

LinuxQMISDK

Lite-0.7.1

Generated by Doxygen 1.8.6

Tue May 31 2016 14:24:35

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	15
5.1	File List	15
6	Module Documentation	17
6.1	QMI pack/unpack (pack)	17
6.1.1	Detailed Description	17
7	Namespace Documentation	19
7.1	Tables Namespace Reference	19
7.1.1	Detailed Description	19
8	Data Structure Documentation	21
8.1	appStats Struct Reference	21
8.1.1	Detailed Description	21
8.1.2	Field Documentation	24
8.1.2.1	aidLength	24
8.1.2.2	aidVal	24

8.1.2.3	appState	24
8.1.2.4	appType	24
8.1.2.5	persoFeature	24
8.1.2.6	persoRetries	24
8.1.2.7	persoState	24
8.1.2.8	persoUnblockRetries	24
8.1.2.9	pin1Retries	24
8.1.2.10	pin1State	24
8.1.2.11	pin2Retries	24
8.1.2.12	pin2State	24
8.1.2.13	puk1Retries	24
8.1.2.14	puk2Retries	25
8.1.2.15	univPin	25
8.2	CarrierImage_t Struct Reference	25
8.2.1	Detailed Description	25
8.2.2	Field Documentation	25
8.2.2.1	m_FwBuildId	26
8.2.2.2	m_FwImageld	26
8.2.2.3	m_nCarrierId	26
8.2.2.4	m_nFolderId	26
8.2.2.5	m_nStorage	26
8.2.2.6	m_PriBuildId	26
8.2.2.7	m_PriImageld	26
8.3	cdmaSSInfo Struct Reference	26
8.3.1	Detailed Description	26
8.3.2	Field Documentation	26
8.3.2.1	ecio	26
8.3.2.2	rssI	26
8.4	connectionStatus Struct Reference	26
8.4.1	Detailed Description	26
8.4.2	Field Documentation	27
8.4.2.1	MDMCallDuration	27
8.4.2.2	MDMConnStatus	27
8.5	currNetworkInfo Struct Reference	27
8.5.1	Detailed Description	27
8.5.2	Field Documentation	27
8.5.2.1	NetworkType	27
8.5.2.2	RATMask	27
8.5.2.3	SOMask	27
8.6	dms_ActivationStatusTlv Struct Reference	27

8.6.1	Detailed Description	27
8.6.2	Field Documentation	28
8.6.2.1	activationStatus	28
8.6.2.2	TlvPresent	28
8.7	dms_OperatingModeTlv Struct Reference	28
8.7.1	Detailed Description	28
8.7.2	Field Documentation	29
8.7.2.1	operatingMode	29
8.7.2.2	TlvPresent	29
8.8	DMScustSettingInfo Struct Reference	29
8.8.1	Detailed Description	29
8.8.2	Field Documentation	29
8.8.2.1	cust_attr	29
8.8.2.2	cust_id	29
8.8.2.3	cust_value	29
8.8.2.4	id_length	29
8.8.2.5	value_length	30
8.9	DMScustSettingList Struct Reference	30
8.9.1	Detailed Description	30
8.9.2	Field Documentation	30
8.9.2.1	custSetting	30
8.9.2.2	list_type	30
8.9.2.3	num_instances	30
8.10	DMSgetCustomFeatureV2 Struct Reference	30
8.10.1	Detailed Description	30
8.10.2	Field Documentation	31
8.10.2.1	pCustSettingInfo	31
8.10.2.2	pCustSettingList	31
8.10.2.3	pGetCustomInput	31
8.11	DMSgetCustomInput Struct Reference	31
8.11.1	Detailed Description	31
8.11.2	Field Documentation	31
8.11.2.1	cust_id	31
8.11.2.2	list_type	31
8.12	dunchannelRate Struct Reference	31
8.12.1	Detailed Description	32
8.12.2	Field Documentation	32
8.12.2.1	CurrChanRxRate	32
8.12.2.2	CurrChanTxRate	32
8.12.2.3	MaxChanRxRate	32

8.12.2.4	MaxChanTxRate	32
8.13	eTWSPLMNInfoTlv Struct Reference	32
8.13.1	Detailed Description	32
8.13.2	Field Documentation	32
8.13.2.1	ETWSPLMNInfo	32
8.13.2.2	TlvPresent	32
8.14	FMSImageElement Struct Reference	32
8.14.1	Detailed Description	33
8.14.2	Field Documentation	33
8.14.2.1	buildId	33
8.14.2.2	buildIdLength	33
8.14.2.3	imageId	33
8.14.2.4	imageType	33
8.15	FMSImageIdElement Struct Reference	33
8.15.1	Detailed Description	33
8.15.2	Field Documentation	34
8.15.2.1	buildID	34
8.15.2.2	buildIDLength	34
8.15.2.3	failureCount	34
8.15.2.4	imageID	34
8.15.2.5	storageIndex	34
8.16	FMSImageIDEntries Struct Reference	34
8.16.1	Detailed Description	34
8.16.2	Field Documentation	35
8.16.2.1	executingImage	35
8.16.2.2	imageIDElement	35
8.16.2.3	imageIDSize	35
8.16.2.4	imageType	35
8.16.2.5	maxImages	35
8.17	FMSImageList Struct Reference	35
8.17.1	Detailed Description	35
8.17.2	Field Documentation	36
8.17.2.1	imageIDEntries	36
8.17.2.2	listSize	36
8.18	FMSPrefImageList Struct Reference	36
8.18.1	Detailed Description	36
8.18.2	Field Documentation	36
8.18.2.1	listEntries	36
8.18.2.2	listSize	36
8.19	hdrSSInfo Struct Reference	36

8.19.1 Detailed Description	36
8.19.2 Field Documentation	37
8.19.2.1 ecio	37
8.19.2.2 io	37
8.19.2.3 rssi	37
8.19.2.4 sinr	37
8.20 image_info_t Struct Reference	37
8.20.1 Field Documentation	37
8.20.1.1 buildID	37
8.20.1.2 buildIDLen	37
8.20.1.3 imageType	37
8.20.1.4 uniqueID	37
8.21 ipv6AddressInfo Struct Reference	37
8.21.1 Detailed Description	37
8.21.2 Field Documentation	38
8.21.2.1 IPAddressV6	38
8.21.2.2 IPV6PrefixLen	38
8.22 LibPackGPRSRequestedQoS Struct Reference	38
8.22.1 Detailed Description	38
8.22.2 Field Documentation	38
8.22.2.1 delayClass	38
8.22.2.2 meanThroughputClass	38
8.22.2.3 peakThroughputClass	38
8.22.2.4 precedenceClass	38
8.22.2.5 reliabilityClass	38
8.23 LibpackProfile3GPP Struct Reference	38
8.23.1 Detailed Description	39
8.23.2 Field Documentation	43
8.23.2.1 pAddrAllocPref	43
8.23.2.2 pAPNClass	43
8.23.2.3 pAPNDisabledFlag	43
8.23.2.4 pAPNName	44
8.23.2.5 pAPNnameSize	44
8.23.2.6 pAuthenticationPref	44
8.23.2.7 pGPRSMinimumQoS	44
8.23.2.8 pGPRSRequestedQos	44
8.23.2.9 pImCnFlag	44
8.23.2.10 pIPv4AddrPref	44
8.23.2.11 pIPv6AddPref	44
8.23.2.12 pPassword	44

8.23.2.13	pPasswordSize	44
8.23.2.14	pPcscfAddrUsingDhcp	44
8.23.2.15	pPcscfAddrUsingPCO	44
8.23.2.16	pPDNInactivTimeout	44
8.23.2.17	pPdpAccessConFlag	44
8.23.2.18	pPdpContext	44
8.23.2.19	pPdpDataCompType	44
8.23.2.20	pPdpHdrCompType	44
8.23.2.21	pPDPTYPE	44
8.23.2.22	pPriDNSIPv4AddPref	44
8.23.2.23	pPriDNSIPv6addpref	44
8.23.2.24	pPrimaryID	44
8.23.2.25	pProfilename	44
8.23.2.26	pProfilenameSize	44
8.23.2.27	pQosClassID	44
8.23.2.28	pSecDNSIPv4AddPref	44
8.23.2.29	pSecDNSIPv6addpref	44
8.23.2.30	pSecondaryFlag	44
8.23.2.31	pTFTID1Params	44
8.23.2.32	pTFTID2Params	45
8.23.2.33	pUMTSMInQoS	45
8.23.2.34	pUMTSMInQoSsigInd	45
8.23.2.35	pUMTSReqQoS	45
8.23.2.36	pUMTSReqQoSSigInd	45
8.23.2.37	pUsername	45
8.23.2.38	pUsernameSize	45
8.24	LibpackProfile3GPP2 Struct Reference	45
8.24.1	Detailed Description	45
8.24.2	Field Documentation	50
8.24.2.1	pAllowLinger	50
8.24.2.2	pAPNClass3GPP2	50
8.24.2.3	pAPNEnabled3GPP2	50
8.24.2.4	pApnString	50
8.24.2.5	pApnStringSize	50
8.24.2.6	pAppPriority	50
8.24.2.7	pAppType	50
8.24.2.8	pAuthPassword	50
8.24.2.9	pAuthPasswordSize	50
8.24.2.10	pAuthProtocol	50
8.24.2.11	pAuthRetryCount	50

8.24.2.12 pAuthTimeout	50
8.24.2.13 pDataMode	50
8.24.2.14 pDataRate	50
8.24.2.15 plpcpAckTimeout	50
8.24.2.16 plpcpCreqRetryCount	50
8.24.2.17 plsPcscfAddressNedded	50
8.24.2.18 pLcpAckTimeout	50
8.24.2.19 pLcpCreqRetryCount	50
8.24.2.20 pNegoDnsSrvrPref	50
8.24.2.21 pPDNInactivTimeout3GPP2	50
8.24.2.22 pPdnType	50
8.24.2.23 pPppSessCloseTimer1x	51
8.24.2.24 pPppSessCloseTimerDO	51
8.24.2.25 pPrimaryV4DnsAddress	51
8.24.2.26 pPriV6DnsAddress	51
8.24.2.27 pRATType	51
8.24.2.28 pSecondaryV4DnsAddress	51
8.24.2.29 pSecV6DnsAddress	51
8.24.2.30 pUserId	51
8.24.2.31 pUserIdSize	51
8.25 LibPackprofile_3GPP Struct Reference	51
8.25.1 Detailed Description	52
8.25.2 Field Documentation	56
8.25.2.1 pAddrAllocPref	56
8.25.2.2 pAPNClass	56
8.25.2.3 pAPNDisabledFlag	56
8.25.2.4 pAPNName	56
8.25.2.5 pAPNnameSize	56
8.25.2.6 pAuthenticationPref	56
8.25.2.7 pGPRSMinimumQoS	56
8.25.2.8 pGPRSRequestedQos	56
8.25.2.9 plmCnFlag	56
8.25.2.10 pIPv4AddrPref	56
8.25.2.11 pIPv6AddPref	56
8.25.2.12 pPassword	56
8.25.2.13 pPasswordSize	56
8.25.2.14 pPcscfAddrUsingDhcp	56
8.25.2.15 pPcscfAddrUsingPCO	56
8.25.2.16 pPDNInactivTimeout	56
8.25.2.17 pPdpAccessConFlag	56

8.25.2.18 pPdpContext	57
8.25.2.19 pPdpDataCompType	57
8.25.2.20 pPdpHdrCompType	57
8.25.2.21 pPDPtype	57
8.25.2.22 pPriDNSIPv4AddPref	57
8.25.2.23 pPriDNSIPv6addpref	57
8.25.2.24 pPrimaryID	57
8.25.2.25 pProfilename	57
8.25.2.26 pProfilenameSize	57
8.25.2.27 pQosClassID	57
8.25.2.28 pSecDNSIPv4AddPref	57
8.25.2.29 pSecDNSIPv6addpref	57
8.25.2.30 pSecondaryFlag	57
8.25.2.31 pTFTID1Params	57
8.25.2.32 pTFTID2Params	57
8.25.2.33 pUMTSMinQoS	57
8.25.2.34 pUMTSMinQoSSigInd	57
8.25.2.35 pUMTSReqQoS	57
8.25.2.36 pUMTSReqQoSSigInd	57
8.25.2.37 pUsername	57
8.25.2.38 pUsernameSize	57
8.26 LibPackprofile_3GPP2 Struct Reference	57
8.26.1 Detailed Description	58
8.26.2 Field Documentation	62
8.26.2.1 pAllowLinger	62
8.26.2.2 pAPNClass3GPP2	62
8.26.2.3 pAPNEnabled3GPP2	62
8.26.2.4 pApnString	62
8.26.2.5 pApnStringSize	62
8.26.2.6 pAppPriority	62
8.26.2.7 pAppType	62
8.26.2.8 pAuthPassword	62
8.26.2.9 pAuthPassword_tSize	62
8.26.2.10 pAuthProtocol	62
8.26.2.11 pAuthRetryCount	62
8.26.2.12 pAuthTimeout	63
8.26.2.13 pDataMode	63
8.26.2.14 pDataRate	63
8.26.2.15 plpcpAckTimeout	63
8.26.2.16 plpcpCreqRetryCount	63

8.26.2.17	plsPcscfAddressNedded	63
8.26.2.18	pLcpAckTimeout	63
8.26.2.19	pLcpCreqRetryCount	63
8.26.2.20	pNegoDnsSrvrPref	63
8.26.2.21	pPDNInactivTimeout3GPP2	63
8.26.2.22	pPdnType	63
8.26.2.23	pPppSessCloseTimer1x	63
8.26.2.24	pPppSessCloseTimerDO	63
8.26.2.25	pPrimaryV4DnsAddress	63
8.26.2.26	pPriV6DnsAddress	63
8.26.2.27	pRATType	63
8.26.2.28	pSecondaryV4DnsAddress	63
8.26.2.29	pSecV6DnsAddress	63
8.26.2.30	pUserId	63
8.26.2.31	pUserIdSize	63
8.27	LibPackQosClassID Struct Reference	63
8.27.1	Detailed Description	63
8.27.2	Field Documentation	64
8.27.2.1	gDIBitRate	64
8.27.2.2	gUIBitRate	64
8.27.2.3	maxDIBitRate	64
8.27.2.4	maxUIBitRate	64
8.27.2.5	QCI	64
8.28	LibPackTFTIDParams Struct Reference	64
8.28.1	Detailed Description	64
8.28.2	Field Documentation	66
8.28.2.1	destPortRangeEnd	66
8.28.2.2	destPortRangeStart	66
8.28.2.3	eValid	66
8.28.2.4	filterId	66
8.28.2.5	flowLabel	66
8.28.2.6	IPSECSPi	66
8.28.2.7	ipVersion	66
8.28.2.8	nextHeader	66
8.28.2.9	pSourceIP	66
8.28.2.10	sourceIPMask	66
8.28.2.11	srcPortRangeEnd	66
8.28.2.12	srcPortRangeStart	66
8.28.2.13	tosMask	66
8.29	LibPackUMTSQoS Struct Reference	66

8.29.1 Detailed Description	66
8.29.2 Field Documentation	69
8.29.2.1 deliveryErrSDU	69
8.29.2.2 grntDownlinkBitrate	69
8.29.2.3 grntUplinkBitrate	69
8.29.2.4 maxDownlinkBitrate	69
8.29.2.5 maxSDUSize	69
8.29.2.6 maxUplinkBitrate	69
8.29.2.7 qosDeliveryOrder	69
8.29.2.8 resBerRatio	69
8.29.2.9 sduErrorRatio	69
8.29.2.10 trafficClass	69
8.29.2.11 trafficPriority	69
8.29.2.12 transferDelay	69
8.30 LibPackUMTSReqQoSsigInd Struct Reference	69
8.30.1 Detailed Description	69
8.30.2 Field Documentation	70
8.30.2.1 SigInd	70
8.30.2.2 UMTSReqQoS	70
8.31 loc_BdsSV Struct Reference	70
8.31.1 Detailed Description	70
8.31.2 Field Documentation	70
8.31.2.1 id	70
8.31.2.2 mask	70
8.32 loc_BdsSVInfo Struct Reference	70
8.32.1 Detailed Description	71
8.32.2 Field Documentation	71
8.32.2.1 len	71
8.32.2.2 pSV	71
8.33 loc_CellDb Struct Reference	71
8.33.1 Detailed Description	71
8.33.2 Field Documentation	72
8.33.2.1 mask	72
8.34 loc_ClkInfo Struct Reference	72
8.34.1 Detailed Description	72
8.34.2 Field Documentation	73
8.34.2.1 mask	74
8.35 loc_GnssData Struct Reference	74
8.35.1 Detailed Description	74
8.35.2 Field Documentation	76

8.35.2.1	mask	76
8.36	loc_gpsTime Struct Reference	76
8.36.1	Detailed Description	76
8.36.2	Field Documentation	76
8.36.2.1	gpsTimeOfWeekMs	76
8.36.2.2	gpsWeek	76
8.37	loc_LocApplicationInfo Struct Reference	76
8.37.1	Detailed Description	76
8.37.2	Field Documentation	77
8.37.2.1	appNameLength	77
8.37.2.2	appProviderLength	77
8.37.2.3	appVersionLength	77
8.37.2.4	appVersionValid	77
8.37.2.5	pAppName	77
8.37.2.6	pAppProvider	77
8.37.2.7	pAppVersion	77
8.38	loc_precisionDilution Struct Reference	77
8.38.1	Detailed Description	78
8.38.2	Field Documentation	78
8.38.2.1	HDOP	78
8.38.2.2	PDOP	78
8.38.2.3	VDOP	78
8.39	loc_sensorDataUsage Struct Reference	78
8.39.1	Detailed Description	78
8.39.2	Field Documentation	79
8.39.2.1	aidingIndicatorMask	79
8.39.2.2	usageMask	79
8.40	loc_SV Struct Reference	79
8.40.1	Detailed Description	79
8.40.2	Field Documentation	80
8.40.2.1	id	80
8.40.2.2	mask	80
8.40.2.3	system	80
8.41	loc_SVInfo Struct Reference	80
8.41.1	Detailed Description	80
8.41.2	Field Documentation	81
8.41.2.1	len	81
8.41.2.2	pSV	81
8.42	loc_svUsedforFix Struct Reference	81
8.42.1	Detailed Description	81

8.42.2	Field Documentation	81
8.42.2.1	gnssSvUsedList	81
8.42.2.2	gnssSvUsedList_len	81
8.43	IteSSInfo Struct Reference	81
8.43.1	Detailed Description	82
8.43.2	Field Documentation	82
8.43.2.1	rsrp	82
8.43.2.2	rsrq	82
8.43.2.3	rssi	82
8.43.2.4	snr	82
8.44	messageModeTlv Struct Reference	82
8.44.1	Detailed Description	82
8.44.2	Field Documentation	82
8.44.2.1	MessageModelInfo	82
8.44.2.2	TlvPresent	82
8.45	nas_acqOrderPref Struct Reference	82
8.45.1	Detailed Description	83
8.45.2	Field Documentation	83
8.45.2.1	acqOrdeLen	83
8.45.2.2	pAcqOrder	83
8.46	nas_AddCDMASysInfo Struct Reference	83
8.46.1	Detailed Description	83
8.46.2	Field Documentation	84
8.46.2.1	geoSysIdx	84
8.46.2.2	regPrd	84
8.47	nas_AddSysInfo Struct Reference	84
8.47.1	Detailed Description	84
8.47.2	Field Documentation	84
8.47.2.1	cellBroadcastCap	84
8.47.2.2	geoSysIdx	84
8.48	nas_CallBarringSysInfo Struct Reference	84
8.48.1	Detailed Description	84
8.48.2	Field Documentation	85
8.48.2.1	csBarStatus	85
8.48.2.2	psBarStatus	85
8.49	nas_callBarStatus Struct Reference	85
8.49.1	Detailed Description	85
8.49.2	Field Documentation	86
8.49.2.1	csBarStatus	86
8.49.2.2	psBarStatus	86

