 - ProfileType, 1005
 - Tlvresult, 1005
- unpack_wds_SLQSGetRuntimeSettings
 - wds.h, 1561
- unpack_wds_SLQSGetRuntimeSettings_t, 1005
 - APNName, 1007
 - Authentication, 1007
 - DomainList, 1007
 - GPRSGrantedQoS, 1007
 - GWAddressV4, 1007
 - IMCNflag, 1008
 - IPFamilyPreference, 1008
 - IPv6AddrInfo, 1008
 - IPv6GWAddrInfo, 1008
 - IPv4, 1008
 - Mtu, 1008
 - PDPTType, 1008
 - ParamPresenceMask, 1008
 - PrimaryDNSV4, 1008
 - PrimaryDNSV6, 1008
 - ProfileID, 1008
 - ProfileName, 1008
 - SecondaryDNSV4, 1008
 - SecondaryDNSV6, 1008
 - ServerAddrList, 1008
 - SubnetMaskV4, 1008
 - Technology, 1008
 - UMTSGrantedQoS, 1008
 - Username, 1008
- unpack_wds_SLQSModifyProfile
 - wds.h, 1561
- unpack_wds_SLQSModifyProfile_t, 1008
 - pExtErrorCode, 1009
 - ParamPresenceMask, 1009
- unpack_wds_SLQSResetPacketStatics
 - wds.h, 1561
- unpack_wds_SLQSResetPacketStatics_t
 - wds.h, 1526

- unpack_wds_SLQSSGetDHCPv4ClientConfig
 - wds.h, [1563](#)
- unpack_wds_SLQSSGetDHCPv4ClientConfig_t, [1017](#)
 - pHwConfig, [1018](#)
 - ParamPresenceMask, [1018](#)
- unpack_wds_SLQSSGetLoopback
 - wds.h, [1564](#)
- unpack_wds_SLQSSGetLoopback_t, [1018](#)
 - ByteLoopbackMode, [1019](#)
 - ByteLoopbackMultiplier, [1019](#)
 - ParamPresenceMask, [1019](#)
- unpack_wds_SLQSSSetDHCPv4ClientConfig
 - wds.h, [1564](#)
- unpack_wds_SLQSSSetDHCPv4ClientConfig_t
 - wds.h, [1526](#)
- unpack_wds_SLQSSSetLoopback
 - wds.h, [1564](#)
- unpack_wds_SLQSSSetLoopback_t
 - wds.h, [1526](#)
- unpack_wds_SLQSSSet3GPPConfigItem
 - wds.h, [1562](#)
- unpack_wds_SLQSSSet3GPPConfigItem_t
 - wds.h, [1526](#)
- unpack_wds_SLQSSSetIPFamilyPreference
 - wds.h, [1562](#)
- unpack_wds_SLQSSSetIPFamilyPreference_t, [1009](#)
 - ParamPresenceMask, [1009](#)
 - Tlvresult, [1009](#)
- unpack_wds_SLQSSSetPacketSrvStatusCallback
 - wds.h, [1562](#)
- unpack_wds_SLQSSSetPacketSrvStatusCallback_t, [1009](#)
 - bearerID, [1011](#)
 - conn_status, [1011](#)
 - ipFamily, [1011](#)
 - ParamPresenceMask, [1011](#)
 - reconfigReqd, [1011](#)
 - sessionEndReason, [1011](#)
 - techName, [1011](#)
 - verboseSessnEndReason, [1011](#)
 - verboseSessnEndReasonType, [1011](#)
- unpack_wds_SLQSSSetWdsEventCallback
 - wds.h, [1563](#)
- unpack_wds_SLQSSSetWdsEventCallback_ind
 - wds.h, [1563](#)
- unpack_wds_SLQSSSetWdsEventCallback_ind_t, [1011](#)
 - currDBTechAvail, [1016](#)
 - currNWInfo, [1016](#)
 - dBTechAvail, [1016](#)
 - dBTechnology, [1017](#)
 - dBtechExtAvail, [1016](#)
 - dBtechnologyExt, [1017](#)
 - dataSysStatAvail, [1016](#)
 - dormancyStatAvail, [1017](#)
 - dormancyStatus, [1017](#)
 - mipStatus, [1017](#)
 - mipstatAvail, [1017](#)
 - netInfoLen, [1017](#)
 - ParamPresenceMask, [1017](#)
 - prefNetwork, [1017](#)
 - ratMask, [1017](#)
 - rx_bytes, [1017](#)
 - rx_pkts, [1017](#)
 - soMask, [1017](#)
 - tx_bytes, [1017](#)
 - tx_pkts, [1017](#)
 - xferStatAvail, [1017](#)
- unpack_wds_SLQSSSetWdsEventCallback_t
 - wds.h, [1526](#)
- unpack_wds_SLQSSStartDataSession
 - wds.h, [1565](#)
- unpack_wds_SLQSSStartDataSession_t, [1019](#)
 - pFailureReason, [1019](#)
 - pVerboseFailReasonType, [1020](#)
 - pVerboseFailureReason, [1020](#)
 - ParamPresenceMask, [1019](#)
 - psid, [1019](#)
- unpack_wds_SLQSSStopDataSession
 - wds.h, [1565](#)
- unpack_wds_SLQSSStopDataSession_t
 - wds.h, [1526](#)
- unpack_wds_SLQSWdsGoActive
 - wds.h, [1565](#)
- unpack_wds_SLQSWdsGoActive_t
 - wds.h, [1526](#)
- unpack_wds_SLQSWdsGoDormant
 - wds.h, [1566](#)
- unpack_wds_SLQSWdsGoDormant_t
 - wds.h, [1526](#)
- unpack_wds_SLQSWdsSetEventReport
 - wds.h, [1566](#)
- unpack_wds_SLQSWdsSetEventReport_t
 - wds.h, [1526](#)
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, [1566](#)
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [1020](#)
 - apnName, [1022](#)
 - bearerId, [1022](#)
 - contextId, [1022](#)
 - ipv4Address, [1022](#)
 - ipv4GWAddress, [1022](#)
 - ipv6Address, [1022](#)
 - ipv6GWAddress, [1022](#)
 - ParamPresenceMask, [1022](#)
- unpack_wds_SetAutoconnect
 - wds.h, [1555](#)
- unpack_wds_SetAutoconnect_t
 - wds.h, [1526](#)
- unpack_wds_SetDefaultProfile
 - wds.h, [1555](#)
- unpack_wds_SetDefaultProfile_t
 - wds.h, [1526](#)
- unpack_wds_SetDefaultProfileNum
 - wds.h, [1556](#)
- unpack_wds_SetDefaultProfileNum_t

- wds.h, 1526
- unpack_wds_SetMobileIP
 - wds.h, 1556
- unpack_wds_SetMobileIP_t
 - wds.h, 1526
- unpack_wds_SetMobileIPParameters
 - wds.h, 1556
- unpack_wds_SetMobileIPParameters_t
 - wds.h, 1526
- unpack_wds_SetMobileIPProfile
 - wds.h, 1557
- unpack_wds_SetMobileIPProfile_t, 993
 - ParamPresenceMask, 993
- unpack_wds_SetMuxID
 - wds.h, 1557
- unpack_wds_SetMuxID_t
 - wds.h, 1526
- UnpackQmiProfileInfo
 - wds.h, 1526
- UnpackQmiProfileInfoV2
 - wds.h, 1527
- UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, 1023
 - bBinaryType, 1025
 - bSerty, 1025
 - bState, 1025
 - bUserInputRequest, 1025
 - szDescription, 1025
 - szName, 1025
 - szVersion, 1025
 - TlvPresent, 1025
 - ulPkgDownloadComplete, 1025
 - ulPkgDownloadSize, 1025
 - wDescriptionLength, 1025
 - wNameLength, 1025
 - wUpdateCompleteStatus, 1025
 - wUserInputTimeout, 1025
 - wVersionLength, 1025
- UnpackSwiAvmsEventReportConfig, 1025
 - bState, 1026
 - bUserInputRequest, 1026
 - szAlertMsg, 1026
 - TlvPresent, 1026
 - wAlertMsgLength, 1026
 - wUserInputTimeout, 1026
- UnpackSwiAvmsEventReportConnectionRequest, 1026
 - bUserInputRequest, 1027
 - TlvPresent, 1027
 - wUserInputTimeout, 1027
- UnpackSwiAvmsEventReportDataSessionStatus, 1027
 - bType, 1027
 - TlvPresent, 1027
 - wErrorCode, 1027
- UnpackSwiAvmsEventReportHTTPStatus, 1027
 - TlvPresent, 1028
 - wHTTPStatus, 1028
- UnpackSwiAvmsEventReportNotification, 1029
 - bNotification, 1029
- TlvPresent, 1030
- wSessionStatus, 1030
- UnpackSwiAvmsEventReportPackageID, 1030
 - bPackageID, 1030
 - TlvPresent, 1030
- UnpackSwiAvmsEventReportRegStatus, 1030
 - bRegStatus, 1030
 - TlvPresent, 1030
- UnpackSwiAvmsEventReportSessionType, 1031
 - bType, 1031
 - TlvPresent, 1031
- UnpackSwiAvmsEventReportWAMSPParamChange, 1031
 - TlvPresent, 1032
 - wWamsChangeMask, 1032
- unpackWdsProfileParam, 1032
 - SlqsProfile3GPP, 1032
 - SlqsProfile3GPP2, 1032
- unpackWdsProfileParamV2, 1032
 - SlqsProfile3GPP, 1033
 - SlqsProfile3GPP2, 1033
- upLink
 - voice_NSSAudioCtrl, 1069
- UpdateCompleteStatus
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, 903
- updateCompleteStatus
 - unpack_omaDmFotaTlv_t, 819
- upinRetries
 - slotInf, 549
 - uim_slotInfo, 602
- upinState
 - slotInf, 549
 - uim_slotInfo, 602
- upukRetries
 - slotInf, 549
 - uim_slotInfo, 602
- urlAddr
 - loc_URLAddrInfo, 166
- usageMask
 - loc_sensorDataUsage, 163
- UsbMTUSize
 - swidms_usbMTUSizeTlv, 573
- usbNetNum
 - unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 895
- userData
 - unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 877
- userInputReq
 - unpack_omaDmConfigTlv_t, 817
 - unpack_omaDmFotaTlv_t, 819
- userInputTimeout
 - unpack_omaDmConfigTlv_t, 817
 - unpack_omaDmFotaTlv_t, 819
- Username
 - unpack_wds_SLQSGetRuntimeSettings_t, 1008
- username

- unpack_wds_GetDefaultProfile_t, 979
- unpack_wds_GetDefaultProfileV2_t, 982
- usersize
 - unpack_wds_GetDefaultProfile_t, 979
 - unpack_wds_GetDefaultProfileV2_t, 982
- ussDCS
 - pack_voice_SLQSOriGinateUSSD_t, 493
 - voice_USSInfo, 1079
- ussData
 - pack_voice_SLQSOriGinateUSSD_t, 493
 - voice_USSInfo, 1079
- ussLen
 - pack_voice_SLQSOriGinateUSSD_t, 493
 - voice_USSInfo, 1079
- UtRat
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 710
- UtRatVal
 - imsa_UtRatInfo, 102
- UtService
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 710
- UtServiceRat
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, 706
- UtServiceStatus
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, 706
- UtSvcStatus