8.50 nas_CDMAECIOThresh Struct Reference	86
8.50.1 Detailed Description	86
8.50.2 Field Documentation	87
8.50.2.1 CDMAECIOThreshListLen	87
8.50.2.2 pCDMAECIOThreshList	87
8.51 nas_CDMAInfo Struct Reference	87
8.51.1 Detailed Description	87
8.51.2 Field Documentation	88
8.51.2.1 baseId	88
8.51.2.2 baseLat	88
8.51.2.3 baseLong	88
8.51.2.4 nid	88
8.51.2.5 refpn	88
8.51.2.6 sid	88
8.52 nas_CDMARSSIThresh Struct Reference	88
8.52.1 Detailed Description	89
8.52.2 Field Documentation	89
8.52.2.1 CDMARSSIThreshListLen	89
8.52.2.2 pCDMARSSIThreshList	89
8.53 nas_CDMASysInfo Struct Reference	89
8.53.1 Detailed Description	89
8.53.2 Field Documentation	93
8.53.2.1 baseId	93
8.53.2.2 baseLat	93
8.53.2.3 baseLong	93
8.53.2.4 bsInfoValid	93
8.53.2.5 bsPRev	93
8.53.2.6 bsPRevValid	93
8.53.2.7 ccsSupported	93
8.53.2.8 ccsSupportedValid	93
8.53.2.9 cdmaSysIdValid	93
8.53.2.10 isSysPrIMatch	93
8.53.2.11 isSysPrIMatchValid	93
8.53.2.12 MCC	93
8.53.2.13 MNC	93
8.53.2.14 networkId	93
8.53.2.15 networkIdValid	93
8.53.2.16 packetZone	93
8.53.2.17 packetZoneValid	93
8.53.2.18 pRevInUse	93

8.53.2.19 pRevInUseValid	93
8.53.2.20 sysInfoCDMA	93
8.53.2.21 systemID	93
8.54 nas_CDMASysInfoExt Struct Reference	93
8.54.1 Detailed Description	94
8.54.2 Field Documentation	94
8.54.2.1 imsi_11_12	94
8.54.2.2 MCC	94
8.55 nas_cellParams Struct Reference	94
8.55.1 Detailed Description	94
8.55.2 Field Documentation	95
8.55.2.1 pci	95
8.55.2.2 rsrp	95
8.55.2.3 rsrq	95
8.55.2.4 rssi	95
8.55.2.5 srxlev	95
8.56 nas_CommInfo Struct Reference	95
8.56.1 Detailed Description	95
8.56.2 Field Documentation	97
8.56.2.1 imsRegState	97
8.56.2.2 modemMode	97
8.56.2.3 psState	97
8.56.2.4 systemMode	97
8.56.2.5 temperature	97
8.57 nas_CSGID Struct Reference	97
8.57.1 Detailed Description	98
8.57.2 Field Documentation	98
8.57.2.1 id	98
8.57.2.2 mcc	98
8.57.2.3 mnc	98
8.57.2.4 mncPcsDigits	98
8.57.2.5 rat	98
8.58 nas_currentPLMN Struct Reference	98
8.58.1 Detailed Description	99
8.58.2 Field Documentation	99
8.58.2.1 MCC	99
8.58.2.2 MNC	99
8.58.2.3 netDescr	99
8.58.2.4 netDescrLength	99
8.59 nas_dataSrvCapabilities Struct Reference	99

8.59.1 Detailed Description	100
8.59.2 Field Documentation	100
8.59.2.1 dataCapabilities	100
8.59.2.2 dataCapabilitiesLen	100
8.60 nas_detailSvcInfo Struct Reference	100
8.60.1 Detailed Description	101
8.60.2 Field Documentation	102
8.60.2.1 hdrHybrid	102
8.60.2.2 hdrSrvStatus	102
8.60.2.3 isSysForbidden	102
8.60.2.4 srvCapability	102
8.60.2.5 srvStatus	102
8.61 nas_ecioListElement Struct Reference	102
8.61.1 Detailed Description	102
8.61.2 Field Documentation	103
8.61.2.1 ecio	103
8.61.2.2 radiolf	103
8.62 nas_errorRateListElement Struct Reference	103
8.62.1 Detailed Description	103
8.62.2 Field Documentation	104
8.62.2.1 errorRate	104
8.62.2.2 radiolf	104
8.63 nas_GERANInfo Struct Reference	104
8.63.1 Detailed Description	105
8.63.2 Field Documentation	106
8.63.2.1 arfcn	106
8.63.2.2 bsic	106
8.63.2.3 cellID	106
8.63.2.4 insNmrCellInfo	106
8.63.2.5 lac	106
8.63.2.6 nmrlnst	106
8.63.2.7 plmn	106
8.63.2.8 rxLev	106
8.63.2.9 timingAdvance	106
8.64 nas_geranInstInfo Struct Reference	106
8.64.1 Detailed Description	107
8.64.2 Field Documentation	107
8.64.2.1 geranArfcn	107
8.64.2.2 geranBsicBcc	107
8.64.2.3 geranBsicNcc	107

8.64.2.4	geranRssi	107
8.65	nas_gsmCellInfo Struct Reference	107
8.65.1	Detailed Description	107
8.65.2	Field Documentation	108
8.65.2.1	arfcn	108
8.65.2.2	band1900	108
8.65.2.3	bsicld	108
8.65.2.4	cellIdValid	108
8.65.2.5	rssI	108
8.65.2.6	srxlev	108
8.66	nas_GSMRSSIthresh Struct Reference	108
8.66.1	Detailed Description	109
8.66.2	Field Documentation	109
8.66.2.1	GSMRSSIthreshListLen	109
8.66.2.2	pGSMRSSIthreshList	109
8.67	nas_GSMSrvStatusInfo Struct Reference	109
8.67.1	Detailed Description	109
8.67.2	Field Documentation	110
8.67.2.1	isPrefDataPath	110
8.67.2.2	srvStatus	110
8.67.2.3	trueSrvStatus	110
8.68	nas_GSMSysInfo Struct Reference	110
8.68.1	Detailed Description	110
8.68.2	Field Documentation	113
8.68.2.1	cellId	113
8.68.2.2	cellIdValid	113
8.68.2.3	dtmSupp	113
8.68.2.4	dtmSuppValid	113
8.68.2.5	egprsSupp	113
8.68.2.6	egprsSuppValid	113
8.68.2.7	lac	113
8.68.2.8	lacValid	113
8.68.2.9	MCC	113
8.68.2.10	MNC	113
8.68.2.11	networkIdValid	113
8.68.2.12	regRejectInfoValid	113
8.68.2.13	rejCause	113
8.68.2.14	rejectSrvDomain	113
8.68.2.15	sysInfoGSM	113
8.69	nas_HDRECIOthresh Struct Reference	113

8.69.1 Detailed Description	113
8.69.2 Field Documentation	114
8.69.2.1 HDRECIOThreshListLen	114
8.69.2.2 pHRECIOThreshList	114
8.70 nas_HDRIOThresh Struct Reference	114
8.70.1 Detailed Description	114
8.70.2 Field Documentation	114
8.70.2.1 HDRIOThreshListLen	114
8.70.2.2 pHDRIOThreshList	114
8.71 nas_HDRRSSIThresh Struct Reference	114
8.71.1 Detailed Description	115
8.71.2 Field Documentation	115
8.71.2.1 HDRRSSIThreshListLen	115
8.71.2.2 pHDRRSSIThreshList	115
8.72 nas_HDRSINRThreshold Struct Reference	115
8.72.1 Detailed Description	115
8.72.2 Field Documentation	116
8.72.2.1 HDRSINRThreshListLen	116
8.72.2.2 pHDRSINRThreshList	116
8.73 nas_HDRSysInfo Struct Reference	116
8.73.1 Detailed Description	116
8.73.2 Field Documentation	118
8.73.2.1 hdrActiveProt	118
8.73.2.2 hdrActiveProtValid	118
8.73.2.3 hdrPersonality	118
8.73.2.4 hdrPersonalityValid	118
8.73.2.5 is856SysId	118
8.73.2.6 is856SysIdValid	118
8.73.2.7 isSysPrIMatch	118
8.73.2.8 isSysPrIMatchValid	118
8.73.2.9 sysInfoHDR	118
8.74 nas_infoInterFreq Struct Reference	119
8.74.1 Detailed Description	119
8.74.2 Field Documentation	119
8.74.2.1 cell_resel_priority	119
8.74.2.2 cellInterFreqParams	120
8.74.2.3 cells_len	120
8.74.2.4 earfcn	120
8.74.2.5 threshXHigh	120
8.74.2.6 threshXLow	120

8.75	nas_lteGsmCellInfo Struct Reference	120
8.75.1	Detailed Description	120
8.75.2	Field Documentation	121
8.75.2.1	cellReselPriority	121
8.75.2.2	cells_len	121
8.75.2.3	GsmCellInfo	121
8.75.2.4	nccPermitted	121
8.75.2.5	threshGsmHigh	121
8.75.2.6	threshGsmLow	121
8.76	nas_LTEInfo Struct Reference	121
8.76.1	Detailed Description	121
8.76.2	Field Documentation	123
8.76.2.1	band	123
8.76.2.2	bandwidth	123
8.76.2.3	emmConnState	123
8.76.2.4	emmState	124
8.76.2.5	emmSubState	124
8.76.2.6	RXChan	124
8.76.2.7	TXChan	124
8.77	nas_LTEInfoInterfreq Struct Reference	124
8.77.1	Detailed Description	124
8.77.2	Field Documentation	124
8.77.2.1	freqsLen	124
8.77.2.2	InfoInterfreq	124
8.77.2.3	ueInIdle	124
8.78	nas_LTEInfoIntrafreq Struct Reference	124
8.78.1	Detailed Description	125
8.78.2	Field Documentation	126
8.78.2.1	CellParams	126
8.78.2.2	cellReselPriority	126
8.78.2.3	cellsLen	126
8.78.2.4	earfcn	127
8.78.2.5	globalCellId	127
8.78.2.6	plmn	127
8.78.2.7	servingCellId	127
8.78.2.8	sIntraSearch	127
8.78.2.9	sNonIntraSearch	127
8.78.2.10	tac	127
8.78.2.11	threshServingLow	127
8.78.2.12	ueInIdle	127

8.79	nas_LTEInfoNeighboringGSM Struct Reference	127
8.79.1	Detailed Description	127
8.79.2	Field Documentation	127
8.79.2.1	freqsLen	127
8.79.2.2	LteGsmCellInfo	127
8.79.2.3	ueInIdle	127
8.80	nas_LTEInfoNeighboringWCDMA Struct Reference	128
8.80.1	Detailed Description	128
8.80.2	Field Documentation	128
8.80.2.1	freqsLen	128
8.80.2.2	LTEWCDMACellInfo	128
8.80.2.3	ueInIdle	128
8.81	nas_lteRsrpInformation Struct Reference	128
8.81.1	Detailed Description	128
8.81.2	Field Documentation	129
8.81.2.1	rsrpLevel	129
8.82	nas_LTERSRPThresh Struct Reference	129
8.82.1	Detailed Description	129
8.82.2	Field Documentation	129
8.82.2.1	LTERSRPThreshListLen	129
8.82.2.2	pLTERSRPThreshList	129
8.83	nas_LTERSRQThresh Struct Reference	129
8.83.1	Detailed Description	129
8.83.2	Field Documentation	130
8.83.2.1	LTERSRQThreshListLen	130
8.83.2.2	pLTERSRQThreshList	130
8.84	nas_LTERSSIThresh Struct Reference	130
8.84.1	Detailed Description	130
8.84.2	Field Documentation	130
8.84.2.1	LTERSSIThreshListLen	130
8.84.2.2	pLTERSSIThreshList	130
8.85	nas_LTESigRptConfig Struct Reference	130
8.85.1	Detailed Description	131
8.85.2	Field Documentation	132
8.85.2.1	avgPeriod	132
8.85.2.2	rptRate	132
8.86	nas_lteSnrInformation Struct Reference	132
8.86.1	Detailed Description	132
8.86.2	Field Documentation	132
8.86.2.1	snrLevel	132

8.87 nas_LTESNRThreshold Struct Reference	132
8.87.1 Detailed Description	132
8.87.2 Field Documentation	133
8.87.2.1 LTESNRThreshListLen	133
8.87.2.2 pLTESNRThreshList	133
8.88 nas_LTESysInfo Struct Reference	133
8.88.1 Detailed Description	133
8.88.2 Field Documentation	135
8.88.2.1 cellId	135
8.88.2.2 cellIdValid	135
8.88.2.3 lac	135
8.88.2.4 lacValid	135
8.88.2.5 MCC	135
8.88.2.6 MNC	135
8.88.2.7 networkIdValid	135
8.88.2.8 regRejectInfoValid	135
8.88.2.9 rejCause	135
8.88.2.10 rejectSrvDomain	135
8.88.2.11 sysInfoLTE	135
8.88.2.12 tac	136
8.88.2.13 tacValid	136
8.89 nas_lteWcdmaCellInfo Struct Reference	136
8.89.1 Detailed Description	136
8.89.2 Field Documentation	137
8.89.2.1 cellReselPriority	137
8.89.2.2 cellsLen	137
8.89.2.3 threshXhigh	137
8.89.2.4 threshXlow	137
8.89.2.5 uarfcn	137
8.89.2.6 WCDMACellInfo	137
8.90 nas_MNRInfo Struct Reference	137
8.90.1 Detailed Description	137
8.90.2 Field Documentation	138
8.90.2.1 mcc	138
8.90.2.2 mnc	138
8.90.2.3 rat	138
8.91 nas_netSelectionPref Struct Reference	138
8.91.1 Detailed Description	138
8.91.2 Field Documentation	138
8.91.2.1 mcc	138

8.91.2.2	mnc	138
8.91.2.3	netReg	138
8.92	nas_nmrCellInfo Struct Reference	138
8.92.1	Detailed Description	139
8.92.2	Field Documentation	140
8.92.2.1	nmrArfcn	140
8.92.2.2	nmrBsic	140
8.92.2.3	nmrCellID	140
8.92.2.4	nmrLac	141
8.92.2.5	nmrPlmn	141
8.92.2.6	nmrRxLev	141
8.93	nas_qaQmi3Gpp2TimeZone Struct Reference	141
8.93.1	Detailed Description	141
8.93.2	Field Documentation	141
8.93.2.1	daylightSavings	141
8.93.2.2	leapSeconds	141
8.93.2.3	localTimeOffset	141
8.94	nas_QmiNas3GppNetworkInfo Struct Reference	141
8.94.1	Detailed Description	142
8.94.2	Field Documentation	142
8.94.2.1	Desription	142
8.94.2.2	Forbidden	142
8.94.2.3	InUse	142
8.94.2.4	MCC	142
8.94.2.5	MNC	142
8.94.2.6	Preferred	142
8.94.2.7	Roaming	142
8.95	nas_QmiNas3GppNetworkRAT Struct Reference	142
8.95.1	Detailed Description	142
8.95.2	Field Documentation	143
8.95.2.1	MCC	143
8.95.2.2	MNC	143
8.95.2.3	RAT	143
8.96	nas_QmisNasPcsDigit Struct Reference	143
8.96.1	Detailed Description	143
8.96.2	Field Documentation	144
8.96.2.1	includes_pcs_digit	144
8.96.2.2	MCC	144
8.96.2.3	MNC	144
8.97	nas_RejectReasonTlv Struct Reference	144

8.97.1 Detailed Description	144
8.97.2 Field Documentation	144
8.97.2.1 rejectCause	144
8.97.2.2 serviceDomain	144
8.97.2.3 TlvPresent	144
8.98 nas_RFInfoTlv Struct Reference	144
8.98.1 Detailed Description	144
8.98.2 Field Documentation	145
8.98.2.1 activeBandClass	145
8.98.2.2 activeChannel	145
8.98.2.3 radiolInterface	145
8.98.2.4 radiolInterfaceSize	145
8.98.2.5 TlvPresent	145
8.99 nas_roamIndList Struct Reference	145
8.99.1 Detailed Description	145
8.99.2 Field Documentation	146
8.99.2.1 numInstances	146
8.99.2.2 radiolInterface	146
8.99.2.3 roamIndicator	146
8.100 nas_rsrqInformation Struct Reference	146
8.100.1 Detailed Description	146
8.100.2 Field Documentation	146
8.100.2.1 radiolf	146
8.100.2.2 rsrq	146
8.101 nas_rxSignalStrengthListElement Struct Reference	146
8.101.1 Detailed Description	146
8.101.2 Field Documentation	147
8.101.2.1 radiolf	147
8.101.2.2 rxSignalStrength	147
8.102 nas_servSystem Struct Reference	147
8.102.1 Detailed Description	147
8.102.2 Field Documentation	149
8.102.2.1 csAttachState	149
8.102.2.2 numRadiolInterfaces	149
8.102.2.3 psAttachState	149
8.102.2.4 radiolInterface	149
8.102.2.5 regState	149
8.102.2.6 selNetwork	149
8.103 nas_SignalStrengthTlv Struct Reference	149
8.103.1 Detailed Description	149

8.103.2 Field Documentation	150
8.103.2.1 radiolInterface	150
8.103.2.2 signalStrength	150
8.103.2.3 TlvPresent	150
8.104nas_SLQSSignalStrengthsIndReq Struct Reference	150
8.104.1 Detailed Description	150
8.104.2 Field Documentation	150
8.104.2.1 ecioDelta	150
8.104.2.2 ecioThresholdList	150
8.104.2.3 ecioThresholdListLen	150
8.104.2.4 ioDelta	150
8.104.2.5 lteRsrpDelta	151
8.104.2.6 lteSnrDelta	151
8.104.2.7 rsrqDelta	151
8.104.2.8 rxSignalStrengthDelta	151
8.104.2.9 sinrDelta	151
8.104.2.10sinrThresholdList	151
8.104.2.11sinrThresholdListLen	151
8.105nas_SLQSSignalStrengthsInformation Struct Reference	151
8.105.1 Detailed Description	151
8.105.2 Field Documentation	151
8.105.2.1 ecioInfo	151
8.105.2.2 errorRateInfo	151
8.105.2.3 io	151
8.105.2.4 lteRsrpinfo	151
8.105.2.5 lteSnrinfo	151
8.105.2.6 rsrqInfo	151
8.105.2.7 rxSignalStrengthInfo	152
8.105.2.8 sinr	152
8.106nas_SLQSSignalStrengthsTlv Struct Reference	152
8.106.1 Detailed Description	152
8.106.2 Field Documentation	152
8.106.2.1 sSLQSSignalStrengthsInfo	152
8.106.2.2 TlvPresent	152
8.107nas_SrvStatusInfo Struct Reference	152
8.107.1 Detailed Description	152
8.107.2 Field Documentation	153
8.107.2.1 isPrefDataPath	153
8.107.2.2 srvStatus	153
8.108nas_sysInfoCommon Struct Reference	153

8.108.1 Detailed Description	153
8.108.2 Field Documentation	156
8.108.2.1 isSysForbidden	156
8.108.2.2 isSysForbiddenValid	156
8.108.2.3 roamStatus	156
8.108.2.4 roamStatusValid	156
8.108.2.5 srvCapability	156
8.108.2.6 srvCapabilityValid	156
8.108.2.7 srvDomain	156
8.108.2.8 srvDomainValid	156
8.109nas_TDSCDMAECIOThresh Struct Reference	156
8.109.1 Detailed Description	156
8.109.2 Field Documentation	157
8.109.2.1 pTDSCDMAECIOThreshList	157
8.109.2.2 TDSCDMAECIOThreshListLen	157
8.110nas_TDSCDMARSCPThresh Struct Reference	157
8.110.1 Detailed Description	157
8.110.2 Field Documentation	157
8.110.2.1 pTDSCDMARSCPThreshList	157
8.110.2.2 TDSCDMARSCPThreshListLen	157
8.111nas_TDSCDMARSSIThresh Struct Reference	157
8.111.1 Detailed Description	157
8.111.2 Field Documentation	158
8.111.2.1 pTDSCDMARSSIThreshList	158
8.111.2.2 TDSCDMARSSIThreshListLen	158
8.112nas_TDSCDMASINRThresh Struct Reference	158
8.112.1 Detailed Description	158
8.112.2 Field Documentation	158
8.112.2.1 pTDSCDMASINRThreshList	158
8.112.2.2 TDSCDMASINRThreshListLen	158
8.113nas_UMTSInfo Struct Reference	158
8.113.1 Detailed Description	159
8.113.2 Field Documentation	160
8.113.2.1 cellID	160
8.113.2.2 ecio	160
8.113.2.3 geranInst	160
8.113.2.4 GeranInstInfo	160
8.113.2.5 lac	160
8.113.2.6 plmn	160
8.113.2.7 psc	160