 - imsa_UtSvcStatusInfo, 102
- uusInfo
 - voice_allCallsUUSInfo, 1035
- VDOP
 - loc_precisionDilution, 160
- val
 - unpack_qos_IPv6TrafCls_t, 836
 - unpack_qos_Tos_t, 855
- val_3GPP2Pri
 - unpack_qos_swiQosFlow_t, 854
- val_3GPPImCn
 - unpack_qos_swiQosFlow_t, 854
- val_3GPPResResidualBER
 - unpack_qos_swiQosFlow_t, 854
- val_3GPPSigInd
 - unpack_qos_swiQosFlow_t, 854
- val_3GPPTraHdlPri
 - unpack_qos_swiQosFlow_t, 854
- ValidBitmasks
 - swidms_supportedIntBitmaskTlv, 572
- validCard
 - uim_validCardStatus, 604
- validCardLength
 - uim_validCardStatus, 605
- ValidMask
 - unpack_pds_SLQSGetGPSSStateInfo_t, 833
- validMask
 - loc_satelliteInfo, 162
- ValidityCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, 810
- ValidityCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, 810
- Value
 - pack_swiaudio_SLQSSetM2MSpkrGain_t, 462
 - unpack_swiaudio_SLQSGetM2MSpkrGain_t, 882
- value
 - pack_dms_UIMSetPINProtection_t, 367
 - pack_dms_UIMVerifyPIN_t, 368
- value_length
 - DMScustSettingInfo, 75
 - pack_dms_SetCustFeaturesV2_t, 356
- verboseSessnEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1011
- verboseSessnEndReasonType
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 1011
- verifyLeft
 - uim_personalizationStatus, 593
 - uim_remainingRetries, 598
- verifyPIN
 - pack_uim_VerifyPin_t, 491
- verifyRetriesLeft
 - unpack_dms_UIMGetControlKeyStatus_t, 681
 - unpack_dms_UIMSetControlKeyProtection_t, 684
 - unpack_dms_UIMSetPINProtection_t, 685
- version
 - unpack_omaDmFotaTlv_t, 819
 - unpack_qos_swiQosFilter_t, 850
- versionStr
 - _litefw_FirmwareFileInfo, 36
- versionString
 - pack_dms_SLQSSwiSetOSInfo_t, 362
 - unpack_dms_SLQSSwiGetOSInfo_t, 675
- versionlength
 - unpack_omaDmFotaTlv_t, 819
- vertConfidence
 - pack_loc_SLQSLOCInjectPosition_t, 388
- vertReliability
 - pack_loc_SLQSLOCInjectPosition_t, 388
- vertUnc
 - pack_loc_SLQSLOCInjectPosition_t, 388
- VerticalUncertainty
 - unpack_pds_SLQSGetGPSSStateInfo_t, 833
- VirtStream
 - nas_protocolSubtypeElement, 272
- voice.h
 - LITE_VOICE_SUPS_SRV_CLASS_DATA, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_DATAACIRCUI-TASYNC, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_DATAACIRCUI-TSYNC, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_FAX, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_NONE, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_PADACCESS, 1492
 - LITE_VOICE_SUPS_SRV_CLASS_SMS, 1492

- LITE_VOICE_SUPS_SRV_CLASS_VOICE, 1492
- voice.h, 1485
 - liteServiceClassInformation, 1491
 - MAXVOICEUSSDLENGTH, 1491
 - pack_voice_AnswerUSSD, 1492
 - pack_voice_CancelUSSD, 1492
 - pack_voice_OriginateUSSD, 1493
 - pack_voice_SLQSOrginateUSSD, 1493
 - pack_voice_SLQSVoiceALSSelectLine, 1493
 - pack_voice_SLQSVoiceALSSetLineSwitching, 1494
 - pack_voice_SLQSVoiceAnswerCall, 1494
 - pack_voice_SLQSVoiceBindSubscription, 1494
 - pack_voice_SLQSVoiceBurstDTMF, 1495
 - pack_voice_SLQSVoiceDialCall, 1495
 - pack_voice_SLQSVoiceEndCall, 1495
 - pack_voice_SLQSVoiceGetAllCallInfo, 1496
 - pack_voice_SLQSVoiceGetCLIP, 1498
 - pack_voice_SLQSVoiceGetCLIR, 1498
 - pack_voice_SLQSVoiceGetCNAP, 1498
 - pack_voice_SLQSVoiceGetCOLP, 1499
 - pack_voice_SLQSVoiceGetCOLR, 1499
 - pack_voice_SLQSVoiceGetCallBarring, 1496
 - pack_voice_SLQSVoiceGetCallForwardingStatus, 1497
 - pack_voice_SLQSVoiceGetCallInfo, 1497
 - pack_voice_SLQSVoiceGetCallWaiting, 1497
 - pack_voice_SLQSVoiceGetConfig, 1499
 - pack_voice_SLQSVoiceIndicationRegister, 1500
 - pack_voice_SLQSVoiceManageCalls, 1500
 - pack_voice_SLQSVoiceOrigUSSDNoWait, 1500
 - pack_voice_SLQSVoiceSendFlash, 1501
 - pack_voice_SLQSVoiceSetCallBarringPassword, 1501
 - pack_voice_SLQSVoiceSetConfig, 1501
 - pack_voice_SLQSVoiceSetPreferredPrivacy, 1502
 - pack_voice_SLQSVoiceSetSUPSService, 1502
 - pack_voice_SLQSVoiceStartContDTMF, 1503
 - pack_voice_SLQSVoiceStopContDTMF, 1503
 - unpack_voice_AnswerUSSD, 1504
 - unpack_voice_AnswerUSSD_t, 1491
 - unpack_voice_CancelUSSD, 1504
 - unpack_voice_CancelUSSD_t, 1491
 - unpack_voice_DTMFEventCallback_ind, 1504
 - unpack_voice_OTASPStatusCallback_ind, 1505
 - unpack_voice_OriginateUSSD, 1505
 - unpack_voice_OriginateUSSD_t, 1491
 - unpack_voice_SLQSOrginateUSSD, 1505
 - unpack_voice_SLQSVoiceALSSelectLine, 1506
 - unpack_voice_SLQSVoiceALSSelectLine_t, 1491
 - unpack_voice_SLQSVoiceALSSetLineSwitching, 1506
 - unpack_voice_SLQSVoiceALSSetLineSwitching_t, 1491
 - unpack_voice_SLQSVoiceAnswerCall, 1507
 - unpack_voice_SLQSVoiceBindSubscription, 1507
 - unpack_voice_SLQSVoiceBindSubscription_t, 1491
 - unpack_voice_SLQSVoiceBurstDTMF, 1507
 - unpack_voice_SLQSVoiceDialCall, 1508
 - unpack_voice_SLQSVoiceEndCall, 1508
 - unpack_voice_SLQSVoiceGetAllCallInfo, 1508
 - unpack_voice_SLQSVoiceGetCLIP, 1510
 - unpack_voice_SLQSVoiceGetCLIR, 1510
 - unpack_voice_SLQSVoiceGetCNAP, 1511
 - unpack_voice_SLQSVoiceGetCOLP, 1511
 - unpack_voice_SLQSVoiceGetCOLR, 1511
 - unpack_voice_SLQSVoiceGetCallBarring, 1509
 - unpack_voice_SLQSVoiceGetCallForwarding-Status, 1509
 - unpack_voice_SLQSVoiceGetCallInfo, 1509
 - unpack_voice_SLQSVoiceGetCallWaiting, 1510
 - unpack_voice_SLQSVoiceGetConfig, 1512
 - unpack_voice_SLQSVoiceIndicationRegister, 1512
 - unpack_voice_SLQSVoiceIndicationRegister_t, 1491
 - unpack_voice_SLQSVoiceManageCalls, 1512
 - unpack_voice_SLQSVoiceOrigUSSDNoWait, 1513
 - unpack_voice_SLQSVoiceOrigUSSDNoWait_t, 1491
 - unpack_voice_SLQSVoiceSUPSCallback_ind, 1515
 - unpack_voice_SLQSVoiceSendFlash, 1513
 - unpack_voice_SLQSVoiceSetCallBarringPassword, 1513
 - unpack_voice_SLQSVoiceSetConfig, 1514
 - unpack_voice_SLQSVoiceSetPreferredPrivacy, 1514
 - unpack_voice_SLQSVoiceSetPreferredPrivacy_t, 1491
 - unpack_voice_SLQSVoiceSetSUPSService, 1514
 - unpack_voice_SLQSVoiceStartContDTMF, 1515
 - unpack_voice_SLQSVoiceStopContDTMF, 1515
 - unpack_voice_SUPSNotificationCallback_ind, 1516
 - unpack_voice_USSDNotificationCallback_ind, 1516
 - unpack_voice_VoiceInfoRecCallback_ind, 1516
 - unpack_voice_allCallStatusCallback_ind, 1503
 - unpack_voice_voicePrivacyChangeCallback_ind, 1517
- voice_CLIPResp, 1055
 - ActiveStatus, 1055
 - ProvisionStatus, 1055
- voice_CLIRResp, 1055
 - ActiveStatus, 1056
 - ProvisionStatus, 1056
- voice_CNAPResp, 1056
 - ActiveStatus, 1057
 - ProvisionStatus, 1057
- voice_COLPResp, 1057
 - ActiveStatus, 1057
 - ProvisionStatus, 1057
- voice_COLRResp, 1057
 - ActiveStatus, 1058
 - ProvisionStatus, 1058

- voice_CUGInfo, 1060
 - CUGIndex, 1060
 - SuppOA, 1060
 - SuppPrefCUG, 1060
- voice_DTMFInfo, 1062
 - callID, 1062
 - DTMFEvent, 1062
 - digitBuff, 1062
 - digitCnt, 1062
- voice_DTMFLengths, 1062
 - DTMFInterdigitInterval, 1063
 - DTMFPulseWidth, 1063
- voice_ECTNum, 1063
 - ECTCallState, 1064
 - number, 1064
 - presentationInd, 1064
- voice_NSSAudioCtrl, 1069
 - downLink, 1069
 - upLink, 1069
- voice_SUPSInfo, 1078
 - isModByCC, 1078
 - svcType, 1078
- voice_USSDNotificationNetworkInfo, 1078
 - networkInfo, 1079
 - tlvPresent, 1079
- voice_USSInfo, 1079
 - ussDCS, 1079
 - ussData, 1079
 - ussLen, 1079
- voice_UUSInfo, 1079
 - UUSData, 1080
 - UUSDataLen, 1080
 - UUSDcs, 1080
 - UUSType, 1080
- voice_airTimer, 1033
 - airTimerValue, 1033
 - namID, 1033
- voice_allCallsAlphaIDInfo, 1033
 - AlphaIDInfo, 1034
 - callID, 1034
- voice_allCallsDiagInfo, 1034
 - callID, 1034
 - DiagInfo, 1034
- voice_allCallsUUSInfo, 1034
 - callID, 1035
 - uusInfo, 1035
- voice_alphaIDInfo, 1035
 - alphaDcs, 1035
 - alphaLen, 1035
 - alphaText, 1035
- voice_arrAlertingPattern, 1035
 - alertingPattern, 1036
 - callID, 1036
 - numInstances, 1036
- voice_arrAlertingType, 1036
 - AlertingType, 1037
 - callID, 1037
 - numInstances, 1037
- voice_arrAlphaID, 1037
 - allCallsAlphaIDInfoArr, 1037
 - numInstances, 1037
- voice_arrCallEndReason, 1038
 - callEndReason, 1039