8.113.2.8 rscp	160
8.113.2.9 uarfcn	160
8.113.2.10 umtsInst	160
8.113.2.11 UMTSInstInfo	160
8.114 nas_UMTSInstInfo Struct Reference	160
8.114.1 Detailed Description	161
8.114.2 Field Documentation	162
8.114.2.1 umtsEcio	162
8.114.2.2 umtsPsc	162
8.114.2.3 umtsRscp	162
8.114.2.4 umtsUarfcn	162
8.115 nas_umtsLTENbrCell Struct Reference	162
8.115.1 Detailed Description	162
8.115.2 Field Documentation	163
8.115.2.1 cellIsTDD	163
8.115.2.2 earfcn	163
8.115.2.3 pci	163
8.115.2.4 rsrp	163
8.115.2.5 rsrq	163
8.115.2.6 srxlev	163
8.116 nas_wcdmaCellInfo Struct Reference	163
8.116.1 Detailed Description	163
8.116.2 Field Documentation	164
8.116.2.1 cpich_ecno	164
8.116.2.2 cpich_rscp	164
8.116.2.3 psc	164
8.116.2.4 srxlev	164
8.117 nas_WCDMAECIOThresh Struct Reference	164
8.117.1 Detailed Description	164
8.117.2 Field Documentation	164
8.117.2.1 pWCDMAECIOThreshList	165
8.117.2.2 WCDMAECIOThreshListLen	165
8.118 nas_WCDMAInfoLTENeighborCell Struct Reference	165
8.118.1 Detailed Description	165
8.118.2 Field Documentation	166
8.118.2.1 UMTSLTENbrCell	166
8.118.2.2 umtsLTENbrCellLen	166
8.118.2.3 wcdmaRRCState	166
8.119 nas_WCDMARSSIThresh Struct Reference	166
8.119.1 Detailed Description	166

8.119.2 Field Documentation	166
8.119.2.1 pWCDMARSSIThreshList	166
8.119.2.2 WCDMARSSIThreshListLen	166
8.120nas_WCDMASysInfo Struct Reference	166
8.120.1 Detailed Description	167
8.120.2 Field Documentation	170
8.120.2.1 cellId	170
8.120.2.2 cellIdValid	170
8.120.2.3 hsCallStatus	170
8.120.2.4 hsCallStatusValid	170
8.120.2.5 hsInd	170
8.120.2.6 hsIndValid	170
8.120.2.7 lac	170
8.120.2.8 lacValid	170
8.120.2.9 MCC	170
8.120.2.10MNC	170
8.120.2.11networkIdValid	170
8.120.2.12psc	170
8.120.2.13pscValid	170
8.120.2.14regRejectInfoValid	170
8.120.2.15rejCause	170
8.120.2.16rejectSrvDomain	170
8.120.2.17sysInfoWCDMA	170
8.121NASBandPreferenceTlv Struct Reference	170
8.121.1 Field Documentation	171
8.121.1.1 band_pref	171
8.121.1.2 TlvPresent	171
8.122NASEmergencyModeTlv Struct Reference	171
8.122.1 Field Documentation	171
8.122.1.1 EmerMode	171
8.122.1.2 TlvPresent	171
8.123NasGetLTECphyCaInfo Struct Reference	171
8.123.1 Field Documentation	171
8.123.1.1 PhyCaAggPcellInfo	171
8.123.1.2 PhyCaAggScellDIBw	171
8.123.1.3 PhyCaAggScellIndex	171
8.123.1.4 PhyCaAggScellIndType	171
8.123.1.5 PhyCaAggScellInfo	171
8.124NASGWAcqOrderPrefTlv Struct Reference	171
8.124.1 Field Documentation	172

8.124.1.1 GWAcqOrderPref	172
8.124.1.2 TlvPresent	172
8.125NASLTEBandPreferenceTlv Struct Reference	172
8.125.1 Field Documentation	172
8.125.1.1 LTEBandPref	172
8.125.1.2 TlvPresent	172
8.126NASLteNasReleaseInfoTlv Struct Reference	172
8.126.1 Field Documentation	172
8.126.1.1 nas_major	172
8.126.1.2 nas_minor	172
8.126.1.3 nas_release	172
8.126.1.4 TlvPresent	172
8.127NASModePreferenceTlv Struct Reference	172
8.127.1 Field Documentation	172
8.127.1.1 ModePref	172
8.127.1.2 TlvPresent	172
8.128NASNetSelPreferenceTlv Struct Reference	173
8.128.1 Field Documentation	173
8.128.1.1 NetSelPref	173
8.128.1.2 TlvPresent	173
8.129NASOTAMessageTlv Struct Reference	173
8.129.1 Field Documentation	173
8.129.1.1 data_buf	173
8.129.1.2 data_len	173
8.129.1.3 message_type	173
8.129.1.4 TlvPresent	173
8.130NASPhyCaAggPcellInfo Struct Reference	173
8.130.1 Detailed Description	173
8.130.2 Field Documentation	174
8.130.2.1 dl_bw_value	174
8.130.2.2 freq	174
8.130.2.3 iLTEbandValue	174
8.130.2.4 pci	174
8.130.2.5 TlvPresent	174
8.131NASPhyCaAggScellIDBw Struct Reference	174
8.131.1 Detailed Description	174
8.131.2 Field Documentation	175
8.131.2.1 dl_bw_value	175
8.131.2.2 TlvPresent	175
8.132NASPhyCaAggScellIndex Struct Reference	175

8.132.1 Detailed Description	175
8.132.2 Field Documentation	175
8.132.2.1 scell_idx	175
8.132.2.2 TlvPresent	175
8.133NASPhyCaAggScellIndType Struct Reference	175
8.133.1 Detailed Description	175
8.133.2 Field Documentation	176
8.133.2.1 freq	176
8.133.2.2 pci	176
8.133.2.3 scell_state	176
8.133.2.4 TlvPresent	176
8.134NASPhyCaAggScellInfo Struct Reference	176
8.134.1 Detailed Description	176
8.134.2 Field Documentation	177
8.134.2.1 dl_bw_value	177
8.134.2.2 freq	177
8.134.2.3 iLTEbandValue	177
8.134.2.4 pci	177
8.134.2.5 scell_state	177
8.134.2.6 TlvPresent	177
8.135NASPRLPreferenceTlv Struct Reference	177
8.135.1 Field Documentation	177
8.135.1.1 PRLPref	177
8.135.1.2 TlvPresent	177
8.136NASQmiCbkNasSwiOTAMessageInd Struct Reference	177
8.136.1 Field Documentation	178
8.136.1.1 nasRelInfoTlv	178
8.136.1.2 otaMsgTlv	178
8.136.1.3 timeTlv	178
8.137NASQmiCbkNasSystemSelPrefInd Struct Reference	178
8.137.1 Field Documentation	178
8.137.1.1 BPTlv	178
8.137.1.2 EMTlv	178
8.137.1.3 GWAOPTlv	178
8.137.1.4 LBPTlv	178
8.137.1.5 MPTlv	178
8.137.1.6 NSPTlv	178
8.137.1.7 PRLPTlv	178
8.137.1.8 RPTlv	178
8.137.1.9 SDPTlv	178

8.138NASRoamPreferenceTlv Struct Reference	178
8.138.1 Field Documentation	179
8.138.1.1 RoamPref	179
8.138.1.2 TlvPresent	179
8.139NASServDomainPrefTlv Struct Reference	179
8.139.1 Field Documentation	179
8.139.1.1 SrvDomainPref	179
8.139.1.2 TlvPresent	179
8.140NASServingSystemInfo Struct Reference	179
8.140.1 Detailed Description	179
8.140.2 Field Documentation	180
8.140.2.1 csAttachState	180
8.140.2.2 hdrPersonality	180
8.140.2.3 psAttachState	180
8.140.2.4 radiolInterfaceList	180
8.140.2.5 radiolInterfaceNo	181
8.140.2.6 registrationState	181
8.140.2.7 selectedNetwork	181
8.141NASTimeInfoTlv Struct Reference	181
8.141.1 Field Documentation	181
8.141.1.1 time	181
8.141.1.2 TlvPresent	181
8.142newMTMessageTlv Struct Reference	181
8.142.1 Detailed Description	181
8.142.2 Field Documentation	181
8.142.2.1 MTMessageInfo	181
8.142.2.2 TlvPresent	181
8.143pack_dms_GetCustFeaturesV2_t Struct Reference	181
8.143.1 Detailed Description	182
8.143.2 Field Documentation	182
8.143.2.1 cust_id	182
8.143.2.2 list_type	182
8.143.2.3 Tlvresult	182
8.144pack_dms_SetCustFeature_t Struct Reference	182
8.144.1 Field Documentation	182
8.144.1.1 DHCPRelayEnabled	182
8.144.1.2 DisableIMSI	182
8.144.1.3 GpsEnable	182
8.144.1.4 GPSSLPM	182
8.144.1.5 GPSSel	182

8.144.1.6 IPFamSupport	182
8.144.1.7 IsVoiceEnabled	183
8.144.1.8 RMAutoConnect	183
8.144.1.9 SMSSupport	183
8.145pack_dms_SetCustFeaturesV2_t Struct Reference	183
8.145.1 Detailed Description	183
8.145.2 Field Documentation	183
8.145.2.1 cust_id	183
8.145.2.2 cust_value	183
8.145.2.3 Tlvresult	183
8.145.2.4 value_length	183
8.146pack_dms_SetEventReport_t Struct Reference	183
8.146.1 Field Documentation	183
8.146.1.1 mode	184
8.147pack_dms_SetPower_t Struct Reference	184
8.147.1 Field Documentation	184
8.147.1.1 mode	184
8.147.1.2 Tlvresult	184
8.148pack_dms_SetUSBComp_t Struct Reference	184
8.148.1 Field Documentation	184
8.148.1.1 Tlvresult	184
8.148.1.2 USBComp	184
8.149pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	184
8.149.1 Detailed Description	184
8.149.2 Field Documentation	184
8.149.2.1 resetInfoInd	184
8.150pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	185
8.150.1 Detailed Description	185
8.150.2 Field Documentation	185
8.150.2.1 pDestSMSContent	185
8.150.2.2 pDestSMSNum	185
8.151pack_dms_UIMGetICCID_t Struct Reference	185
8.151.1 Detailed Description	185
8.151.2 Field Documentation	185
8.151.2.1 Tlvresult	185
8.152pack_fms_GetImagesPreference_t Struct Reference	185
8.152.1 Detailed Description	186
8.152.2 Field Documentation	186
8.152.2.1 Tlvresult	186
8.153pack_fms_GetStoredImages_t Struct Reference	186

8.153.1 Detailed Description	186
8.153.2 Field Documentation	186
8.153.2.1 Tlvresult	186
8.154pack_fms_SetImagesPreference_t Struct Reference	186
8.154.1 Detailed Description	186
8.154.2 Field Documentation	187
8.154.2.1 bForceDownload	187
8.154.2.2 imageListSize	187
8.154.2.3 modemindex	187
8.154.2.4 pImageList	187
8.154.2.5 Tlvresult	187
8.155pack_loc_Delete_Assist_Data_t Struct Reference	187
8.155.1 Detailed Description	187
8.155.2 Field Documentation	188
8.155.2.1 pBdsSVInfo	188
8.155.2.2 pCellDb	188
8.155.2.3 pCikInfo	188
8.155.2.4 pGnssData	188
8.155.2.5 pSVInfo	188
8.155.2.6 Tlvresult	188
8.156pack_loc_EventRegister_t Struct Reference	188
8.156.1 Detailed Description	188
8.156.2 Field Documentation	190
8.156.2.1 eventRegister	190
8.156.2.2 Tlvresult	190
8.157pack_loc_SetExtPowerState_t Struct Reference	190
8.157.1 Detailed Description	190
8.157.2 Field Documentation	190
8.157.2.1 extPowerState	190
8.157.2.2 Tlvresult	190
8.158pack_loc_SetOperationMode_t Struct Reference	190
8.158.1 Detailed Description	191
8.158.2 Field Documentation	191
8.158.2.1 mode	191
8.158.2.2 Tlvresult	191
8.159pack_loc_Start_t Struct Reference	191
8.159.1 Detailed Description	191
8.159.2 Field Documentation	193
8.159.2.1 pApplicationInfo	193
8.159.2.2 pConfigAltitudeAssumed	193

8.159.2.3 pHorizontalAccuracyLvl	193
8.159.2.4 pIntermediateReportState	193
8.159.2.5 pMinIntervalTime	193
8.159.2.6 pRecurrenceType	193
8.159.2.7 SessionId	193
8.159.2.8 Tlvresult	193
8.160pack_loc_Stop_t Struct Reference	193
8.160.1 Detailed Description	193
8.160.2 Field Documentation	193
8.160.2.1 SessionId	193
8.160.2.2 Tlvresult	193
8.161pack_nas_SetACCOLC_t Struct Reference	193
8.161.1 Detailed Description	194
8.161.2 Field Documentation	195
8.161.2.1 accolc	195
8.161.2.2 spc	195
8.162pack_nas_SetNetworkPreference_t Struct Reference	195
8.162.1 Detailed Description	195
8.162.2 Field Documentation	196
8.162.2.1 Duration	196
8.162.2.2 TechnologyPref	196
8.162.2.3 Tlvresult	196
8.163pack_nas_SLQSGetPLMNName_t Struct Reference	196
8.163.1 Detailed Description	196
8.163.2 Field Documentation	197
8.163.2.1 mcc	197
8.163.2.2 mnc	197
8.163.2.3 pMncPcsStatus	197
8.164pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference	197
8.164.1 Detailed Description	197
8.164.2 Field Documentation	198
8.164.2.1 pChangeDuration	198
8.164.2.2 pMncPcsDigitStatus	198
8.164.2.3 pMNRInfo	198
8.164.2.4 regAction	198
8.165pack_nas_SLQSNasConfigSigInfo2_t Struct Reference	198
8.165.1 Detailed Description	200
8.165.2 Field Documentation	202
8.165.2.1 pCDMAECIODelta	202
8.165.2.2 pCDMAECIOThresh	202

8.165.2.3 pCDMARSSIDelta	202
8.165.2.4 pCDMARSSIThresh	202
8.165.2.5 pGSMRSSIDelta	202
8.165.2.6 pGSMRSSIThresh	202
8.165.2.7 pHDRICIODelta	202
8.165.2.8 pHDRICIOThresh	202
8.165.2.9 pHDRIODelta	203
8.165.2.10 pHDRIOThresh	203
8.165.2.11 pHDRRSSIDelta	203
8.165.2.12 pHDRRSSIThresh	203
8.165.2.13 pHDRSINRDelta	203
8.165.2.14 pHDRSINRThresh	203
8.165.2.15 pLTERSRPDelta	203
8.165.2.16 pLTERSRPThresh	203
8.165.2.17 pLTERSRQDelta	203
8.165.2.18 pLTERSRQThresh	203
8.165.2.19 pLTERSSIDelta	203
8.165.2.20 pLTERSSIThresh	203
8.165.2.21 pLTESigRptConfig	203
8.165.2.22 pLTESNRDelta	203
8.165.2.23 pLTESNRThresh	203
8.165.2.24 pTDSCDMAECIODelta	203
8.165.2.25 pTDSCDMAECIOThresh	203
8.165.2.26 pTDSCDMARSCPDelta	203
8.165.2.27 pTDSCDMARSCPThresh	203
8.165.2.28 pTDSCDMARSSIDelta	203
8.165.2.29 pTDSCDMARSSIThresh	203
8.165.2.30 pTDSCDMASINRDelta	203
8.165.2.31 pTDSCDMASINRThresh	203
8.165.2.32 pWCDMAECIODelta	203
8.165.2.33 pWCDMAECIOThresh	203
8.165.2.34 pWCDMARSSIDelta	203
8.165.2.35 pWCDMARSSIThresh	203
8.166 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference	203
8.166.1 Detailed Description	204
8.166.2 Field Documentation	206
8.166.2.1 pDDTMInd	206
8.166.2.2 pDualStandByPrefInd	206
8.166.2.3 pErrorRateInd	206
8.166.2.4 pHDRNewUATIAssInd	206

8.166.2.5 pHDRSessionCloseInd	207
8.166.2.6 pLTECphyCa	207
8.166.2.7 pManagedRoamingInd	207
8.166.2.8 pNetworkTimeInd	207
8.166.2.9 pServingSystemInd	207
8.166.2.10pSignalStrengthInd	207
8.166.2.11pSubscriptionInfoInd	207
8.166.2.12pSysInfoInd	207
8.166.2.13pSystemSelectionInd	207
8.167pack_nas_SLQSNasSwiOTAMessageCallback_t Struct Reference	207
8.167.1 Detailed Description	207
8.167.2 Field Documentation	208
8.167.2.1 gsmUmtsDI	208
8.167.2.2 gsmUmtsUI	208
8.167.2.3 lteEmmDI	208
8.167.2.4 lteEmmUI	208
8.167.2.5 lteEsmDI	208
8.167.2.6 lteEsmUI	208
8.167.2.7 pRankIndicatorInd	208
8.168pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference	208
8.168.1 Detailed Description	209
8.168.2 Field Documentation	209
8.168.2.1 bEnable	209
8.168.2.2 pSigIndReq	209
8.169pack_nas_SLQSSetSysSelectionPref_t Struct Reference	209
8.169.1 Detailed Description	209
8.169.2 Field Documentation	212
8.169.2.1 pAcqOrderPref	212
8.169.2.2 pBandPref	212
8.169.2.3 pChgDuration	213
8.169.2.4 pCSGID	213
8.169.2.5 pEmerMode	213
8.169.2.6 pGWAcqOrderPref	213
8.169.2.7 pLTEBandPref	213
8.169.2.8 pMNCIncPCSDigStat	213
8.169.2.9 pModePref	213
8.169.2.10pNetSelPref	213
8.169.2.11pPRLPref	213
8.169.2.12pRAT	213
8.169.2.13pRoamPref	213

8.169.2.14 pSrvDomainPref	213
8.169.2.15 pSrvRegRestriction	213
8.169.2.16 pTdsdmaBandPref	213
8.170 pack_qmi_t Struct Reference	213
8.170.1 Detailed Description	213
8.170.2 Field Documentation	213
8.170.2.1 msgid	213
8.170.2.2 svc	213
8.170.2.3 timeout	214
8.170.2.4 xid	214
8.171 pack_qos_SLQSQosSviReadApnExtraParams_t Struct Reference	214
8.171.1 Detailed Description	214
8.171.2 Field Documentation	214
8.171.2.1 apnId	214
8.172 pack_qos_SLQSQosSviReadDataStats_t Struct Reference	214
8.172.1 Detailed Description	214
8.172.2 Field Documentation	214
8.172.2.1 apnId	214
8.173 pack_qos_SLQSSetQosEventCallback_t Struct Reference	214
8.173.1 Detailed Description	215
8.173.2 Field Documentation	216
8.173.2.1 enable	216
8.174 pack_sms_SendSMS_t Struct Reference	216
8.174.1 Detailed Description	216
8.174.2 Field Documentation	216
8.174.2.1 messageFormat	216
8.174.2.2 messageSize	216
8.174.2.3 pLinktimer	216
8.174.2.4 pMessage	216
8.175 pack_sms_SetNewSMSCallback_t Struct Reference	217
8.175.1 Detailed Description	217
8.175.2 Field Documentation	217
8.175.2.1 status	217
8.176 pack_sms_SLQSDDeleteSMS_t Struct Reference	217
8.176.1 Detailed Description	218
8.176.2 Field Documentation	218
8.176.2.1 pMessageIndex	218
8.176.2.2 pMessageMode	218
8.176.2.3 pMessageTag	218
8.176.2.4 storageType	218

8.177	pack_sms_SLQSGetSMS_t Struct Reference	218
8.177.1	Detailed Description	218
8.177.2	Field Documentation	219
8.177.2.1	messageIndex	219
8.177.2.2	pMessageMode	219
8.177.2.3	storageType	219
8.178	pack_sms_SLQSGetSMSList_t Struct Reference	219
8.178.1	Detailed Description	219
8.178.2	Field Documentation	220
8.178.2.1	pMessageMode	220
8.178.2.2	pRequestedTag	220
8.178.2.3	storageType	220
8.179	pack_sms_SLQSModifySMSStatus_t Struct Reference	220
8.179.1	Detailed Description	221
8.179.2	Field Documentation	221
8.179.2.1	messageIndex	221
8.179.2.2	messageTag	221
8.179.2.3	pMessageMode	221
8.179.2.4	storageType	221
8.180	pack_swiloc_SwiLocSetAutoStart_t Struct Reference	221
8.180.1	Detailed Description	221
8.180.2	Field Documentation	223
8.180.2.1	fix_rate	223
8.180.2.2	fix_type	223
8.180.2.3	function	223
8.180.2.4	max_dist	223
8.180.2.5	max_time	223
8.180.2.6	set_fix_rate	223
8.180.2.7	set_fix_type	223
8.180.2.8	set_function	223
8.180.2.9	set_max_dist	223
8.180.2.10	set_max_time	223
8.181	pack_swioma_SLQSOMADMCancelSession_t Struct Reference	223
8.181.1	Detailed Description	223
8.181.2	Field Documentation	223
8.181.2.1	sessionType	223
8.182	pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference	223
8.182.1	Detailed Description	224
8.182.2	Field Documentation	224
8.182.2.1	SessionType	224

8.183pack_swioa_SLQSOMADMSelectSelection_t Struct Reference	224
8.183.1 Detailed Description	224
8.183.2 Field Documentation	225
8.183.2.1 pDeferTime	225
8.183.2.2 pRejectReason	225
8.183.2.3 selection	225
8.184pack_swioa_SLQSOMADMSetSettings_t Struct Reference	225
8.184.1 Detailed Description	225
8.184.2 Field Documentation	226
8.184.2.1 FOTAdownload	226
8.184.2.2 FOTAUpdate	226
8.184.2.3 pAutosdm	226
8.184.2.4 pFwAutoCheck	226
8.185pack_swioa_SLQSOMADMStartSession_t Struct Reference	226
8.185.1 Detailed Description	226
8.185.2 Field Documentation	226
8.185.2.1 sessionType	226
8.186pack_uim_ChangePin_t Struct Reference	226
8.186.1 Detailed Description	227
8.186.2 Field Documentation	227
8.186.2.1 changePIN	227
8.186.2.2 EncryptedPIN1	227
8.186.2.3 pIndicationToken	227
8.186.2.4 pKeyReferenceID	227
8.186.2.5 sessionInfo	227
8.186.2.6 Tlvresult	227
8.187pack_uim_ReadTransparent_t Struct Reference	227
8.187.1 Detailed Description	228
8.187.2 Field Documentation	228
8.187.2.1 fileIndex	228
8.187.2.2 pEncryptData	228
8.187.2.3 pIndicationToken	228
8.187.2.4 readTransparent	228
8.187.2.5 sessionInfo	229
8.187.2.6 Tlvresult	229
8.188pack_uim_SetPinProtection_t Struct Reference	229
8.188.1 Detailed Description	229
8.188.2 Field Documentation	229
8.188.2.1 EncryptedPIN1	229
8.188.2.2 pIndicationToken	230

8.188.2.3 pinProtection	230
8.188.2.4 pKeyReferenceID	230
8.188.2.5 sessionInfo	230
8.188.2.6 Tlvresult	230
8.189pack_uim_SLQSUIMEventRegister_t Struct Reference	230
8.189.1 Detailed Description	230
8.189.2 Field Documentation	230
8.189.2.1 eventMask	230
8.190pack_uim_SLQSUIMSwitchSlot_t Struct Reference	230
8.190.1 Detailed Description	230
8.190.2 Field Documentation	231
8.190.2.1 bLogicalSlot	231
8.190.2.2 ulPhysicalSlot	231
8.191pack_uim_UnblockPin_t Struct Reference	231
8.191.1 Detailed Description	231
8.191.2 Field Documentation	232
8.191.2.1 EncryptedPIN1	232
8.191.2.2 pIndicationToken	232
8.191.2.3 pinProtection	232
8.191.2.4 pKeyReferenceID	232
8.191.2.5 sessionInfo	232
8.191.2.6 Tlvresult	232
8.192pack_uim_VerifyPin_t Struct Reference	232
8.192.1 Detailed Description	233
8.192.2 Field Documentation	234
8.192.2.1 pEncryptedPIN1	234
8.192.2.2 pIndicationToken	234
8.192.2.3 pKeyReferenceID	234
8.192.2.4 sessionInfo	234
8.192.2.5 Tlvresult	234
8.192.2.6 verifyPIN	234
8.193pack_wds_GetDefaultProfile_t Struct Reference	234
8.193.1 Detailed Description	235
8.193.2 Field Documentation	235
8.193.2.1 profiletype	235
8.194pack_wds_GetDefaultProfileNum_t Struct Reference	235
8.194.1 Detailed Description	235
8.194.2 Field Documentation	235
8.194.2.1 family	235
8.194.2.2 type	235

8.195pack_wds_GetDormancyState_t Struct Reference	235
8.196pack_wds_GetLastMobileIPError_t Struct Reference	235
8.197pack_wds_GetMobileIP_t Struct Reference	235
8.198pack_wds_GetMobileIPProfile_t Struct Reference	235
8.198.1 Detailed Description	236
8.198.2 Field Documentation	237
8.198.2.1 index	237
8.199pack_wds_GetPacketStatus_t Struct Reference	237
8.199.1 Detailed Description	237
8.199.2 Field Documentation	237
8.199.2.1 statmask	237
8.200pack_wds_GetSessionDuration_t Struct Reference	237
8.201pack_wds_RMSetTransferStatistics_t Struct Reference	237
8.201.1 Detailed Description	237
8.201.2 Field Documentation	237
8.201.2.1 RmTrasnferStaticsReq	237
8.202pack_wds_SetDefaultProfile_t Struct Reference	237
8.202.1 Detailed Description	238
8.202.2 Field Documentation	238
8.202.2.1 authentication	238
8.202.2.2 ipAddress	238
8.202.2.3 pApnname	238
8.202.2.4 pdpType	238
8.202.2.5 pName	238
8.202.2.6 pPassword	238
8.202.2.7 primaryDNS	238
8.202.2.8 profileType	238
8.202.2.9 pUsername	238
8.202.2.10secondaryDNS	238
8.203pack_wds_SetDefaultProfileNum_t Struct Reference	238
8.203.1 Field Documentation	239
8.203.1.1 family	239
8.203.1.2 index	239
8.203.1.3 type	239
8.204pack_wds_SetMobileIPProfile_t Struct Reference	239
8.204.1 Detailed Description	239
8.204.2 Field Documentation	239
8.204.2.1 index	239
8.204.2.2 pAAASPI	239
8.204.2.3 pAddress	239

8.204.2.4 pEnabled	239
8.204.2.5 pHASPI	239
8.204.2.6 pMNAAB	240
8.204.2.7 pMNHA	240
8.204.2.8 pNAI	240
8.204.2.9 pPrimaryHA	240
8.204.2.10 pRevTunneling	240
8.204.2.11 pSecondaryHA	240
8.204.2.12 spc	240
8.205 pack_wds_SLQSCreateProfile_t Struct Reference	240
8.205.1 Detailed Description	240
8.205.2 Field Documentation	240
8.205.2.1 pCurProfile	240
8.205.2.2 pProfileId	240
8.205.2.3 pProfileType	241
8.206 pack_wds_SLQSDeleteProfile_t Struct Reference	241
8.206.1 Detailed Description	241
8.206.2 Field Documentation	241
8.206.2.1 profileIndex	241
8.206.2.2 profileType	241
8.207 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	241
8.208 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference	241
8.209 pack_wds_SLQSGetDUNCallInfo_t Struct Reference	241
8.209.1 Detailed Description	241
8.209.2 Field Documentation	242
8.209.2.1 Mask	242
8.209.2.2 pReportChannelRate	242
8.209.2.3 pReportConnStatus	242
8.209.2.4 pReportDataBearerTech	242
8.209.2.5 pReportDormStatus	242
8.209.2.6 pTransferStatInd	242
8.210 pack_wds_SLQSGetProfileSettings_t Struct Reference	242
8.210.1 Detailed Description	242
8.210.2 Field Documentation	243
8.210.2.1 ProfileId	243
8.210.2.2 ProfileType	243
8.211 pack_wds_SLQSGetRuntimeSettings_t Struct Reference	243
8.211.1 Detailed Description	243
8.211.2 Field Documentation	244
8.211.2.1 pReqSettings	244

8.212	pack_wds_SLQSModifyProfile_t Struct Reference	244
8.212.1	Detailed Description	244
8.212.2	Field Documentation	244
8.212.2.1	curProfile	244
8.212.2.2	pProfileId	244
8.212.2.3	pProfileType	244
8.213	pack_wds_SLQSSet3GPPConfigItem_t Struct Reference	244
8.213.1	Detailed Description	245
8.213.2	Field Documentation	245
8.213.2.1	_3gppRelease	245
8.213.2.2	defaultPDNEnabled	245
8.213.2.3	LTEAttachProfileList	245
8.213.2.4	LTEAttachProfileListLen	245
8.213.2.5	profileList	245
8.214	pack_wds_SLQSSetIPFamilyPreference_t Struct Reference	245
8.214.1	Detailed Description	246
8.214.2	Field Documentation	246
8.214.2.1	IPFamilyPreference	246
8.215	pack_wds_SLQSSetWdsEventCallback_t Struct Reference	246
8.215.1	Detailed Description	246
8.215.2	Field Documentation	246
8.215.2.1	currentDataBearer	246
8.215.2.2	dataBearer	246
8.215.2.3	dataSystemStatus	246
8.215.2.4	dormancyStatus	246
8.215.2.5	interval	247
8.215.2.6	mobileIP	247
8.215.2.7	transferStats	247
8.216	pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	247
8.216.1	Detailed Description	247
8.216.2	Field Documentation	247
8.216.2.1	pProfileId	247
8.217	pack_wds_SLQSStartDataSession_t Struct Reference	247
8.217.1	Detailed Description	247
8.217.2	Field Documentation	248
8.217.2.1	pAuth	248
8.217.2.2	pPass	248
8.217.2.3	pprofileid3gpp	248
8.217.2.4	pprofileid3gpp2	248
8.217.2.5	pTech	248

8.217.2.6 pUser	248
8.218pack_wds_SLQSSStopDataSession_t Struct Reference	248
8.218.1 Detailed Description	248
8.218.2 Field Documentation	249
8.218.2.1 psid	249
8.219pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	249
8.219.1 Detailed Description	249
8.219.2 Field Documentation	249
8.219.2.1 contextId	249
8.219.2.2 contextType	249
8.220PackCreateProfileOut Struct Reference	249
8.220.1 Field Documentation	249
8.220.1.1 ExtErrorCode	249
8.220.1.2 ProfileIndex	249
8.220.1.3 ProfileType	249
8.221packgetDyingGaspCfg Struct Reference	249
8.221.1 Detailed Description	249
8.221.2 Field Documentation	250
8.221.2.1 pDestSMSContent	250
8.221.2.2 pDestSMSNum	250
8.222packgetDyingGaspStatistics Struct Reference	250
8.222.1 Detailed Description	250
8.222.2 Field Documentation	250
8.222.2.1 pSMSAttemptedFlag	250
8.222.2.2 pTimeStamp	250
8.223qmiSmsMessageList Struct Reference	250
8.223.1 Detailed Description	250
8.223.2 Field Documentation	251
8.223.2.1 messageIndex	251
8.223.2.2 messageTag	251
8.224qmiWDSDataBearerTechnology Struct Reference	251
8.224.1 Detailed Description	251
8.224.2 Field Documentation	251
8.224.2.1 currentNetwork	251
8.224.2.2 ratMask	251
8.224.2.3 soMask	251
8.225RFBandInfoElements Struct Reference	251
8.225.1 Detailed Description	251
8.225.2 Field Documentation	252
8.225.2.1 activeBandClass	252

8.225.2.2 activeChannel	252
8.225.2.3 radiolInterface	252
8.226rmTrasferStaticsReq Struct Reference	252
8.226.1 Detailed Description	252
8.226.2 Field Documentation	252
8.226.2.1 bResetStatistics	252
8.226.2.2 ulMask	252
8.227slot_t Struct Reference	252
8.227.1 Detailed Description	252
8.227.2 Field Documentation	253
8.227.2.1 bICCID	253
8.227.2.2 bICCIDLength	253
8.227.2.3 bLogicalSlot	253
8.227.2.4 uPhyCardStatus	253
8.227.2.5 uPhySlotStatus	253
8.228slotInf Struct Reference	253
8.228.1 Detailed Description	254
8.228.2 Field Documentation	256
8.228.2.1 AppStatus	256
8.228.2.2 cardState	256
8.228.2.3 errorState	256
8.228.2.4 numApp	256
8.228.2.5 upinRetries	256
8.228.2.6 upinState	256
8.228.2.7 upukRetries	256
8.229slots_t Struct Reference	256
8.229.1 Field Documentation	256
8.229.1.1 uimSlotStatus	256
8.230sMSCAddress Struct Reference	256
8.230.1 Detailed Description	256
8.230.2 Field Documentation	256
8.230.2.1 data	257
8.230.2.2 length	257
8.231sMSCAddressTlv Struct Reference	257
8.231.1 Detailed Description	257
8.231.2 Field Documentation	257
8.231.2.1 SMSCInfo	257
8.231.2.2 TlvPresent	257
8.232sMSEtwsMessage Struct Reference	257
8.232.1 Detailed Description	257

8.232.2 Field Documentation	258
8.232.2.1 data	258
8.232.2.2 length	258
8.232.2.3 notificationType	258
8.233sMSEtwsMessageTlv Struct Reference	258
8.233.1 Detailed Description	258
8.233.2 Field Documentation	258
8.233.2.1 EtwsMessageInfo	258
8.233.2.2 TlvPresent	258
8.234sMSEtwsPlmn Struct Reference	258
8.234.1 Detailed Description	258
8.234.2 Field Documentation	259
8.234.2.1 mobileCountryCode	259
8.234.2.2 mobileNetworkCode	259
8.235sMSMessageMode Struct Reference	259
8.235.1 Detailed Description	259
8.235.2 Field Documentation	259
8.235.2.1 messageMode	259
8.236sMSMTMessage Struct Reference	259
8.236.1 Detailed Description	259
8.236.2 Field Documentation	259
8.236.2.1 messageIndex	259
8.236.2.2 storageType	259
8.237sMSOnIMS Struct Reference	260
8.237.1 Detailed Description	260
8.237.2 Field Documentation	260
8.237.2.1 smsOnIMS	260
8.238sMSOnIMSTlv Struct Reference	260
8.238.1 Detailed Description	260
8.238.2 Field Documentation	260
8.238.2.1 IMSInfo	260
8.238.2.2 TlvPresent	260
8.239sMSTransferRouteMTMessage Struct Reference	260
8.239.1 Detailed Description	261
8.239.2 Field Documentation	262
8.239.2.1 ackIndicator	262
8.239.2.2 data	262
8.239.2.3 format	262
8.239.2.4 length	262
8.239.2.5 transactionID	262

8.240tdscdmaSigInfoExt Struct Reference	262
8.240.1 Detailed Description	262
8.240.2 Field Documentation	262
8.240.2.1 ecio	263
8.240.2.2 rscp	263
8.240.2.3 rssi	263
8.240.2.4 sinr	263
8.241transferRouteMessageTlv Struct Reference	263
8.241.1 Detailed Description	263
8.241.2 Field Documentation	263
8.241.2.1 TlvPresent	263
8.241.2.2 TransferRouteMTMessageInfo	263
8.242transferStatInd Struct Reference	263
8.242.1 Detailed Description	263
8.242.2 Field Documentation	263
8.242.2.1 StatsMask	263
8.242.2.2 StatsPeriod	264
8.243uim_appStatus Struct Reference	264
8.243.1 Detailed Description	264
8.243.2 Field Documentation	267
8.243.2.1 aidLength	267
8.243.2.2 aidVal	267
8.243.2.3 appState	267
8.243.2.4 appType	267
8.243.2.5 persoFeature	267
8.243.2.6 persoRetries	267
8.243.2.7 persoState	267
8.243.2.8 persoUnblockRetries	267
8.243.2.9 pin1Retries	267
8.243.2.10pin1State	267
8.243.2.11pin2Retries	267
8.243.2.12pin2State	267
8.243.2.13puk1Retries	267
8.243.2.14puk2Retries	267
8.243.2.15univPin	267
8.244uim_cardResult Struct Reference	267
8.244.1 Detailed Description	267
8.244.2 Field Documentation	268
8.244.2.1 sw1	268
8.244.2.2 sw2	268

8.245uim_cardStatus Struct Reference	268
8.245.1 Detailed Description	268
8.245.2 Field Documentation	269
8.245.2.1 index1xPri	269
8.245.2.2 index1xSec	269
8.245.2.3 indexGwPri	269
8.245.2.4 indexGwSec	269
8.245.2.5 numSlot	269
8.245.2.6 SlotInfo	269
8.246uim_changeUIMPIN Struct Reference	269
8.246.1 Detailed Description	269
8.246.2 Field Documentation	270
8.246.2.1 oldPINLen	270
8.246.2.2 oldPINVal	270
8.246.2.3 pinID	270
8.246.2.4 pinLen	270
8.246.2.5 pinVal	270
8.247uim_encryptedPIN1 Struct Reference	270
8.247.1 Detailed Description	270
8.247.2 Field Documentation	271
8.247.2.1 pin1Len	271
8.247.2.2 pin1Val	271
8.248uim_fileInfo Struct Reference	271
8.248.1 Detailed Description	271
8.248.2 Field Documentation	271
8.248.2.1 fileID	271
8.248.2.2 path	271
8.248.2.3 pathLen	271
8.249uim_hotSwapStatus Struct Reference	272
8.249.1 Detailed Description	272
8.249.2 Field Documentation	272
8.249.2.1 hotSwap	272
8.249.2.2 hotSwapLength	272
8.250uim_readResult Struct Reference	272
8.250.1 Detailed Description	272
8.250.2 Field Documentation	273
8.250.2.1 content	273
8.250.2.2 contentLen	273
8.251uim_readTransparentInfo Struct Reference	273
8.251.1 Detailed Description	273

8.251.2 Field Documentation	273
8.251.2.1 length	273
8.251.2.2 offset	273
8.252uim_remainingRetries Struct Reference	273
8.252.1 Detailed Description	273
8.252.2 Field Documentation	274
8.252.2.1 unblockLeft	274
8.252.2.2 verifyLeft	274
8.253uim_sessionInformation Struct Reference	274
8.253.1 Detailed Description	274
8.253.2 Field Documentation	275
8.253.2.1 aid	275
8.253.2.2 aidLength	275
8.253.2.3 sessionType	275
8.254uim_setPINProtection Struct Reference	275
8.254.1 Detailed Description	275
8.254.2 Field Documentation	276
8.254.2.1 pinID	276
8.254.2.2 pinLength	276
8.254.2.3 pinOperation	276
8.254.2.4 pinValue	276
8.255uim_slotInfo Struct Reference	276
8.255.1 Detailed Description	276
8.255.2 Field Documentation	278
8.255.2.1 AppStatus	278
8.255.2.2 cardState	278
8.255.2.3 errorState	278
8.255.2.4 numApp	278
8.255.2.5 upinRetries	278
8.255.2.6 upinState	278
8.255.2.7 upukRetries	278
8.256uim_UIMSessionInformation Struct Reference	278
8.256.1 Detailed Description	278
8.256.2 Field Documentation	279
8.256.2.1 aid	279
8.256.2.2 aidLength	279
8.256.2.3 sessionType	279
8.257uim_unblockUIMPIN Struct Reference	279
8.257.1 Detailed Description	279
8.257.2 Field Documentation	280

8.257.2.1 newPINLen	280
8.257.2.2 newPINVal	280
8.257.2.3 pinID	280
8.257.2.4 pukLen	280
8.257.2.5 pukVal	280
8.258uim_verifyUIMPIN Struct Reference	280
8.258.1 Detailed Description	280
8.258.2 Field Documentation	281
8.258.2.1 pinID	281
8.258.2.2 pinLen	281
8.258.2.3 pinVal	281
8.259unpack_dms_GetActivationState_t Struct Reference	281
8.259.1 Detailed Description	281
8.259.2 Field Documentation	281
8.259.2.1 state	281
8.260unpack_dms_GetBandCapability_t Struct Reference	281
8.260.1 Field Documentation	282
8.260.1.1 BandCapability	282
8.260.1.2 Tlvresult	282
8.261unpack_dms_GetCrashAction_t Struct Reference	282
8.261.1 Field Documentation	282
8.261.1.1 DevCrashState	282
8.261.1.2 Tlvresult	282
8.262unpack_dms_GetCustFeature_t Struct Reference	282
8.262.1 Field Documentation	282
8.262.1.1 DHCPRelayEnabled	282
8.262.1.2 DisableIMSI	282
8.262.1.3 GpsEnable	282
8.262.1.4 GPSLPM	282
8.262.1.5 GPSSel	282
8.262.1.6 IPFamSupport	282
8.262.1.7 IsVoiceEnabled	283
8.262.1.8 RMAutoConnect	283
8.262.1.9 SMSSupport	283
8.262.1.10Tlvresult	283
8.263unpack_dms_GetCustFeaturesV2_t Struct Reference	283
8.263.1 Detailed Description	283
8.263.2 Field Documentation	283
8.263.2.1 GetCustomFeatureV2	283
8.263.2.2 Tlvresult	283