 - callID, 1039
 - numInstances, 1039
- voice_arrCallInfo, 1039
 - getAllCallInfo, 1039
 - numInstances, 1039
- voice_arrCalledPartyNum, 1037
 - CalledPartyNum, 1038
 - numInstances, 1038
- voice_arrConnectPartyNum, 1039
 - ConnectedPartyNum, 1040
 - numInstances, 1040
- voice_arrDiagInfo, 1040
 - DiagInfo, 1040
 - numInstances, 1040
- voice_arrRedirPartyNum, 1040
 - numInstances, 1041
 - RedirPartyNum, 1041
- voice_arrRemotePartyName, 1041
 - GetAllCallRmtPtyName, 1041
 - numInstances, 1041
- voice_arrRemotePartyNum, 1041
 - numInstances, 1042
 - RmtPtyNum, 1042
- voice_arrSvcOption, 1042
 - callID, 1042
 - numInstances, 1042
 - srvOption, 1042
- voice_arrUUSInfo, 1043
 - AllCallsUUSInfo, 1043
 - numInstances, 1043
- voice_burstDTMFInfo, 1043
 - digitCnt, 1044
 - pCallID, 1044
 - pDigitBuff, 1044
- voice_callFWExtInfo, 1048
 - noReplyTimer, 1049
 - numLen, 1049
 - numPlan, 1050
 - numType, 1050
 - number, 1049
 - PI, 1050
 - SI, 1050
 - SvcClass, 1050
 - SvcStatus, 1050
- voice_callFWInfo, 1050
 - noReplyTimer, 1050
 - numLen, 1051
 - number, 1050
 - SvcClass, 1051
 - SvcStatus, 1051
- voice_callFwdTypeAndPlan, 1047
 - numberPlan, 1048
 - numberType, 1048

- voice_callInfo, 1051
 - callID, 1052
 - callState, 1052
 - callType, 1052
 - direction, 1052
 - mode, 1052
- voice_calledPartyInfo, 1044
 - numLen, 1045
 - numPlan, 1045
 - numType, 1045
 - number, 1045
 - PI, 1045
 - SI, 1045
- voice_calledPartySubAdd, 1045
 - extBit, 1046
 - oddEvenInd, 1046
 - subAddr, 1046
 - subAddrLen, 1046
 - subAddrType, 1046
- voice_callerIDInfo, 1046
 - callerID, 1047
 - callerIDLen, 1047
 - PI, 1047
- voice_callingPartyInfo, 1052
 - numLen, 1054
 - numPlan, 1054
 - numType, 1054
 - number, 1054
 - PI, 1054
 - SI, 1054
- voice_ccSUPSType, 1054
 - reason, 1054
 - svcType, 1055
- voice_connectNumInfo, 1058
 - callerID, 1059
 - callerIDLen, 1059
 - numPlan, 1059
 - numPresInd, 1059
 - numType, 1060
 - screeningInd, 1060
- voice_curAMRConfig, 1060
 - gsmAmrStat, 1061
 - wcdmaAmrStat, 1061
- voice_diagInfo, 1061
 - diagInfoLen, 1061
 - diagnosticInfo, 1061
- voice_extDispRecInfo, 1064
 - dispType, 1064
 - extDispInfo, 1064
 - extDispInfoLen, 1064
- voice_getAllCallInformation, 1065
 - ALS, 1065
 - Callinfo, 1065
 - isEmpty, 1065
- voice_getAllCallRmtPtyName, 1065
 - callID, 1066
 - RemotePartyName, 1066
- voice_getAllCallRmtPtyNum, 1066
 - callID, 1066
 - RemotePartyNum, 1066
- voice_getCallFWExtInfo, 1066
 - CallFWExtInfo, 1067
 - numInstances, 1067
- voice_getCallFWInfo, 1067
 - CallFWInfo, 1067
 - numInstances, 1067
- voice_lineCtrlInfo, 1067
 - polarityIncluded, 1068
 - pwrDenialTime, 1068
 - revPolarity, 1068
 - toggleMode, 1068
- voice_newPwdData, 1068
 - newPwd, 1069
 - newPwdAgain, 1069
- voice_peerNumberInfo, 1069
 - callID, 1070
 - numLen, 1071
 - numPI, 1071
 - numPlan, 1071
 - numSI, 1071
 - numType, 1071
 - number, 1071
- voice_prefVoiceSO, 1071
 - evrcCapability, 1073
 - homeOrigVoiceSO, 1073
 - homePageVoiceSO, 1073
 - namID, 1073
 - roamOrigVoiceSO, 1073
- voice_redirNumInfo, 1073
 - numLen, 1074
 - numPlan, 1074
 - numType, 1074
 - number, 1074
 - PI, 1074
 - reason, 1074
 - SI, 1074
- voice_remotePartyName, 1075
 - callerName, 1075
 - codingScheme, 1075
 - nameLen, 1075
 - namePI, 1075
- voice_remotePartyNum, 1075
 - numLen, 1076
 - presentationInd, 1076
 - remPartyNumber, 1076
- voice_roamTimer, 1076
 - namID, 1077
 - roamTimerValue, 1077
- voice_signalInfo, 1077
 - alertPitch, 1077
 - signal, 1077
 - signalType, 1077
- VoiceNumber
 - unpack_dms_GetVoiceNumber_t, 643
- voiceNumberSize
 - unpack_dms_GetVoiceNumber_t, 643

- voicePrivacy
 - unpack_voice_voicePrivacyChangeCallback_ind_t, [972](#)
- voiceSvc
 - pack_voice_SLQSVoiceSetSUPSService_t, [509](#)
- VoipRat
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, [710](#)
- VoipRatVal
 - imsa_VoipRatInfo, [103](#)
- VoipService