8.264unpack_dms_GetDeviceCap_t Struct Reference	283
8.264.1 Field Documentation	283
8.264.1.1 DataServiceCapability	283
8.264.1.2 MaxRXChannelRate	283
8.264.1.3 MaxTXChannelRate	283
8.264.1.4 Radiofaces	283
8.264.1.5 RadiofacesSize	284
8.264.1.6 SimCapability	284
8.264.1.7 Tlvresult	284
8.265unpack_dms_GetDeviceCapabilities_t Struct Reference	284
8.265.1 Detailed Description	284
8.265.2 Field Documentation	284
8.265.2.1 dataServiceCaCapability	284
8.265.2.2 maxRxChannelRate	284
8.265.2.3 maxTxChannelRate	284
8.265.2.4 Radiofaces	284
8.265.2.5 radiofacesSize	284
8.265.2.6 simCapability	284
8.266unpack_dms_GetDeviceHardwareRev_t Struct Reference	284
8.266.1 Field Documentation	285
8.266.1.1 String	285
8.266.1.2 stringSize	285
8.266.1.3 Tlvresult	285
8.267unpack_dms_GetDeviceMfr_t Struct Reference	285
8.267.1 Field Documentation	285
8.267.1.1 String	285
8.267.1.2 stringSize	285
8.267.1.3 Tlvresult	285
8.268unpack_dms_GetDeviceSerialNumbers_t Struct Reference	285
8.268.1 Field Documentation	285
8.268.1.1 esnSize	285
8.268.1.2 ESNString	285
8.268.1.3 imeiSize	285
8.268.1.4 IMEIString	285
8.268.1.5 imeiSvnSize	285
8.268.1.6 IimeiSvnString	285
8.268.1.7 meidSize	286
8.268.1.8 MEIDString	286
8.268.1.9 Tlvresult	286
8.269unpack_dms_GetFirmwareInfo_t Struct Reference	286

8.269.1 Detailed Description	286
8.269.2 Field Documentation	286
8.269.2.1 appversion_str	286
8.269.2.2 bootversion_str	286
8.269.2.3 carrier_str	286
8.269.2.4 cur_carr_name	286
8.269.2.5 cur_carr_rev	286
8.269.2.6 modelid_str	286
8.269.2.7 packageid_str	287
8.269.2.8 priversion_str	287
8.269.2.9 sku_str	287
8.269.2.10Tlvresult	287
8.270unpack_dms_GetFirmwareRevision_t Struct Reference	287
8.270.1 Field Documentation	287
8.270.1.1 amssSize	287
8.270.1.2 AMSSString	287
8.270.1.3 Tlvresult	287
8.271unpack_dms_GetFirmwareRevisions_t Struct Reference	287
8.271.1 Detailed Description	287
8.271.2 Field Documentation	287
8.271.2.1 amssSize	287
8.271.2.2 AMSSString	287
8.271.2.3 bootSize	287
8.271.2.4 BootString	288
8.271.2.5 priSize	288
8.271.2.6 PRIString	288
8.271.2.7 Tlvresult	288
8.272unpack_dms_GetFSN_t Struct Reference	288
8.272.1 Field Documentation	288
8.272.1.1 String	288
8.272.1.2 Tlvresult	288
8.273unpack_dms_GetHardwareRevision_t Struct Reference	288
8.273.1 Detailed Description	288
8.273.2 Field Documentation	288
8.273.2.1 hwVer	288
8.274unpack_dms_GetIMSI_t Struct Reference	288
8.274.1 Field Documentation	288
8.274.1.1 imsi	288
8.274.1.2 Tlvresult	288
8.275unpack_dms_GetModelID_t Struct Reference	289

8.275.1 Detailed Description	289
8.275.2 Field Documentation	289
8.275.2.1 modelid	289
8.275.2.2 Tlvresult	289
8.276unpack_dms_GetNetworkTime_t Struct Reference	289
8.276.1 Detailed Description	289
8.276.2 Field Documentation	289
8.276.2.1 source	289
8.276.2.2 timestamp	289
8.276.2.3 Tlvresult	289
8.277unpack_dms_GetPower_t Struct Reference	289
8.277.1 Detailed Description	290
8.277.2 Field Documentation	290
8.277.2.1 HardwareControlledMode	290
8.277.2.2 OfflineReason	290
8.277.2.3 OperationMode	290
8.277.2.4 Tlvresult	290
8.278unpack_dms_GetPRLVersion_t Struct Reference	290
8.278.1 Field Documentation	290
8.278.1.1 Tlvresult	290
8.278.1.2 u16PRLVersion	290
8.278.1.3 u8PRLPreference	290
8.279unpack_dms_GetSerialNumbers_t Struct Reference	290
8.279.1 Detailed Description	290
8.279.2 Field Documentation	291
8.279.2.1 esn	291
8.279.2.2 imei_no	291
8.279.2.3 imeisv_svn	291
8.279.2.4 meid	291
8.280unpack_dms_GetUSBComp_t Struct Reference	291
8.280.1 Field Documentation	291
8.280.1.1 NumSupUSBComps	291
8.280.1.2 SupUSBComps	291
8.280.1.3 Tlvresult	291
8.280.1.4 USBComp	291
8.281unpack_dms_GetVoiceNumber_t Struct Reference	291
8.281.1 Field Documentation	291
8.281.1.1 MIN	291
8.281.1.2 minSize	291
8.281.1.3 Tlvresult	292

8.281.1.4 VoiceNumber	292
8.281.1.5 voiceNumberSize	292
8.282unpack_dms_SetCustFeature_t Struct Reference	292
8.282.1 Field Documentation	292
8.282.1.1 Tlvresult	292
8.283unpack_dms_SetCustFeaturesV2_t Struct Reference	292
8.283.1 Detailed Description	292
8.283.2 Field Documentation	292
8.283.2.1 Tlvresult	292
8.284unpack_dms_SetEventReport_ind_t Struct Reference	292
8.284.1 Detailed Description	292
8.284.2 Field Documentation	293
8.284.2.1 ActivationStatusTlv	293
8.284.2.2 OperatingModeTlv	293
8.284.2.3 Tlvresult	293
8.285unpack_dms_SetEventReport_t Struct Reference	293
8.285.1 Field Documentation	293
8.285.1.1 Tlvresult	293
8.286unpack_dms_SetFirmwarePreference_t Struct Reference	293
8.286.1 Field Documentation	293
8.286.1.1 Tlvresult	293
8.287unpack_dms_SetPower_t Struct Reference	293
8.287.1 Field Documentation	293
8.287.1.1 Tlvresult	293
8.288unpack_dms_SetUSBComp_t Struct Reference	294
8.288.1 Field Documentation	294
8.288.1.1 Tlvresult	294
8.289unpack_dms_SLQSDmsSwiGetResetInfo_ind_t Struct Reference	294
8.289.1 Detailed Description	294
8.289.2 Field Documentation	295
8.289.2.1 source	295
8.289.2.2 Tlvresult	295
8.289.2.3 type	295
8.290unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference	295
8.290.1 Detailed Description	295
8.290.2 Field Documentation	296
8.290.2.1 source	296
8.290.2.2 Tlvresult	296
8.290.2.3 type	296
8.291unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	296

8.291.1 Detailed Description	296
8.291.2 Field Documentation	296
8.291.2.1 Tlvresult	296
8.292unpack_dms_SLQSGetBandCapability_t Struct Reference	296
8.292.1 Detailed Description	296
8.292.2 Field Documentation	298
8.292.2.1 bandCapability	298
8.292.2.2 is_LteBandCapability_Available	298
8.292.2.3 is_TdsBandCapability_Available	298
8.292.2.4 LteBandCapability	298
8.292.2.5 TdsBandCapability	298
8.293unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference	298
8.293.1 Detailed Description	298
8.293.2 Field Documentation	298
8.293.2.1 Tlvresult	298
8.294unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference	298
8.294.1 Detailed Description	299
8.294.2 Field Documentation	300
8.294.2.1 pGetDyingGaspCfg	300
8.294.2.2 Tlvresult	300
8.295unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference	300
8.295.1 Detailed Description	300
8.295.2 Field Documentation	300
8.295.2.1 pGetDyingGaspStatistics	300
8.295.2.2 Tlvresult	300
8.296unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference	300
8.296.1 Detailed Description	300
8.296.2 Field Documentation	301
8.296.2.1 carrier	301
8.296.2.2 fwvers	301
8.296.2.3 numEntries	301
8.296.2.4 pCurrImgInfo	301
8.296.2.5 pkgver	301
8.296.2.6 priver	301
8.297unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	301
8.297.1 Detailed Description	301
8.297.2 Field Documentation	302
8.297.2.1 Tlvresult	302
8.298unpack_dms_UIMGetICCID_t Struct Reference	302
8.298.1 Detailed Description	302

8.298.2 Field Documentation	302
8.298.2.1 String	302
8.298.2.2 stringSize	302
8.298.2.3 Tlvresult	302
8.299unpack_fms_GetImagesPreference_t Struct Reference	302
8.299.1 Detailed Description	302
8.299.2 Field Documentation	303
8.299.2.1 ImageListSize	303
8.299.2.2 plmageList	303
8.299.2.3 Tlvresult	303
8.300unpack_fms_GetStoredImages_t Struct Reference	303
8.300.1 Detailed Description	303
8.300.2 Field Documentation	303
8.300.2.1 imageList	303
8.300.2.2 imagelistSize	303
8.300.2.3 Tlvresult	304
8.301unpack_fms_SetImagesPreference_t Struct Reference	304
8.301.1 Detailed Description	304
8.301.2 Field Documentation	304
8.301.2.1 ImageTypes	304
8.301.2.2 ImageTypesSize	304
8.301.2.3 Tlvresult	304
8.302unpack_loc_Delete_Assist_Data_t Struct Reference	304
8.302.1 Detailed Description	304
8.302.2 Field Documentation	305
8.302.2.1 Tlvresult	305
8.303unpack_loc_EngineState_Ind_t Struct Reference	305
8.303.1 Detailed Description	305
8.303.2 Field Documentation	305
8.303.2.1 engineState	305
8.303.2.2 Tlvresult	305
8.304unpack_loc_EventRegister_t Struct Reference	305
8.304.1 Detailed Description	305
8.304.2 Field Documentation	306
8.304.2.1 Tlvresult	306
8.305unpack_loc_PositionRpt_Ind_t Struct Reference	306
8.305.1 Detailed Description	306
8.305.2 Field Documentation	311
8.305.2.1 pAltitudeAssumed	311
8.305.2.2 pAltitudeWrtEllipsoid	311

8.305.2.3 pAltitudeWrtMeanSeaLevel	311
8.305.2.4 pFixId	311
8.305.2.5 pGpsTime	311
8.305.2.6 pHeading	311
8.305.2.7 pHeadingUnc	311
8.305.2.8 pHorConfidence	311
8.305.2.9 pHorReliability	311
8.305.2.10pHorUncCircular	311
8.305.2.11pHorUncEllipseOrientAzimuth	311
8.305.2.12pHorUncEllipseSemiMajor	311
8.305.2.13pHorUncEllipseSemiMinor	311
8.305.2.14pLatitude	311
8.305.2.15pLeapSeconds	311
8.305.2.16pLongitude	311
8.305.2.17pMagneticDeviation	311
8.305.2.18pPrecisionDilution	312
8.305.2.19pSensorDataUsage	312
8.305.2.20pSpeedHorizontal	312
8.305.2.21pSpeedUnc	312
8.305.2.22pSpeedVertical	312
8.305.2.23pSvUsedforFix	312
8.305.2.24pTechnologyMask	312
8.305.2.25pTimeSrc	312
8.305.2.26pTimestampUtc	312
8.305.2.27pTimeUnc	312
8.305.2.28pVertConfidence	312
8.305.2.29pVertReliability	312
8.305.2.30pVertUnc	312
8.305.2.31sessionId	312
8.305.2.32sessionStatus	312
8.305.2.33Tlvresult	312
8.306unpack_loc_SetExtPowerState_t Struct Reference	312
8.306.1 Detailed Description	312
8.306.2 Field Documentation	312
8.306.2.1 Tlvresult	312
8.307unpack_loc_SetOperationMode_t Struct Reference	312
8.307.1 Detailed Description	313
8.307.2 Field Documentation	313
8.307.2.1 Tlvresult	313
8.308unpack_loc_Start_t Struct Reference	313

8.308.1 Detailed Description	313
8.308.2 Field Documentation	313
8.308.2.1 Tlvresult	313
8.309unpack_loc_Stop_t Struct Reference	313
8.309.1 Detailed Description	313
8.309.2 Field Documentation	314
8.309.2.1 Tlvresult	314
8.310unpack_nas_GetCDMANetworkParameters_t Struct Reference	314
8.310.1 Detailed Description	314
8.310.2 Field Documentation	314
8.310.2.1 Application	314
8.310.2.2 Broadcast	314
8.310.2.3 CustomSCP	314
8.310.2.4 ForceRev0	314
8.310.2.5 Protocol	314
8.310.2.6 RegForeignNID	315
8.310.2.7 RegForeignSID	315
8.310.2.8 RegHomeSID	315
8.310.2.9 Roaming	315
8.310.2.10SCI	315
8.310.2.11SCM	315
8.311unpack_nas_GetHomeNetwork_t Struct Reference	315
8.311.1 Detailed Description	315
8.311.2 Field Documentation	315
8.311.2.1 mcc	315
8.311.2.2 mnc	315
8.311.2.3 name	315
8.311.2.4 nid	315
8.311.2.5 sid	315
8.312unpack_nas_GetNetworkPreference_t Struct Reference	315
8.312.1 Detailed Description	316
8.312.2 Field Documentation	317
8.312.2.1 ActiveTechPref	317
8.312.2.2 Duration	317
8.312.2.3 PersistentTechPref	317
8.312.2.4 Tlvresult	317
8.313unpack_nas_GetRFInfo_t Struct Reference	317
8.313.1 Detailed Description	318
8.313.2 Field Documentation	318
8.313.2.1 instancesSize	318

8.313.2.2 RFBandInfoElements	318
8.314unpack_nas_GetServingNetwork_t Struct Reference	318
8.314.1 Detailed Description	318
8.314.2 Field Documentation	319
8.314.2.1 CSDomain	319
8.314.2.2 DataCaps	319
8.314.2.3 DataCapsLen	319
8.314.2.4 MCC	319
8.314.2.5 MNC	319
8.314.2.6 Name	319
8.314.2.7 nameSize	319
8.314.2.8 PSDomain	319
8.314.2.9 Radiolfaces	319
8.314.2.10RadiolfacesSize	319
8.314.2.11RAN	319
8.314.2.12RegistrationState	319
8.314.2.13Roaming	319
8.315unpack_nas_GetServingNetworkCapabilities_t Struct Reference	319
8.315.1 Detailed Description	319
8.315.2 Field Documentation	319
8.315.2.1 DataCaps	319
8.315.2.2 DataCapsLen	320
8.316unpack_nas_GetSignalStrengths_t Struct Reference	320
8.316.1 Detailed Description	320
8.316.2 Field Documentation	320
8.316.2.1 len	320
8.316.2.2 radio	320
8.316.2.3 rssi	320
8.317unpack_nas_PerformNetworkScan_t Struct Reference	320
8.317.1 Detailed Description	320
8.317.2 Field Documentation	320
8.317.2.1 p3GppNetworkInfoInstances	320
8.317.2.2 p3GppNetworkInstanceSize	320
8.317.2.3 pPCSInstance	321
8.317.2.4 pPCSInstanceSize	321
8.317.2.5 pRATInstance	321
8.317.2.6 pRATInstanceSize	321
8.317.2.7 pScanResult	321
8.318unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	321
8.318.1 Detailed Description	321

8.318.2 Field Documentation	321
8.318.2.1 dataCaps	321
8.318.2.2 dataCapsSize	321
8.319unpack_nas_SetEventReportInd_t Struct Reference	321
8.319.1 Detailed Description	321
8.319.2 Field Documentation	321
8.319.2.1 RFTlv	321
8.319.2.2 RRTlv	322
8.319.2.3 SLQSSSTlv	322
8.319.2.4 SSTlv	322
8.320unpack_nas_SetNetworkPreference_t Struct Reference	322
8.320.1 Detailed Description	322
8.320.2 Field Documentation	323
8.320.2.1 Tlvresult	323
8.321unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference	323
8.321.1 Detailed Description	323
8.321.2 Field Documentation	323
8.321.2.1 roaming	323
8.322unpack_nas_SetServingSystemCallback_ind_t Struct Reference	323
8.322.1 Detailed Description	324
8.322.2 Field Documentation	324
8.322.2.1 SSInfo	324
8.322.2.2 Tlvresult	324
8.323unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference	324
8.323.1 Detailed Description	324
8.323.2 Field Documentation	324
8.323.2.1 LTECphyCAInfo	324
8.323.2.2 Tlvresult	324
8.324unpack_nas_SLQSGetPLMNName_t Struct Reference	324
8.324.1 Field Documentation	325
8.324.1.1 longName	325
8.324.1.2 longNameCI	325
8.324.1.3 longNameEn	325
8.324.1.4 longNameLen	325
8.324.1.5 longNameSB	325
8.324.1.6 shortName	325
8.324.1.7 shortNameCI	325
8.324.1.8 shortNameEn	325
8.324.1.9 shortNameLen	325
8.324.1.10shortNameSB	325

8.324.1.11spn	325
8.324.1.12spnEncoding	325
8.324.1.13spnLength	325
8.325unpack_nas_SLQSGetServingSystem_t Struct Reference	325
8.325.1 Detailed Description	326
8.325.2 Field Documentation	327
8.325.2.1 BasestationID	327
8.325.2.2 BasestationLatitude	327
8.325.2.3 BasestationLongitude	327
8.325.2.4 CallBarStatus	327
8.325.2.5 CDMA_P_Rev	327
8.325.2.6 CDMASystemInfoExt	327
8.325.2.7 CellID	327
8.325.2.8 ConcSvcInfo	327
8.325.2.9 CurrentPLMN	327
8.325.2.10DataSrvCapabilities	327
8.325.2.11DefaultRoamInd	327
8.325.2.12DetailedSvcInfo	327
8.325.2.13DTMInd	327
8.325.2.14Gpp2TimeZone	327
8.325.2.15GppNetworkDSTAdjustment	327
8.325.2.16GppTimeZone	327
8.325.2.17HdrPersonality	327
8.325.2.18Lac	327
8.325.2.19NetworkID	327
8.325.2.20PRLInd	327
8.325.2.21RoamIndicatorVal	327
8.325.2.22RoamingIndicatorList	327
8.325.2.23ServingSystem	327
8.325.2.24SystemID	328
8.325.2.25TrackAreaCode	328
8.326unpack_nas_SLQSGetSignalStrength_t Struct Reference	328
8.326.1 Detailed Description	328
8.326.2 Field Documentation	328
8.326.2.1 ecioList	328
8.326.2.2 ecioListLen	328
8.326.2.3 errorRateList	328
8.326.2.4 errorRateListLen	328
8.326.2.5 lo	328
8.326.2.6 ltersrp	328

8.326.2.7 ltesnr	329
8.326.2.8 rsrqInfo	329
8.326.2.9 rxSignalStrengthList	329
8.326.2.10rxSignalStrengthListLen	329
8.326.2.11signalStrengthReqMask	329
8.326.2.12sinr	329
8.327unpack_nas_SLQSGetSysInfo_t Struct Reference	329
8.327.1 Detailed Description	329
8.327.2 Field Documentation	331
8.327.2.1 pAddCDMASysInfo	331
8.327.2.2 pAddGSMSysInfo	331
8.327.2.3 pAddHDRSysInfo	331
8.327.2.4 pAddLTESysInfo	331
8.327.2.5 pAddWCDMASysInfo	331
8.327.2.6 pCDMASrvStatusInfo	331
8.327.2.7 pCDMASysInfo	331
8.327.2.8 pGSMCallBarringSysInfo	331
8.327.2.9 pGSMCipherDomainSysInfo	331
8.327.2.10pGSMSrvStatusInfo	331
8.327.2.11pGSMSysInfo	331
8.327.2.12pHDRSrvStatusInfo	331
8.327.2.13pHDRSysInfo	331
8.327.2.14pLTESrvStatusInfo	331
8.327.2.15pLTESysInfo	331
8.327.2.16pLTEVoiceSupportSysInfo	331
8.327.2.17pWCDMACallBarringSysInfo	331
8.327.2.18pWCDMACipherDomainSysInfo	331
8.327.2.19pWCDMASrvStatusInfo	331
8.327.2.20pWCDMASysInfo	331
8.328unpack_nas_SLQSGetSysSelectionPref_t Struct Reference	331
8.328.1 Detailed Description	332
8.328.2 Field Documentation	335
8.328.2.1 pBandPref	335
8.328.2.2 pEmerMode	335
8.328.2.3 pGWAcqOrderPref	335
8.328.2.4 pLTEBandPref	335
8.328.2.5 pModePref	335
8.328.2.6 pNetSelPref	335
8.328.2.7 pPRLPref	335
8.328.2.8 pRoamPref	335

8.328.2.9 pSrvDomainPref	335
8.329unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference	335
8.329.1 Detailed Description	336
8.329.2 Field Documentation	336
8.329.2.1 pCDMAInfo	336
8.329.2.2 pGERANInfo	336
8.329.2.3 pLTEInfoInterfreq	337
8.329.2.4 pLTEInfoIntrafreq	337
8.329.2.5 pLTEInfoNeighboringGSM	337
8.329.2.6 pLTEInfoNeighboringWCDMA	337
8.329.2.7 pUMTSCellID	337
8.329.2.8 pUMTSInfo	337
8.329.2.9 pWCDMAInfoLTENeighborCell	337
8.330unpack_nas_SLQSNasGetSigInfo_t Struct Reference	337
8.330.1 Detailed Description	337
8.330.2 Field Documentation	337
8.330.2.1 CDMASigInfo	337
8.330.2.2 GSMSSInfo	337
8.330.2.3 HDRSSInfo	337
8.330.2.4 LTESigInfo	337
8.330.2.5 WCDMASigInfo	337
8.331unpack_nas_SLQSNasSigInfoCallback_t Struct Reference	337
8.331.1 Detailed Description	338
8.331.2 Field Documentation	338
8.331.2.1 pCDMASigInfo	338
8.331.2.2 pGSMSSigInfo	338
8.331.2.3 pHDRSigInfo	338
8.331.2.4 pLTESigInfo	338
8.331.2.5 pRscp	338
8.331.2.6 pTDSCDMASigInfoExt	338
8.331.2.7 pWCDMASigInfo	338
8.332unpack_nas_SLQSNasSwiModemStatus_t Struct Reference	338
8.332.1 Detailed Description	338
8.332.2 Field Documentation	339
8.332.2.1 commonInfo	339
8.332.2.2 pLTEInfo	339
8.333unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference	339
8.333.1 Detailed Description	339
8.333.2 Field Documentation	339
8.333.2.1 Info	339

8.333.2.2 Tlvresult	339
8.334unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference	339
8.334.1 Detailed Description	339
8.334.2 Field Documentation	340
8.334.2.1 Info	340
8.334.2.2 Tlvresult	340
8.335unpack_nas_SLQSSwiGetLteCQI_t Struct Reference	340
8.335.1 Detailed Description	340
8.335.2 Field Documentation	341
8.335.2.1 CQIValueCW0	341
8.335.2.2 CQIValueCW1	341
8.335.2.3 ValidityCW0	341
8.335.2.4 ValidityCW1	341
8.336unpack_nas_SLQSSysInfoCallback_t Struct Reference	341
8.336.1 Detailed Description	341
8.336.2 Field Documentation	343
8.336.2.1 pAddCDMASysInfo	343
8.336.2.2 pAddGSMSysInfo	343
8.336.2.3 pAddHDRSysInfo	343
8.336.2.4 pAddLTESysInfo	343
8.336.2.5 pAddWCDMASysInfo	343
8.336.2.6 pCDMASrvStatusInfo	343
8.336.2.7 pCDMASysInfo	343
8.336.2.8 pGSMCallBarringSysInfo	343
8.336.2.9 pGSMCipherDomainSysInfo	343
8.336.2.10pGSMSrvStatusInfo	344
8.336.2.11pGSMSysInfo	344
8.336.2.12pHDRSrvStatusInfo	344
8.336.2.13pHDRSysInfo	344
8.336.2.14pLTESrvStatusInfo	344
8.336.2.15pLTESysInfo	344
8.336.2.16pLTEVoiceSupportSysInfo	344
8.336.2.17pSysInfoNoChange	344
8.336.2.18pWCDMACallBarringSysInfo	344
8.336.2.19pWCDMACipherDomainSysInfo	344
8.336.2.20pWCDMASrvStatusInfo	344
8.336.2.21pWCDMASysInfo	344
8.337unpack_omaDmConfigTlv_t Struct Reference	344
8.337.1 Detailed Description	344
8.337.2 Field Documentation	345

8.337.2.1 alertmsg	345
8.337.2.2 alertmsglength	345
8.337.2.3 state	345
8.337.2.4 userInputReq	345
8.337.2.5 userInputTimeout	345
8.338unpack_omaDmFotaTlv_t Struct Reference	345
8.338.1 Detailed Description	345
8.338.2 Field Documentation	347
8.338.2.1 description	347
8.338.2.2 descriptionlength	347
8.338.2.3 fwdloadsize	347
8.338.2.4 fwloadComplete	347
8.338.2.5 namelength	347
8.338.2.6 package_name	347
8.338.2.7 sessionType	347
8.338.2.8 severity	347
8.338.2.9 state	347
8.338.2.10updateCompleteStatus	347
8.338.2.11userInputReq	347
8.338.2.12userInputTimeout	347
8.338.2.13version	347
8.338.2.14versionlength	347
8.339unpack_omaDmNotificationsTlv_t Struct Reference	347
8.339.1 Field Documentation	348
8.339.1.1 notification	348
8.339.1.2 sessionStatus	348
8.340unpack_qmi_t Struct Reference	348
8.340.1 Detailed Description	348
8.340.2 Field Documentation	348
8.340.2.1 msgid	348
8.340.2.2 type	348
8.340.2.3 xid	348
8.341unpack_qos_dataRate_t Struct Reference	348
8.341.1 Detailed Description	348
8.341.2 Field Documentation	348
8.341.2.1 dataRateMax	348
8.341.2.2 guaranteedRate	348
8.342unpack_qos_IPv4Addr_t Struct Reference	349
8.342.1 Detailed Description	349
8.342.2 Field Documentation	349

8.342.2.1 addr	349
8.342.2.2 subnetMask	349
8.343unpack_qos_IPv6Addr_t Struct Reference	349
8.343.1 Detailed Description	349
8.343.2 Field Documentation	349
8.343.2.1 addr	349
8.343.2.2 prefixLen	349
8.344unpack_qos_IPv6TrafCls_t Struct Reference	349
8.344.1 Detailed Description	350
8.344.2 Field Documentation	350
8.344.2.1 mask	350
8.344.2.2 val	350
8.345unpack_qos_pktErrRate_t Struct Reference	350
8.345.1 Detailed Description	350
8.345.2 Field Documentation	350
8.345.2.1 exponent	350
8.345.2.2 multiplier	350
8.346unpack_qos_Port_t Struct Reference	350
8.346.1 Detailed Description	351
8.346.2 Field Documentation	351
8.346.2.1 port	351
8.346.2.2 range	351
8.347unpack_qos_QosFlowInfo_t Struct Reference	351
8.347.1 Detailed Description	351
8.347.2 Field Documentation	352
8.347.2.1 BearerID	352
8.347.2.2 is_RxQFlowGranted_Available	352
8.347.2.3 is_TxQFlowGranted_Available	352
8.347.2.4 NumRxFilters	352
8.347.2.5 NumTxFilters	352
8.347.2.6 QFlowState	352
8.347.2.7 RxQFilter	353
8.347.2.8 RxQFlowGranted	353
8.347.2.9 TxQFilter	353
8.347.2.10TxQFlowGranted	353
8.348unpack_qos_QosFlowInfoState_t Struct Reference	353
8.348.1 Detailed Description	353
8.348.2 Field Documentation	353
8.348.2.1 id	353
8.348.2.2 isNewFlow	353

8.348.2.3 state	353
8.349unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference	353
8.349.1 Detailed Description	354
8.349.2 Field Documentation	355
8.349.2.1 NWQoSStatus	355
8.350unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference	355
8.350.1 Detailed Description	355
8.350.2 Field Documentation	356
8.350.2.1 ambr_dl	356
8.350.2.2 ambr_dl_ext	356
8.350.2.3 ambr_dl_ext2	356
8.350.2.4 ambr_ul	356
8.350.2.5 ambr_ul_ext	356
8.350.2.6 ambr_ul_ext2	356
8.350.2.7 apnId	356
8.351unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference	356
8.351.1 Detailed Description	357
8.351.2 Field Documentation	357
8.351.2.1 apnId	358
8.351.2.2 numQosFlow	358
8.351.2.3 qosFlow	358
8.351.2.4 total_rx_bytes	358
8.351.2.5 total_rx_pkt	358
8.351.2.6 total_tx_bytes	358
8.351.2.7 total_tx_bytes_drp	358
8.351.2.8 total_tx_pkt	358
8.351.2.9 total_tx_pkt_drp	358
8.352unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference	358
8.352.1 Detailed Description	358
8.352.2 Field Documentation	358
8.352.2.1 NumFlows	358
8.352.2.2 QosFlowInfo	358
8.353unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference	358
8.353.1 Detailed Description	359
8.353.2 Field Documentation	360
8.353.2.1 status	360
8.354unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference	360
8.354.1 Detailed Description	360
8.354.2 Field Documentation	360
8.354.2.1 event	360

8.355unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference	360
8.355.1 Detailed Description	360
8.355.2 Field Documentation	362
8.355.2.1 event	362
8.355.2.2 id	362
8.355.2.3 reason	362
8.355.2.4 status	362
8.356unpack_qos_swiQosFilter_t Struct Reference	362
8.356.1 Detailed Description	362
8.356.2 Field Documentation	364
8.356.2.1 EspSpi	364
8.356.2.2 Id	364
8.356.2.3 index	364
8.356.2.4 IPv4DstAddr	364
8.356.2.5 IPv4SrcAddr	364
8.356.2.6 IPv4Tos	364
8.356.2.7 IPv6DstAddr	364
8.356.2.8 IPv6Label	364
8.356.2.9 IPv6SrcAddr	364
8.356.2.10IPv6TrafCls	364
8.356.2.11is_EspSpi_Available	364
8.356.2.12s_Id_Available	364
8.356.2.13s_IPv4DstAddr_Available	365
8.356.2.14s_IPv4SrcAddr_Available	365
8.356.2.15s_IPv4Tos_Available	365
8.356.2.16s_IPv6DstAddr_Available	365
8.356.2.17s_IPv6Label_Available	365
8.356.2.18s_IPv6SrcAddr_Available	365
8.356.2.19s_IPv6TrafCls_Available	365
8.356.2.20s_NxtHdrProto_Available	365
8.356.2.21is_Precedence_Available	365
8.356.2.22s_TCPDstPort_Available	365
8.356.2.23s_TCPSrcPort_Available	365
8.356.2.24s_TranDstPort_Available	365
8.356.2.25s_TranSrcPort_Available	365
8.356.2.26s_UDPDstPort_Available	365
8.356.2.27s_UDPSrcPort_Available	365
8.356.2.28NxtHdrProto	365
8.356.2.29Precedence	365
8.356.2.30TCPDstPort	365

8.356.2.31TCPSrcPort	365
8.356.2.32TranDstPort	365
8.356.2.33TranSrcPort	365
8.356.2.34UDPDstPort	365
8.356.2.35UDPSrcPort	365
8.356.2.36version	365
8.357unpack_qos_swiQosFlow_t Struct Reference	365
8.357.1 Detailed Description	366
8.357.2 Field Documentation	369
8.357.2.1 DataRate	369
8.357.2.2 index	369
8.357.2.3 is_DataRate_Available	369
8.357.2.4 is_Jitter_Available	369
8.357.2.5 is_Latency_Available	369
8.357.2.6 is_LteQci_Available	369
8.357.2.7 is_MaxAllowedPktSz_Available	369
8.357.2.8 is_MinPktSz_Available	369
8.357.2.9 is_PktErrRate_Available	369
8.357.2.10s_ProfileId3GPP2_Available	369
8.357.2.11is_TokenBucket_Available	369
8.357.2.12s_TrafficClass_Available	369
8.357.2.13s_val_3GPP2Pri_Available	369
8.357.2.14s_val_3GPPImCn_Available	369
8.357.2.15s_val_3GPPResResidualBER_Available	369
8.357.2.16s_val_3GPPSigInd_Available	369
8.357.2.17s_val_3GPPTraHdlPri_Available	370
8.357.2.18Jitter	370
8.357.2.19Latency	370
8.357.2.20LteQci	370
8.357.2.21MaxAllowedPktSz	370
8.357.2.22MinPktSz	370
8.357.2.23PktErrRate	370
8.357.2.24ProfileId3GPP2	370
8.357.2.25TokenBucket	370
8.357.2.26TrafficClass	370
8.357.2.27val_3GPP2Pri	370
8.357.2.28val_3GPPImCn	370
8.357.2.29val_3GPPResResidualBER	370
8.357.2.30val_3GPPSigInd	370
8.357.2.31val_3GPPTraHdlPri	370

8.358unpack_qos_tokenBucket_t Struct Reference	370
8.358.1 Detailed Description	370
8.358.2 Field Documentation	370
8.358.2.1 bucketSz	370
8.358.2.2 peakRate	371
8.358.2.3 tokenRate	371
8.359unpack_qos_Tos_t Struct Reference	371
8.359.1 Detailed Description	371
8.359.2 Field Documentation	371
8.359.2.1 mask	371
8.359.2.2 val	371
8.360unpack_QosFlowStat_t Struct Reference	371
8.360.1 Detailed Description	371
8.360.2 Field Documentation	372
8.360.2.1 bearerId	372
8.360.2.2 tx_bytes	372
8.360.2.3 tx_bytes_drp	372
8.360.2.4 tx_pkt	372
8.360.2.5 tx_pkt_drp	372
8.361unpack_sms_SendSMS_t Struct Reference	372
8.361.1 Detailed Description	372
8.361.2 Field Documentation	372
8.361.2.1 messageFailureCode	372
8.361.2.2 messageId	373
8.362unpack_sms_SetNewSMSCallback_ind_t Struct Reference	373
8.362.1 Detailed Description	373
8.362.2 Field Documentation	374
8.362.2.1 ETWSPLMNTlv	374
8.362.2.2 ETWSTlv	374
8.362.2.3 IMSTlv	374
8.362.2.4 MMTlv	374
8.362.2.5 NewMMTlv	374
8.362.2.6 SMSCTlv	374
8.362.2.7 TRMessageTlv	374
8.363unpack_sms_SetNewSMSCallback_t Struct Reference	374
8.364unpack_sms_SLQSDeleteSMS_t Struct Reference	374
8.365unpack_sms_SLQSGetSMS_t Struct Reference	374
8.365.1 Detailed Description	375
8.365.2 Field Documentation	375
8.365.2.1 message	375

8.365.2.2 messageFormat	375
8.365.2.3 messageSize	375
8.365.2.4 messageTag	375
8.366unpack_sms_SLQSGetSMSList_t Struct Reference	375
8.366.1 Detailed Description	375
8.366.2 Field Documentation	376
8.366.2.1 messageList	376
8.366.2.2 messageListSize	376
8.367unpack_sms_SLQSModifySMSStatus_t Struct Reference	376
8.368unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference	376
8.368.1 Detailed Description	376
8.368.2 Field Documentation	376
8.368.2.1 messageMode	376
8.368.2.2 storageType	376
8.369unpack_swiloc_SwiLocGetAutoStart_t Struct Reference	376
8.369.1 Detailed Description	377
8.369.2 Field Documentation	378
8.369.2.1 fix_rate	378
8.369.2.2 fix_rate_reported	378
8.369.2.3 fix_type	378
8.369.2.4 fix_type_reported	378
8.369.2.5 function	378
8.369.2.6 function_reported	378
8.369.2.7 max_dist	378
8.369.2.8 max_dist_reported	378
8.369.2.9 max_time	378
8.369.2.10max_time_reported	378
8.370unpack_swioma_SLQSOMADMAAlertCallback_ind_t Struct Reference	378
8.370.1 Detailed Description	379
8.370.2 Field Documentation	379
8.370.2.1 eventType	379
8.370.2.2 SessionInfoConfig	379
8.370.2.3 SessionInfoFota	379
8.370.2.4 SessionInfoNotification	379
8.371unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference	379
8.371.1 Detailed Description	380
8.371.2 Field Documentation	381
8.371.2.1 Date	382
8.371.2.2 DateLength	382
8.371.2.3 PkgDescLength	382

8.371.2.4 PkgDescription	382
8.371.2.5 PkgName	382
8.371.2.6 PkgNameLength	382
8.371.2.7 RetryCount	382
8.371.2.8 SessionState	382
8.371.2.9 SessionType	382
8.371.2.10Severity	382
8.371.2.11Source	382
8.371.2.12SourceLength	382
8.371.2.13Status	382
8.371.2.14Time	382
8.371.2.15TimeLength	382
8.371.2.16UpdateCompleteStatus	382
8.372unpack_swioama_SLQSOMADMGetSettings_t Struct Reference	382
8.372.1 Detailed Description	382
8.372.2 Field Documentation	384
8.372.2.1 Autosdm	384
8.372.2.2 FOTAdownload	384
8.372.2.3 FOTAUpdate	384
8.372.2.4 FwAutoCheck	384
8.372.2.5 OMADMEEnabled	384
8.373unpack_swioama_SLQSOMADMStartSession_t Struct Reference	384
8.373.1 Detailed Description	384
8.373.2 Field Documentation	384
8.373.2.1 FwAvailability	384
8.374unpack_uim_ChangePin_t Struct Reference	384
8.374.1 Detailed Description	385
8.374.2 Field Documentation	385
8.374.2.1 pEncryptedPIN1	385
8.374.2.2 pIndicationToken	385
8.374.2.3 pRemainingRetries	385
8.374.2.4 Tlvresult	385
8.375unpack_uim_GetCardStatus_t Struct Reference	385
8.375.1 Detailed Description	385
8.375.2 Field Documentation	386
8.375.2.1 pCardStatus	386
8.375.2.2 pHotSwapStatus	386
8.375.2.3 Tlvresult	386
8.376unpack_uim_ReadTransparent_t Struct Reference	386
8.376.1 Detailed Description	386

8.376.2 Field Documentation	387
8.376.2.1 pCardResult	387
8.376.2.2 pEncryptedData	387
8.376.2.3 pIndicationToken	387
8.376.2.4 pReadResult	387
8.376.2.5 Tlvresult	387
8.377unpack_uim_SetPinProtection_t Struct Reference	387
8.377.1 Detailed Description	387
8.377.2 Field Documentation	388
8.377.2.1 pEncryptedPIN1	388
8.377.2.2 pIndicationToken	388
8.377.2.3 pRemainingRetries	388
8.377.2.4 Tlvresult	388
8.378unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference	388
8.378.1 Detailed Description	388
8.378.2 Field Documentation	388
8.378.2.1 bNumberOfPhySlots	388
8.378.2.2 slotsstatusChange	388
8.379unpack_uim_SLQSUIEventRegister_t Struct Reference	388
8.379.1 Detailed Description	388
8.379.2 Field Documentation	389
8.379.2.1 eventMask	389
8.380unpack_uim_SLQSUIGetSlotsStatus_t Struct Reference	389
8.380.1 Detailed Description	389
8.380.2 Field Documentation	389
8.380.2.1 pNumberOfPhySlot	389
8.380.2.2 pUimSlotsStatus	389
8.381unpack_uim_SLQSUISetStatusChangeCallBack_ind_t Struct Reference	389
8.381.1 Detailed Description	389
8.381.2 Field Documentation	390
8.381.2.1 pCardStatus	390
8.382unpack_uim_UnblockPin_t Struct Reference	390
8.382.1 Detailed Description	390
8.382.2 Field Documentation	390
8.382.2.1 pEncryptedPIN1	390
8.382.2.2 pIndicationToken	390
8.382.2.3 pRemainingRetries	390
8.382.2.4 Tlvresult	390
8.383unpack_uim_VerifyPin_t Struct Reference	390
8.383.1 Detailed Description	391