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, [710](#)
- VoipServiceRat
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, [706](#)
- VoipServiceStatus
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, [706](#)
- VoipSvcStatus
 - imsa_VoipSvcStatusInfo, [103](#)
- VolValue
 - pack_audio_SLQSSetAudioVolTLBConfig_t, [348](#)
- VoltStat
 - dms_VoltageTlv, [74](#)
- VoltTlv
 - unpack_dms_SwiEventReportCallBack_ind_t, [679](#)
- Voltage
 - dms_VoltageTlv, [74](#)
- Volume
 - pack_audio_SLQSGetAudioVolTLBConfig_t, [344](#)
 - pack_audio_SLQSSetAudioProfile_t, [347](#)
 - pack_audio_SLQSSetAudioVolTLBConfig_t, [348](#)
 - unpack_audio_SLQSGetAudioProfile_t, [609](#)
 - unpack_swiaudio_SLQSGetM2MAudioProfile_t, [880](#)
- voteForInit
 - uim_registerRefresh, [598](#)
- VsServiceRat
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, [706](#)
- VsServiceStatus
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, [706](#)
- VtRat
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, [710](#)
- VtRatVal
 - imsa_VtRatInfo, [104](#)
- VtService
 - unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, [710](#)
- VtServiceRat
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, [706](#)
- VtServiceStatus
 - unpack_imsa_SLQSGetIMSAServiceStatus_t, [706](#)
- VtSvcStatus
 - imsa_VtSvcStatusInfo, [104](#)
- wAlertMsgLength
 - UnpackSwiAvmsEventReportConfig, [1026](#)
- WCDMACellInfo
 - nas_lteWcdmaCellInfo, [244](#)
- WCDMAECIOThreshListLen
 - nas_WCDMAECIOThresh, [316](#)
- WCDMARSSIOThreshListLen
 - nas_WCDMARSSIOThresh, [318](#)
- WCDMASSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [796](#)
- WDS_PROFILE_3GPP
 - wds.h, [1526](#)
- WDS_PROFILE_3GPP2
 - wds.h, [1526](#)
- wDescriptionLength
 - UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, [1025](#)
- wErrorCode
 - UnpackSwiAvmsEventReportDataSessionStatus, [1027](#)
- wHTTPStatus
 - UnpackSwiAvmsEventReportHTTPStatus, [1028](#)
- wNameLength
 - UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, [1025](#)
- WORD
 - SwiDataTypes.h, [1433](#)
- wSessionStatus
 - UnpackSwiAvmsEventReportNotification, [1030](#)
- wUpdateCompeteStatus
 - UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, [1025](#)
- wUserInputTimeout
 - UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, [1025](#)
 - UnpackSwiAvmsEventReportConfig, [1026](#)
 - UnpackSwiAvmsEventReportConnectionRequest, [1027](#)
- wVersionLength
 - UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, [1025](#)
- wWamsChangeMask
 - UnpackSwiAvmsEventReportWAMSParmChange, [1032](#)
- wake_mask
 - pack_rms_SetSMSWake_t, [446](#)
 - unpack_rms_GetSMSWake_t, [857](#)
- wcdmaAmrStat
 - voice_curAMRConfig, [1061](#)
- wcdmaRRState
 - nas_WCDMAInfoLTENeighborCell, [317](#)
- Wdisable
 - unpack_dms_SLQSDmsSwiGetPCInfo_t, [653](#)
- wds.h
 - QMI_LITE_WDS_CURRENT_CALL_DB_MASK, [1527](#)
 - QMI_LITE_WDS_LAST_CALL_DB_MASK, [1527](#)
- wds.h, [1517](#)
 - BYT_STAT_STAT_MASK, [1525](#)
 - liteQmiDataBearerMasks, [1527](#)
 - PACK_WDS_IPV4, [1526](#)
 - PACK_WDS_IPV6, [1526](#)

- pack_wds_DHCPv4ClientLeaseChange, 1527
- pack_wds_GetAutoconnect, 1528
- pack_wds_GetByteTotals, 1528
- pack_wds_GetConnectionRate, 1528
- pack_wds_GetDataBearerTechnology, 1529
- pack_wds_GetDefaultProfile, 1529
- pack_wds_GetDefaultProfileNum, 1530
- pack_wds_GetDefaultProfileV2, 1530
- pack_wds_GetDormancyState, 1530
- pack_wds_GetLastMobileIPError, 1531
- pack_wds_GetMobileIP, 1531
- pack_wds_GetMobileIPProfile, 1532
- pack_wds_GetPacketStatistics, 1532
- pack_wds_GetPacketStatus, 1533
- pack_wds_GetSessionDuration, 1533
- pack_wds_GetSessionDurationV2, 1533
- pack_wds_GetSessionState, 1534
- pack_wds_RMSetTransferStatistics, 1534
- pack_wds_SLQSCreateProfile, 1538
- pack_wds_SLQSDeleteProfile, 1538
- pack_wds_SLQSGet3GPPConfigItem, 1538
- pack_wds_SLQSGetCurrDataSystemStat, 1539
- pack_wds_SLQSGetCurrentChannelRate, 1539