8.383.2 Field Documentation	392
8.383.2.1 pEncryptedPIN1	392
8.383.2.2 pIndicationToken	392
8.383.2.3 pRemainingRetries	392
8.383.2.4 Tlvresult	392
8.384unpack_wds_GetConnectionRate_t Struct Reference	392
8.384.1 Detailed Description	392
8.384.2 Field Documentation	393
8.384.2.1 currentChannelRXRate	393
8.384.2.2 currentChannelTXRate	393
8.384.2.3 maxChannelRXRate	393
8.384.2.4 maxChannelTXRate	393
8.385unpack_wds_GetDefaultProfile_t Struct Reference	393
8.385.1 Detailed Description	393
8.385.2 Field Documentation	393
8.385.2.1 apnname	393
8.385.2.2 apnsize	393
8.385.2.3 auth	394
8.385.2.4 ipaddr	394
8.385.2.5 ipaddrv6	394
8.385.2.6 name	394
8.385.2.7 namesize	394
8.385.2.8 pdptype	394
8.385.2.9 pridns	394
8.385.2.10pridnsv6	394
8.385.2.11secdns	394
8.385.2.12secdnsv6	394
8.385.2.13username	394
8.385.2.14usersize	394
8.386unpack_wds_GetDefaultProfileNum_t Struct Reference	394
8.386.1 Detailed Description	394
8.386.2 Field Documentation	394
8.386.2.1 index	394
8.387unpack_wds_GetDormancyState_t Struct Reference	394
8.387.1 Detailed Description	394
8.387.2 Field Documentation	395
8.387.2.1 dormancyState	395
8.388unpack_wds_GetLastMobileIPError_t Struct Reference	395
8.388.1 Detailed Description	395
8.388.2 Field Documentation	395

8.388.2.1 error	395
8.389unpack_wds_GetMobileIP_t Struct Reference	395
8.389.1 Detailed Description	395
8.389.2 Field Documentation	395
8.389.2.1 mipMode	395
8.390unpack_wds_GetMobileIPProfile_t Struct Reference	395
8.390.1 Detailed Description	396
8.390.2 Field Documentation	396
8.390.2.1 AAASPI	396
8.390.2.2 AAASPI	396
8.390.2.3 address	396
8.390.2.4 enabled	396
8.390.2.5 HASPI	396
8.390.2.6 HASPI	396
8.390.2.7 NAI	396
8.390.2.8 naiSize	396
8.390.2.9 primaryHA	396
8.390.2.10revTunneling	396
8.390.2.11secondaryHA	396
8.391unpack_wds_GetPacketStatus_t Struct Reference	396
8.391.1 Detailed Description	397
8.391.2 Field Documentation	397
8.391.2.1 rXroppedCount	397
8.391.2.2 rXOkBytesCount	397
8.391.2.3 rXOKBytesLastCall	397
8.391.2.4 rXPacketErrors	397
8.391.2.5 rXPacketOverflows	397
8.391.2.6 rXPacketSuccesses	397
8.391.2.7 tXroppedCount	397
8.391.2.8 tXOkBytesCount	397
8.391.2.9 tXOKBytesLastCall	397
8.391.2.10tXPacketErrors	397
8.391.2.11tXPacketOverflows	397
8.391.2.12tXPacketSuccesses	398
8.392unpack_wds_GetSessionDuration_t Struct Reference	398
8.392.1 Detailed Description	398
8.392.2 Field Documentation	398
8.392.2.1 callDuration	398
8.393unpack_wds_GetSessionState_t Struct Reference	398
8.393.1 Detailed Description	398

8.393.2 Field Documentation	398
8.393.2.1 connectionStatus	398
8.394unpack_wds_RMSetTransferStatistics_t Struct Reference	398
8.395unpack_wds_SetMobileIPProfile_t Struct Reference	398
8.396unpack_wds_SLQSCreateProfile_t Struct Reference	398
8.396.1 Detailed Description	398
8.396.2 Field Documentation	399
8.396.2.1 pCreateProfileOut	399
8.396.2.2 pProfileID	399
8.396.2.3 Tlvresult	399
8.397unpack_wds_SLQSDeleteProfile_t Struct Reference	399
8.397.1 Detailed Description	399
8.397.2 Field Documentation	399
8.397.2.1 extendedErrorCode	399
8.398unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference	399
8.398.1 Detailed Description	400
8.398.2 Field Documentation	400
8.398.2.1 _3gppRelease	400
8.398.2.2 defaultPDNEnabled	400
8.398.2.3 LTEAttachProfileList	400
8.398.2.4 LTEAttachProfileListLen	400
8.398.2.5 profileList	400
8.399unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	401
8.399.1 Detailed Description	401
8.399.2 Field Documentation	401
8.399.2.1 currNetworkInfo	401
8.399.2.2 networkInfoLen	401
8.399.2.3 prefNetwork	401
8.400unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference	401
8.400.1 Detailed Description	401
8.400.2 Field Documentation	401
8.400.2.1 curDataBearerTechnology	401
8.400.2.2 dataBearerMask	401
8.400.2.3 lastCallDataBearerTechnology	401
8.401unpack_wds_SLQSGetDUNCallInfo_t Struct Reference	402
8.401.1 Detailed Description	402
8.401.2 Field Documentation	402
8.401.2.1 callEndReason	402
8.401.2.2 channelRate	402
8.401.2.3 connectionStatus	402

8.401.2.4 dataBearerTech	402
8.401.2.5 dormancyStatus	402
8.401.2.6 lastCallDataBearerTech	402
8.401.2.7 lastCallRXOKBytesCnt	402
8.401.2.8 lastCallTXOKBytesCnt	402
8.401.2.9 mdmCallDurationActive	403
8.401.2.10rxOKBytesCount	403
8.401.2.11txOKBytesCount	403
8.402unpack_wds_SLQSGetProfileSettings_t Struct Reference	403
8.402.1 Field Documentation	403
8.402.1.1 pProfileSettings	403
8.402.1.2 ProfileType	403
8.402.1.3 Tlvresult	403
8.403unpack_wds_SLQSGetRuntimeSettings_t Struct Reference	403
8.403.1 Detailed Description	403
8.403.2 Field Documentation	404
8.403.2.1 APNName	404
8.403.2.2 Authentication	404
8.403.2.3 DomainList	404
8.403.2.4 GPRSGrantedQoS	404
8.403.2.5 GWAddressV4	404
8.403.2.6 IMCNflag	404
8.403.2.7 IPFamilyPreference	404
8.403.2.8 IPv4	404
8.403.2.9 IPV6AddrInfo	404
8.403.2.10IPV6GWAddrInfo	404
8.403.2.11Mtu	404
8.403.2.12PCSCFAddrPCO	405
8.403.2.13PCSCFFQDNAddrList	405
8.403.2.14PDPTType	405
8.403.2.15PrimaryDNSV4	405
8.403.2.16PrimaryDNSV6	405
8.403.2.17ProfileID	405
8.403.2.18ProfileName	405
8.403.2.19SecondaryDNSV4	405
8.403.2.20SecondaryDNSV6	405
8.403.2.21ServerAddrList	405
8.403.2.22SubnetMaskV4	405
8.403.2.23Technology	405
8.403.2.24UMTSGrantedQoS	405

8.403.2.25Username	405
8.404unpack_wds_SLQSMModifyProfile_t Struct Reference	405
8.404.1 Detailed Description	405
8.404.2 Field Documentation	405
8.404.2.1 pExtErrorCode	405
8.405unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference	405
8.405.1 Detailed Description	405
8.405.2 Field Documentation	406
8.405.2.1 Tlvresult	406
8.406unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference	406
8.406.1 Detailed Description	406
8.406.2 Field Documentation	406
8.406.2.1 bearerID	406
8.406.2.2 conn_status	406
8.406.2.3 ipFamily	406
8.406.2.4 reconfigReqd	406
8.406.2.5 sessionEndReason	406
8.406.2.6 techName	407
8.406.2.7 verboseSessnEndReason	407
8.406.2.8 verboseSessnEndReasonType	407
8.407unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference	407
8.407.1 Detailed Description	407
8.407.2 Field Documentation	408
8.407.2.1 currDBTechAvail	408
8.407.2.2 currNWInfo	408
8.407.2.3 dataSysStatAvail	408
8.407.2.4 dBTechAvail	408
8.407.2.5 dBTechnology	408
8.407.2.6 dormancyStatAvail	408
8.407.2.7 dormancyStatus	408
8.407.2.8 mipstatAvail	408
8.407.2.9 mipStatus	408
8.407.2.10netInfoLen	408
8.407.2.11prefNetwork	408
8.407.2.12ratMask	408
8.407.2.13rx_bytes	408
8.407.2.14rx_pkts	408
8.407.2.15soMask	408
8.407.2.16tx_bytes	408
8.407.2.17tx_pkts	408

8.407.2.1&ferStatAvail	408
8.408unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	408
8.408.1 Detailed Description	408
8.408.2 Field Documentation	409
8.408.2.1 pHwConfig	409
8.408.2.2 pRequestOptionList	409
8.409unpack_wds_SLQSSStartDataSession_t Struct Reference	409
8.409.1 Detailed Description	409
8.409.2 Field Documentation	409
8.409.2.1 pFailureReason	409
8.409.2.2 psid	409
8.409.2.3 pVerboseFailReasonType	409
8.409.2.4 pVerboseFailureReason	409
8.410unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	409
8.410.1 Detailed Description	410
8.410.2 Field Documentation	410
8.410.2.1 apnName	410
8.410.2.2 bearerId	410
8.410.2.3 contextId	410
8.410.2.4 ipv4Address	410
8.410.2.5 ipv4GWAddress	410
8.410.2.6 ipv6Address	411
8.410.2.7 ipv6GWAddress	411
8.410.2.8 prDNSIPv4Address	411
8.410.2.9 prDNSIPv6Address	411
8.410.2.10prPCSCFIPv4Address	411
8.410.2.11prPCSCFIPv6Address	411
8.410.2.12seDNSIPv4Address	411
8.410.2.13seDNSIPv6Address	411
8.410.2.14sePCSCFIPv4Address	411
8.410.2.15sePCSCFIPv6Address	411
8.411UnPackGetProfileSettingOut Struct Reference	411
8.411.1 Field Documentation	411
8.411.1.1 curProfile	411
8.411.1.2 pExtErrCode	411
8.412unpackWdsProfileParam Union Reference	411
8.412.1 Field Documentation	411
8.412.1.1 SlqsProfile3GPP	411
8.412.1.2 SlqsProfile3GPP2	411
8.413wds_currNetworkInfo Struct Reference	411

8.413.1 Detailed Description	412
8.413.2 Field Documentation	413
8.413.2.1 NetworkType	413
8.413.2.2 RATMask	413
8.413.2.3 SOMask	413
8.414wds_Domain Struct Reference	413
8.414.1 Detailed Description	413
8.414.2 Field Documentation	414
8.414.2.1 domainLen	414
8.414.2.2 domainName	414
8.415wds_DomainNameList Struct Reference	414
8.415.1 Detailed Description	414
8.415.2 Field Documentation	414
8.415.2.1 domain	414
8.415.2.2 numInstances	414
8.416wds_GPRSQoS Struct Reference	414
8.416.1 Detailed Description	415
8.416.2 Field Documentation	416
8.416.2.1 delayClass	416
8.416.2.2 meanThroughputClass	416
8.416.2.3 peakThroughputClass	416
8.416.2.4 precedenceClass	416
8.416.2.5 reliabilityClass	416
8.417wds_IPV6AddressInfo Struct Reference	416
8.417.1 Detailed Description	416
8.417.2 Field Documentation	417
8.417.2.1 IPAddressV6	417
8.417.2.2 IPV6PrefixLen	417
8.418wds_IPV6GWAddressInfo Struct Reference	417
8.418.1 Detailed Description	417
8.418.2 Field Documentation	417
8.418.2.1 gwAddressV6	417
8.418.2.2 gwV6PrefixLen	417
8.419wds_PCSCFFQDNAddress Struct Reference	417
8.419.1 Detailed Description	417
8.419.2 Field Documentation	418
8.419.2.1 fqdnAddr	418
8.419.2.2 fqdnLen	418
8.420wds_PCSCFFQDNAddressList Struct Reference	418
8.420.1 Detailed Description	418

8.420.2 Field Documentation	418
8.420.2.1 numInstances	418
8.420.2.2 pcsfQDNAddress	418
8.421 wds_PCSCFIPv4ServerAddressList Struct Reference	418
8.421.1 Detailed Description	418
8.421.2 Field Documentation	419
8.421.2.1 numInstances	419
8.421.2.2 pscsfIPv4Addr	419
8.422 wds_ProfileIdentifier Struct Reference	419
8.422.1 Detailed Description	419
8.422.2 Field Documentation	419
8.422.2.1 profileIndex	419
8.422.2.2 profileType	419
8.423 wds_profileInfo Union Reference	419
8.423.1 Detailed Description	419
8.423.2 Field Documentation	419
8.423.2.1 SIqsProfile3GPP	419
8.423.2.2 SIqsProfile3GPP2	419
8.424 wds_UMTSMInQoS Struct Reference	420
8.424.1 Detailed Description	420
8.424.2 Field Documentation	422
8.424.2.1 deliveryErrSDU	422
8.424.2.2 grntDownlinkBitrate	422
8.424.2.3 grntUplinkBitrate	422
8.424.2.4 maxDownlinkBitrate	422
8.424.2.5 maxSDUSize	422
8.424.2.6 maxUplinkBitrate	422
8.424.2.7 qosDeliveryOrder	422
8.424.2.8 resBerRatio	422
8.424.2.9 sduErrorRatio	422
8.424.2.10 trafficClass	422
8.424.2.11 trafficPriority	422
8.424.2.12 transferDelay	422
8.425 wdsDhcpv4HwConfig Struct Reference	423
8.425.1 Detailed Description	423
8.425.2 Field Documentation	423
8.425.2.1 chaddr	423
8.425.2.2 chaddrLen	423
8.425.2.3 hwType	423
8.426 wdsDhcpv4Option Struct Reference	423

8.426.1 Detailed Description	423
8.426.2 Field Documentation	424
8.426.2.1 optCode	424
8.426.2.2 optVal	424
8.426.2.3 optValLen	424
8.427 wdsDhcpv4OptionList Struct Reference	424
8.427.1 Detailed Description	424
8.427.2 Field Documentation	424
8.427.2.1 numOpt	424
8.427.2.2 pOptList	424
8.428 wdsDhcpv4ProfileId Struct Reference	424
8.428.1 Detailed Description	424
8.428.2 Field Documentation	424
8.428.2.1 profileId	425
8.428.2.2 profileType	425
9 File Documentation	427
9.1 apdxyPages.c File Reference	427
9.1.1 Detailed Description	427
9.2 common.h File Reference	427
9.2.1 Macro Definition Documentation	428
9.2.1.1 DEault_LOC_TIMEOUT_IN_SEC	428
9.2.1.2 MINREQBKLEN	429
9.2.1.3 MSGID_AND_LEN	429
9.2.1.4 MSGID_DONT_CARE	429
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM	429
9.2.1.6 SDU_HDR_LEN	429
9.2.1.7 UNUSEDPARAM	429
9.2.2 Typedef Documentation	429
9.2.2.1 logger	429
9.2.3 Enumeration Type Documentation	429
9.2.3.1 eLOG_LEVEL	429
9.2.3.2 eQMI_SVC	429
9.2.3.3 eTimeout	430
9.2.3.4 msgtype	430
9.2.4 Function Documentation	430
9.2.4.1 fill_pack_ctx	430
9.2.4.2 fill_sdu_hdr	430
9.2.4.3 get_version	430
9.2.4.4 helper_get_resp_ctx	430

9.2.4.5	helper_get_xid	431
9.2.4.6	helper_set_log_func	431
9.2.4.7	helper_set_log_lvl	431
9.2.4.8	libpack_log	431
9.2.4.9	unpack_result_code_only	431
9.2.5	Variable Documentation	431
9.2.5.1	glog	431
9.2.5.2	gloglvl	431
9.3	dms.h File Reference	431
9.3.1	Macro Definition Documentation	435
9.3.1.1	DMS_IMGDETAILS_LEN	435
9.3.1.2	DMS_MAX_CUST_ID_LEN	435
9.3.1.3	DMS_MAX_CUST_VALUE_LEN	435
9.3.1.4	DMS_PM_FACTORY	435
9.3.1.5	DMS_PM_LOW	435
9.3.1.6	DMS_PM_OFFLINE	435
9.3.1.7	DMS_PM_ONLINE	435
9.3.1.8	DMS_PM_PERSISTENT_LOW	435
9.3.1.9	DMS_PM_RESET	435
9.3.1.10	DMS_PM_SHUT_DOWN	435
9.3.1.11	DMS_SET_REPORT_DISABLE	435
9.3.1.12	DMS_SET_REPORT_ENABLE	435
9.3.1.13	DMS_SWI_SET_IND_DISABLE	435
9.3.1.14	DMS_SWI_SET_IND_ENABLE	435
9.3.1.15	DMS_UINT8_MAX_STRING_SZ	435
9.3.1.16	MAX_BUILD_ID_LEN	435
9.3.1.17	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	435
9.3.1.18	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	435
9.3.1.19	SLQSFWINFO_APPVERSION_SZ	436
9.3.1.20	SLQSFWINFO_BOOTVERSION_SZ	436
9.3.1.21	SLQSFWINFO_CARRIER_SZ	436
9.3.1.22	SLQSFWINFO_CUR_CARR_NAME	436
9.3.1.23	SLQSFWINFO_CUR_CARR_REV	436
9.3.1.24	SLQSFWINFO_MODELID_SZ	436
9.3.1.25	SLQSFWINFO_PACKAGEID_SZ	436
9.3.1.26	SLQSFWINFO_PRIVERSION_SZ	436
9.3.1.27	SLQSFWINFO_SKU_SZ	436
9.3.1.28	UNIQUE_ID_LEN	436
9.3.2	Function Documentation	436
9.3.2.1	pack_dms_GetActivationState	436

9.3.2.2	pack_dms_GetBandCapability	436
9.3.2.3	pack_dms_GetCrashAction	437
9.3.2.4	pack_dms_GetCustFeature	438
9.3.2.5	pack_dms_GetCustFeaturesV2	438
9.3.2.6	pack_dms_GetDeviceCap	438
9.3.2.7	pack_dms_GetDeviceCapabilities	438
9.3.2.8	pack_dms_GetDeviceHardwareRev	439
9.3.2.9	pack_dms_GetDeviceMfr	439
9.3.2.10	pack_dms_GetDeviceSerialNumbers	439
9.3.2.11	pack_dms_GetFirmwareInfo	440
9.3.2.12	pack_dms_GetFirmwareRevision	440
9.3.2.13	pack_dms_GetFirmwareRevisions	440
9.3.2.14	pack_dms_GetFSN	441
9.3.2.15	pack_dms_GetHardwareRevision	441
9.3.2.16	pack_dms_GetIMSI	441
9.3.2.17	pack_dms_GetModelID	442
9.3.2.18	pack_dms_GetNetworkTime	442
9.3.2.19	pack_dms_GetPower	442
9.3.2.20	pack_dms_GetPRLVersion	443
9.3.2.21	pack_dms_GetSerialNumbers	443
9.3.2.22	pack_dms_GetUSBComp	443
9.3.2.23	pack_dms_GetVoiceNumber	444
9.3.2.24	pack_dms_SetCustFeature	444
9.3.2.25	pack_dms_SetCustFeaturesV2	444
9.3.2.26	pack_dms_SetEventReport	445
9.3.2.27	pack_dms_SetFirmwarePreference	445
9.3.2.28	pack_dms_SetPower	445
9.3.2.29	pack_dms_SetUSBComp	446
9.3.2.30	pack_dms_SLQSDmsSwiGetResetInfo	446
9.3.2.31	pack_dms_SLQSDmsSwiIndicationRegister	446
9.3.2.32	pack_dms_SLQSGetBandCapability	447
9.3.2.33	pack_dms_SLQSSwiClearDyingGaspStatistics	447
9.3.2.34	pack_dms_SLQSSwiGetDyingGaspCfg	447
9.3.2.35	pack_dms_SLQSSwiGetDyingGaspStatistics	448
9.3.2.36	pack_dms_SLQSSwiGetFirmwareCurr	448
9.3.2.37	pack_dms_SLQSSwiSetDyingGaspCfg	448
9.3.2.38	pack_dms_UIMGetICCID	449
9.3.2.39	unpack_dms_GetActivationState	449
9.3.2.40	unpack_dms_GetBandCapability	449
9.3.2.41	unpack_dms_GetCrashAction	450