- pack_wds_SLQSGetDUNCallInfo, 1540
- pack_wds_SLQSGetDataBearerTechnology, 1540
- pack_wds_SLQSGetProfileSettings, 1541
- pack_wds_SLQSGetProfileSettingsV2, 1541
- pack_wds_SLQSGetRuntimeSettings, 1541
- pack_wds_SLQSModifyProfile, 1542
- pack_wds_SLQSResetPacketStatics, 1542
- pack_wds_SLQSSetDHCPv4ClientConfig, 1544
- pack_wds_SLQSSetLoopback, 1544
- pack_wds_SLQSSetDHCPv4ClientConfig, 1545
- pack_wds_SLQSSetLoopback, 1545
- pack_wds_SLQSSet3GPPConfigItem, 1543
- pack_wds_SLQSSetIPFamilyPreference, 1543
- pack_wds_SLQSSetWdsEventCallback, 1543
- pack_wds_SLQSStartDataSession, 1545
- pack_wds_SLQSStopDataSession, 1546
- pack_wds_SLQSWdsGoActive, 1546
- pack_wds_SLQSWdsGoDormant, 1547
- pack_wds_SLQSWdsSetEventReport, 1547
- pack_wds_SLQSWdsSwiPDPRuntimeSettings, 1548
- pack_wds_SetAutoconnect, 1535
- pack_wds_SetDefaultProfile, 1535
- pack_wds_SetDefaultProfileNum, 1535
- pack_wds_SetMobileIP, 1536
- pack_wds_SetMobileIPParameters, 1536
- pack_wds_SetMobileIPProfile, 1537
- pack_wds_SetMuxID, 1537
- unpack_wds_DHCPv4ClientLease_ind, 1548
- unpack_wds_DHCPv4ClientLeaseChange, 1548
- unpack_wds_DHCPv4ClientLeaseChange_t, 1526
- unpack_wds_GetAutoconnect, 1549
- unpack_wds_GetByteTotals, 1549
- unpack_wds_GetConnectionRate, 1549
- unpack_wds_GetDataBearerTechnology, 1550
- unpack_wds_GetDefaultProfile, 1550
- unpack_wds_GetDefaultProfileNum, 1550
- unpack_wds_GetDefaultProfileV2, 1551
- unpack_wds_GetDormancyState, 1551
- unpack_wds_GetLastMobileIPError, 1552
- unpack_wds_GetMobileIP, 1552
- unpack_wds_GetMobileIPProfile, 1552
- unpack_wds_GetPacketStatistics, 1553
- unpack_wds_GetPacketStatus, 1553
- unpack_wds_GetSessionDuration, 1553
- unpack_wds_GetSessionDurationV2, 1554
- unpack_wds_GetSessionState, 1554
- unpack_wds_RMSetTransferStatistics, 1554
- unpack_wds_RMTransferStatistics_ind, 1555
- unpack_wds_RMTransferStatistics_ind_t, 1526
- unpack_wds_SLQSCreateProfile, 1557
- unpack_wds_SLQSDUNCallInfoCallback_ind, 1558
- unpack_wds_SLQSDeleteProfile, 1558
- unpack_wds_SLQSGet3GPPConfigItem, 1558
- unpack_wds_SLQSGetCurrDataSystemStat, 1559
- unpack_wds_SLQSGetCurrentChannelRate, 1559
- unpack_wds_SLQSGetDUNCallInfo, 1560
- unpack_wds_SLQSGetDataBearerTechnology, 1559
- unpack_wds_SLQSGetProfileSettings, 1560
- unpack_wds_SLQSGetProfileSettingsV2, 1560
- unpack_wds_SLQSGetRuntimeSettings, 1561
- unpack_wds_SLQSModifyProfile, 1561
- unpack_wds_SLQSResetPacketStatics, 1561
- unpack_wds_SLQSResetPacketStatics_t, 1526
- unpack_wds_SLQSSetDHCPv4ClientConfig, 1563
- unpack_wds_SLQSSetLoopback, 1564
- unpack_wds_SLQSSetDHCPv4ClientConfig, 1564
- unpack_wds_SLQSSetDHCPv4ClientConfig_t, 1526
- unpack_wds_SLQSSetLoopback, 1564
- unpack_wds_SLQSSetLoopback_t, 1526
- unpack_wds_SLQSSet3GPPConfigItem, 1562
- unpack_wds_SLQSSet3GPPConfigItem_t, 1526
- unpack_wds_SLQSSetIPFamilyPreference, 1562
- unpack_wds_SLQSSetPacketSrvStatusCallback, 1562
- unpack_wds_SLQSSetWdsEventCallback, 1563
- unpack_wds_SLQSSetWdsEventCallback_ind, 1563
- unpack_wds_SLQSSetWdsEventCallback_t, 1526
- unpack_wds_SLQSStartDataSession, 1565
- unpack_wds_SLQSStopDataSession, 1565
- unpack_wds_SLQSStopDataSession_t, 1526
- unpack_wds_SLQSWdsGoActive, 1565
- unpack_wds_SLQSWdsGoActive_t, 1526
- unpack_wds_SLQSWdsGoDormant, 1566
- unpack_wds_SLQSWdsGoDormant_t, 1526
- unpack_wds_SLQSWdsSetEventReport, 1566
- unpack_wds_SLQSWdsSetEventReport_t, 1526

- unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 1566
- unpack_wds_SetAutoconnect, 1555
- unpack_wds_SetAutoconnect_t, 1526
- unpack_wds_SetDefaultProfile, 1555
- unpack_wds_SetDefaultProfile_t, 1526
- unpack_wds_SetDefaultProfileNum, 1556
- unpack_wds_SetDefaultProfileNum_t, 1526
- unpack_wds_SetMobileIP, 1556
- unpack_wds_SetMobileIP_t, 1526
- unpack_wds_SetMobileIPParameters, 1556
- unpack_wds_SetMobileIPParameters_t, 1526
- unpack_wds_SetMobileIPProfile, 1557
- unpack_wds_SetMuxID, 1557
- unpack_wds_SetMuxID_t, 1526
- UnpackQmiProfileInfo, 1526
- UnpackQmiProfileInfoV2, 1527
- WDS_PROFILE_3GPP, 1526
- WDS_PROFILE_3GPP2, 1526
- wds_ChannelRateTlv, 1081
 - ChannelRate, 1081
 - TlvPresent, 1081
- wds_ConnStatusTlv, 1082
 - MDMConnStatus, 1082
 - TlvPresent, 1082
- wds_DHCPLeaseOptTlv, 1085
 - numOpt, 1085
 - optList, 1085
 - optListData, 1086
 - TlvPresent, 1086
- wds_DHCPLeaseStateTlv, 1086
 - leaseState, 1086
 - TlvPresent, 1086
- wds_DHCPOpt, 1086
 - optCode, 1087
 - optValLen, 1087
 - pOptVal, 1087
- wds_DHCPProfileIdTlv, 1087
 - profileId, 1087
 - profileType, 1087
 - TlvPresent, 1087
- wds_DHCPv4HWConfig, 1087
 - chaddr, 1088
 - chaddrLen, 1088
 - hwType, 1088
- wds_DHCPv4Option, 1088
 - optCode, 1089
 - optVal, 1089
 - optValLen, 1089
- wds_DHCPv4OptionList, 1089
 - numOpt, 1089
 - pOptList, 1089
- wds_DHCPv4ProfileId, 1089
 - profileId, 1090
 - profileType, 1090
- wds_DataBearTechTlv, 1083
 - DataBearerTech, 1084
 - TlvPresent, 1084
- wds_DataULongLongTlv, 1084
 - TlvPresent, 1085
 - ulldata, 1085
- wds_DataULongTlv, 1085
 - TlvPresent, 1085
 - ulldata, 1085
- wds_Domain, 1090
 - domainLen, 1090
 - domainName, 1090
- wds_DomainNameList, 1090
 - domain, 1091
 - numInstances, 1091
- wds_DormStatTlv, 1091
 - DormancyStat, 1091
 - TlvPresent, 1091
- wds_GPRSQoS, 1091
 - delayClass, 1092
 - meanThroughputClass, 1092
 - peakThroughputClass, 1092
 - precedenceClass, 1092
 - reliabilityClass, 1092
- wds_IPV6AddressInfo, 1093
 - IPAddressV6, 1093
 - IPv6PrefixLen, 1093
- wds_IPV6GWAddressInfo, 1093
 - gwAddressV6, 1094
 - gwV6PrefixLen, 1094
- wds_IPv4AdTlv, 1092
 - IPv4Addr, 1093
 - TlvPresent, 1093
- wds_LastMdmCallEndRsnTlv, 1094
 - CallEndReason, 1094
 - TlvPresent, 1094
- wds_PCSCFFQDNAddress, 1094
 - fqdnAddr, 1095
 - fqdnLen, 1095
- wds_PCSCFFQDNAddressList, 1095
 - numInstances, 1095
 - pcsfFQDNAddress, 1095
- wds_PCSCFIPv4ServerAddressList, 1095
 - numInstances, 1096
 - pcscfIPv4Addr, 1096
- wds_ProfileIdentifier, 1096
 - profileIndex, 1096
 - profileType, 1097
- wds_RXBytesOKTlv, 1097
 - RxByteOKCnt, 1097
 - TlvPresent, 1097
- wds_TXBytesOKTlv, 1099
 - TlvPresent, 1099
 - TxByteOKCnt, 1099
- wds_TrStatInd, 1098
 - statsMask, 1099
 - statsPeriod, 1099
- wds_UMTSMInQoS, 1100
 - deliveryErrSDU, 1101
 - grntDownlinkBitrate, 1101
 - grntUplinkBitrate, 1102

- maxDownlinkBitrate, [1102](#)
- maxSDUSize, [1102](#)
- maxUplinkBitrate, [1102](#)
- qosDeliveryOrder, [1102](#)
- resBerRatio, [1102](#)
- sduErrorRatio, [1102](#)
- trafficClass, [1102](#)
- trafficPriority, [1102](#)
- transferDelay, [1102](#)
- wds_channelRate, [1081](#)
 - CurrChanRxRate, [1081](#)
 - CurrChanTxRate, [1081](#)
- wds_currNetworkInfo, [1082](#)
 - NetworkType, [1083](#)
 - RATMask, [1083](#)
 - SOMask, [1083](#)
- wds_profileInfo, [1097](#)
 - SlqsProfile3GPP, [1097](#)
 - SlqsProfile3GPP2, [1097](#)
- wds_transferStatInd, [1098](#)
 - StatsMask, [1098](#)
 - StatsPeriod, [1098](#)
- wdsDhcpv4HwConfig, [1102](#)
 - chaddr, [1102](#)
 - chaddrLen, [1102](#)
 - hwType, [1102](#)
- wdsDhcpv4Option, [1103](#)
 - optCode, [1103](#)
 - optVal, [1103](#)
 - optValLen, [1103](#)
- wdsDhcpv4OptionList, [1103](#)
 - numOpt, [1103](#)
 - pOptList, [1104](#)
- wdsDhcpv4ProfileId, [1104](#)
 - profileId, [1104](#)
 - profileType, [1104](#)
- word
 - swi_uint256_t, [568](#)
- xAxis
 - sensorData_t, [546](#)
- xferStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [1017](#)
- xid
 - pack_loc_SLQSLOCGetBestAvailPos_t, [382](#)
 - pack_qmi_t, [444](#)
 - unpack_qmi_t, [834](#)
- xtra_start_gps_minutes
 - unpack_pds_SLQSGetGPSSStateInfo_t, [833](#)
- xtra_start_gps_week
 - unpack_pds_SLQSGetGPSSStateInfo_t, [834](#)
- xtra_valid_duration_hours
 - unpack_pds_SLQSGetGPSSStateInfo_t, [834](#)
- yAxis
 - sensorData_t, [546](#)
- year
 - nas_timeInfo, [309](#)
 - nas_UniversalTime, [315](#)
- zAxis
 - sensorData_t, [546](#)