9.3.2.42	unpack_dms_GetCustFeature	450
9.3.2.43	unpack_dms_GetCustFeaturesV2	450
9.3.2.44	unpack_dms_GetDeviceCap	450
9.3.2.45	unpack_dms_GetDeviceCapabilities	451
9.3.2.46	unpack_dms_GetDeviceHardwareRev	451
9.3.2.47	unpack_dms_GetDeviceMfr	451
9.3.2.48	unpack_dms_GetDeviceSerialNumbers	452
9.3.2.49	unpack_dms_GetFirmwareInfo	452
9.3.2.50	unpack_dms_GetFirmwareRevision	452
9.3.2.51	unpack_dms_GetFirmwareRevisions	453
9.3.2.52	unpack_dms_GetFSN	453
9.3.2.53	unpack_dms_GetHardwareRevision	453
9.3.2.54	unpack_dms_GetIMSI	454
9.3.2.55	unpack_dms_GetModelID	454
9.3.2.56	unpack_dms_GetNetworkTime	454
9.3.2.57	unpack_dms_GetPower	455
9.3.2.58	unpack_dms_GetPRLVersion	455
9.3.2.59	unpack_dms_GetSerialNumbers	455
9.3.2.60	unpack_dms_GetUSBComp	456
9.3.2.61	unpack_dms_GetVoiceNumber	456
9.3.2.62	unpack_dms_SetCustFeature	456
9.3.2.63	unpack_dms_SetCustFeaturesV2	457
9.3.2.64	unpack_dms_SetEventReport	457
9.3.2.65	unpack_dms_SetEventReport_ind	457
9.3.2.66	unpack_dms_SetFirmwarePreference	458
9.3.2.67	unpack_dms_SetPower	458
9.3.2.68	unpack_dms_SetUSBComp	458
9.3.2.69	unpack_dms_SLQSDmsSwiGetResetInfo	459
9.3.2.70	unpack_dms_SLQSDmsSwiGetResetInfo_Ind	459
9.3.2.71	unpack_dms_SLQSDmsSwiIndicationRegister	459
9.3.2.72	unpack_dms_SLQSGetBandCapability	460
9.3.2.73	unpack_dms_SLQSSwiClearDyingGaspStatistics	460
9.3.2.74	unpack_dms_SLQSSwiGetDyingGaspCfg	460
9.3.2.75	unpack_dms_SLQSSwiGetDyingGaspStatistics	461
9.3.2.76	unpack_dms_SLQSSwiGetFirmwareCurr	461
9.3.2.77	unpack_dms_SLQSSwiSetDyingGaspCfg	461
9.3.2.78	unpack_dms_UIMGetICCID	462
9.4	fms.h File Reference	462
9.4.1	Macro Definition Documentation	463
9.4.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	463

9.4.1.2	FMS_GOBI_LISTENTRIES_MAX	463
9.4.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	463
9.4.1.4	FMS_GOBI_MBN_IMG_ID_STR_LEN	463
9.4.1.5	FMS_IMAGE_ID_BUILD_ID_LEN	463
9.4.1.6	FMS_IMAGE_ID_IMG_ID_LEN	463
9.4.1.7	FMS_IMAGE_ID_MAX_ENTRIES	463
9.4.1.8	FMS_IMAGE_ID_PRI_IMGTYPE	463
9.4.1.9	FMS_MAX_IMAGE_ID_ELEMENT	463
9.4.1.10	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	463
9.4.2	Function Documentation	463
9.4.2.1	GetValidFwPriCombinations	463
9.4.2.2	pack_fms_GetImagesPreference	464
9.4.2.3	pack_fms_GetStoredImages	464
9.4.2.4	pack_fms_SetImagesPreference	464
9.4.2.5	unpack_fms_GetImagesPreference	465
9.4.2.6	unpack_fms_GetStoredImages	465
9.4.2.7	unpack_fms_SetImagesPreference	465
9.5	loc.h File Reference	465
9.5.1	Macro Definition Documentation	467
9.5.1.1	LOC_UINT8_MAX_STRING_SZ	467
9.5.1.2	LOCEVENTMASKBATCHFULLNOTIFICATION	467
9.5.1.3	LOCEVENTMASKENGINESTATE	467
9.5.1.4	LOCEVENTMASKFIXSESSIONSTATE	467
9.5.1.5	LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION	467
9.5.1.6	LOCEVENTMASKGEOFENCEBREACHNOTIFICATION	467
9.5.1.7	LOCEVENTMASKGEOFENCEGENALERT	468
9.5.1.8	LOCEVENTMASKGNSSMEASUREMENTREPORT	468
9.5.1.9	LOCEVENTMASKGNSSSVINFO	468
9.5.1.10	LOCEVENTMASKINJECTPOSITIONREQ	468
9.5.1.11	LOCEVENTMASKINJECTPREDICTEDORBITSREQ	468
9.5.1.12	LOCEVENTMASKINJECTTIMERREQ	468
9.5.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	468
9.5.1.14	LOCEVENTMASKINVALIDVALUE	468
9.5.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	468
9.5.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	468
9.5.1.17	LOCEVENTMASKMOTIONDATACONTROL	468
9.5.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	469
9.5.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	469
9.5.1.20	LOCEVENTMASKNMEA	469
9.5.1.21	LOCEVENTMASKPEDOMETERCONTROL	469

9.5.1.22	LOCEVENTMASKPOSITIONREPORT	469
9.5.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	469
9.5.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	469
9.5.1.25	LOCEVENTMASKTIMESYNCREQ	469
9.5.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	469
9.5.1.27	LOCEVENTMASKWIFIREQ	469
9.5.2	Enumeration Type Documentation	469
9.5.2.1	anonymous enum	470
9.5.3	Function Documentation	470
9.5.3.1	pack_loc_DeleteAssistData	470
9.5.3.2	pack_loc_EventRegister	470
9.5.3.3	pack_loc_SetExtPowerState	470
9.5.3.4	pack_loc_SetOperationMode	471
9.5.3.5	pack_loc_Start	471
9.5.3.6	pack_loc_Stop	471
9.5.3.7	unpack_loc_DeleteAssistData	472
9.5.3.8	unpack_loc_EngineState_Ind	472
9.5.3.9	unpack_loc_EventRegister	472
9.5.3.10	unpack_loc_PositionRpt_Ind	473
9.5.3.11	unpack_loc_SetExtPowerState	473
9.5.3.12	unpack_loc_SetOperationMode	473
9.5.3.13	unpack_loc_Start	474
9.5.3.14	unpack_loc_Stop	474
9.6	nas.h File Reference	474
9.6.1	Macro Definition Documentation	480
9.6.1.1	NAS_MAX_DESCRIPTION_LENGTH	480
9.6.1.2	NAS_MAX_NUM_NETWORKS	480
9.6.1.3	NAS_OTA_MESSAGE_MAX_BUF_SIZE	480
9.6.1.4	NAS_PLMN_LENGTH	480
9.6.1.5	NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST	480
9.6.2	Enumeration Type Documentation	480
9.6.2.1	LIBPACK_NAS_LTE_CPHY_CA_BW_NRB	480
9.6.2.2	LIBPACK_NAS_LTE_CPHY_SCELL_STATE	480
9.6.3	Function Documentation	480
9.6.3.1	pack_nas_GetACCOLC	480
9.6.3.2	pack_nas_GetANAAAAAuthenticationStatus	480
9.6.3.3	pack_nas_GetCDMANetworkParameters	481
9.6.3.4	pack_nas_GetHomeNetwork	481
9.6.3.5	pack_nas_GetNetworkPreference	481
9.6.3.6	pack_nas_GetRFInfo	482

9.6.3.7	pack_nas_GetServingNetwork	482
9.6.3.8	pack_nas_GetServingNetworkCapabilities	482
9.6.3.9	pack_nas_GetSignalStrengths	483
9.6.3.10	pack_nas_PerformNetworkScan	484
9.6.3.11	pack_nas_SetACCOLC	484
9.6.3.12	pack_nas_SetLURejectCallback	484
9.6.3.13	pack_nas_SetNetworkPreference	485
9.6.3.14	pack_nas_SetRFInfoCallback	485
9.6.3.15	pack_nas_SlqsGetLTECphyCAInfo	485
9.6.3.16	pack_nas_SLQSGetPLMNName	485
9.6.3.17	pack_nas_SLQSGetServingSystem	486
9.6.3.18	pack_nas_SLQSGetSignalStrength	486
9.6.3.19	pack_nas_SLQSGetSysInfo	486
9.6.3.20	pack_nas_SLQSGetSysSelectionPref	487
9.6.3.21	pack_nas_SLQSIInitiateNetworkRegistration	487
9.6.3.22	pack_nas_SLQSNasConfigSigInfo2	487
9.6.3.23	pack_nas_SLQSNasGetCellLocationInfo	488
9.6.3.24	pack_nas_SLQSNasGetSigInfo	488
9.6.3.25	pack_nas_SLQSNasIndicationRegisterExt	488
9.6.3.26	pack_nas_SLQSNasSwiModemStatus	489
9.6.3.27	pack_nas_SLQSNasSwiOTAMessageCallback	489
9.6.3.28	pack_nas_SLQSSetBandPreference	489
9.6.3.29	pack_nas_SLQSSetSignalStrengthsCallback	490
9.6.3.30	pack_nas_SLQSSetSysSelectionPref	490
9.6.3.31	pack_nas_SLQSSwiGetLteCQI	490
9.6.3.32	unpack_nas_GetACCOLC	491
9.6.3.33	unpack_nas_GetANAAAAuthenticationStatus	491
9.6.3.34	unpack_nas_GetCDMANetworkParameters	491
9.6.3.35	unpack_nas_GetHomeNetwork	492
9.6.3.36	unpack_nas_GetNetworkPreference	492
9.6.3.37	unpack_nas_GetRFInfo	492
9.6.3.38	unpack_nas_GetServingNetwork	492
9.6.3.39	unpack_nas_GetServingNetworkCapabilities	493
9.6.3.40	unpack_nas_GetSignalStrengths	493
9.6.3.41	unpack_nas_PerformNetworkScan	493
9.6.3.42	unpack_nas_SetACCOLC	494
9.6.3.43	unpack_nas_SetDataCapabilitiesCallback_ind	494
9.6.3.44	unpack_nas_SetEventReportInd	494
9.6.3.45	unpack_nas_SetLURejectCallback	494
9.6.3.46	unpack_nas_SetNetworkPreference	495

9.6.3.47	unpack_nas_SetRFInfoCallback	495
9.6.3.48	unpack_nas_SetRoamingIndicatorCallback_ind	495
9.6.3.49	unpack_nas_SetServingSystemCallback_ind	495
9.6.3.50	unpack_nas_SLqsGetLTECphyCAInfo	495
9.6.3.51	unpack_nas_SLQSGetPLMNName	495
9.6.3.52	unpack_nas_SLQSGetServingSystem	496
9.6.3.53	unpack_nas_SLQSGetSignalStrength	496
9.6.3.54	unpack_nas_SLQSGetSysInfo	496
9.6.3.55	unpack_nas_SLQSGetSysSelectionPref	497
9.6.3.56	unpack_nas_SLQSInitiateNetworkRegistration	497
9.6.3.57	unpack_nas_SLQSNasConfigSigInfo2	497
9.6.3.58	unpack_nas_SLQSNasGetCellLocationInfo	498
9.6.3.59	unpack_nas_SLQSNasGetSigInfo	498
9.6.3.60	unpack_nas_SLQSNasIndicationRegisterExt	498
9.6.3.61	unpack_nas_SLQSNasSigInfoCallback	499
9.6.3.62	unpack_nas_SLQSNasSwiModemStatus	499
9.6.3.63	unpack_nas_SLQSNasSwiOTAMessageCallback	499
9.6.3.64	unpack_nas_SLQSNasSwiOTAMessageCallback_ind	500
9.6.3.65	unpack_nas_SLQSNasSysInfoCallback	500
9.6.3.66	unpack_nas_SLQSSetBandPreference	500
9.6.3.67	unpack_nas_SLQSSetSignalStrengthsCallback	501
9.6.3.68	unpack_nas_SLQSSetSysSelectionPref	501
9.6.3.69	unpack_nas_SLQSSetSysSelectionPrefCallBack_ind	501
9.6.3.70	unpack_nas_SLQSSwiGetLteCQI	501
9.7	qaGobiApiTableBandClasses.h File Reference	502
9.7.1	Detailed Description	502
9.7.2	Band Classes (Value - Description)	502
9.7.2.1	LTE Bands	504
9.8	qaGobiApiTableCallControlReturnReasons.h File Reference	505
9.8.1	Detailed Description	505
9.8.2	Call Control Result Reasons (Value - Name - Description)	505
9.9	qaGobiApiTableCallEndReasons.h File Reference	506
9.9.1	Detailed Description	506
9.9.2	Call end reason codes (Code - Reason)	506
9.9.2.1	Technology-agnostic call end reasons	506
9.9.2.2	EVDO CDMA 1xEV-DO	506
9.9.2.3	WCDMA/GSM call end reasons	507
9.9.2.4	EVDO CDMA 1xEV-DO	509
9.9.2.5	call end reason type	510
9.9.2.6	Mobile IP call end reasons (Type=1)	510

9.9.2.7	Internal call end reasons (Type=2)	512
9.9.2.8	Call Manager defined call end reasons (Type=3)	513
9.9.2.9	3GPP specification defined call end reasons (Type=6)	518
9.9.2.10	PPP call end reasons (Type=7)	520
9.9.2.11	EHRPD call end reasons (Type=8)	520
9.9.2.12	IPv6 call end reasons (Type=9)	521
9.10	qaGobiApiTableCarrierCodes.h File Reference	521
9.10.1	Detailed Description	521
9.10.2	Carrier Codes (Number - Carrier)	521
9.11	qaGobiApiTableCodingScheme.h File Reference	523
9.11.1	Detailed Description	523
9.11.2	Call Control Result Reasons (Value - Name - Description)	523
9.11.2.1	Use of bits 3..0	523
9.11.3	Coding Group Bits 7..4(0001)	524
9.11.3.1	use of bits 3..0	524
9.11.4	Coding Group Bits 7..4(0010)	524
9.11.4.1	use of bits 3..0	524
9.11.5	Coding Group Bits 7..4(0011)	524
9.11.5.1	use of bits 3..0	524
9.11.6	Coding Group Bits 7..4(01xx)	525
9.11.6.1	use of bits 3..0	525
9.11.7	Coding Group Bits 7..4(1001)	525
9.11.7.1	Reserved coding groups	525
9.11.8	Coding Group Bits 7..4(1010..1101)	525
9.11.8.1	Reserved coding groups	525
9.11.9	Coding Group Bits 7..4(1110)	525
9.11.9.1	Defined by the WAP Forum	525
9.11.10	Coding Group Bits 7..4 (1111)	525
9.11.10.1	Data coding / message handling	526
9.11.11	Macro Definition Documentation	526
9.11.11.1	__GOBI_API_CODING_SCHEME_H__	526
9.12	qaGobiApiTableGpsCapabilityCodes.h File Reference	526
9.12.1	Detailed Description	526
9.12.2	GPS capability (Value - Capability)	526
9.13	qaGobiApiTablePowerModes.h File Reference	526
9.13.1	Detailed Description	526
9.13.2	Power Modes (Value - Description)	527
9.14	qaGobiApiTableRadioInterfaces.h File Reference	527
9.14.1	Detailed Description	527
9.14.2	Radio interface	527

9.14.2.1 Technology (Value - Radio Interface Technology)	527
9.15 qaGobiApiTableRegionCodes.h File Reference	527
9.15.1 Detailed Description	528
9.15.2 Region Codes (Code - Region)	528
9.16 qaGobiApiTableServiceOptions.h File Reference	528
9.16.1 Detailed Description	528
9.16.2 Service Option codes (Code - Reason)	528
9.16.2.1 Description	528
9.17 qaGobiApiTableSupServiceInfoClasses.h File Reference	530
9.17.1 Detailed Description	530
9.17.2 Supplementary Service Information Classes (Value - Service Class)	531
9.18 qaGobiApiTableSwiAudio.h File Reference	531
9.18.1 Detailed Description	531
9.18.2 ACDB Device (Device ID - description)	531
9.18.3 Physical Interface (Device ID - description - Interface parameters)	531
9.19 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	531
9.19.1 Detailed Description	532
9.19.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	532
9.20 qaGobiApiTableVoiceCallEndReasons.h File Reference	533
9.20.1 Detailed Description	533
9.20.2 Voice Call and supplementary services end reason codes (Code - Reason)	533
9.20.2.1 General	533
9.20.2.2 service Errors	535
9.20.2.3 control cause values	536
9.20.2.4 reject causes	537
9.20.2.5 reject causes	538
9.20.2.6 reject causes	538
9.20.2.7 stratum reject causes	538
9.20.2.8 reject causes	539
9.20.2.9 IP end reasons	539
9.21 qmerrno.h File Reference	539
9.21.1 Enumeration Type Documentation	541
9.21.1.1 eQCWWANError	541
9.21.1.2 qm_wds_ds_profile_extended_err_codes	546
9.22 qos.h File Reference	546
9.22.1 Macro Definition Documentation	548
9.22.1.1 LIBPACK_MAX_QOS_FILTERS	548
9.22.1.2 LIBPACK_MAX_QOS_FLOW_PER_APN_STATS	548
9.22.1.3 LIBPACK_MAX_QOS_FLOWS	548
9.22.2 Function Documentation	548

9.22.2.1	pack_qos_SLQSQosGetNetworkStatus	548
9.22.2.2	pack_qos_SLQSQosSwiReadApnExtraParams	548
9.22.2.3	pack_qos_SLQSQosSwiReadDataStats	549
9.22.2.4	pack_qos_SLQSSetQosEventCallback	550
9.22.2.5	unpack_qos_SLQSQosGetNetworkStatus	551
9.22.2.6	unpack_qos_SLQSQosSwiReadApnExtraParams	552
9.22.2.7	unpack_qos_SLQSQosSwiReadDataStats	552
9.22.2.8	unpack_qos_SLQSSetQosEventCallback	553
9.22.2.9	unpack_qos_SLQSSetQosEventCallback_ind	553
9.22.2.10	unpack_qos_SLQSSetQosNWStatusCallback_ind	554
9.22.2.11	unpack_qos_SLQSSetQosPriEventCallback_ind	555
9.22.2.12	unpack_qos_SLQSSetQosStatusCallback_ind	556
9.23	sms.h File Reference	557
9.23.1	Macro Definition Documentation	558
9.23.1.1	MAX_CDMA_ENC_MO_TXT_MSG_SIZE	558
9.23.1.2	MAX_MS_TRANSFER_ROUTE_MSG	558
9.23.1.3	MAX_MSC_ADDRESS_SIZE	558
9.23.1.4	MAX_MSE_TWS_MSG	559
9.23.1.5	MAX_SMS_LIST_SIZE	559
9.23.1.6	MAX_SMS_MESSAGE_SIZE	559
9.23.2	Typedef Documentation	559
9.23.2.1	sMSCAddressInfo	559
9.23.2.2	sMSEtwSMessageInfo	559
9.23.2.3	sMSEtwSPImnInfo	559
9.23.2.4	sMSMessageModelInfo	559
9.23.2.5	sMSMTMessageInfo	560
9.23.2.6	sMSOnIMSInfo	560
9.23.2.7	sMSTransferRouteMTMessageInfo	560
9.23.3	Enumeration Type Documentation	560
9.23.3.1	eqmiCbKSetStatus	560
9.23.4	Function Documentation	560
9.23.4.1	pack_sms_SendSMS	561
9.23.4.2	pack_sms_SetNewSMSCallback	562
9.23.4.3	pack_sms_SLQSDelSms	562
9.23.4.4	pack_sms_SLQSGetSMS	562
9.23.4.5	pack_sms_SLQSGetSMSList	563
9.23.4.6	pack_sms_SLQSModifySMSStatus	563
9.23.4.7	unpack_sms_SendSMS	563
9.23.4.8	unpack_sms_SetNewSMSCallback	564
9.23.4.9	unpack_sms_SetNewSMSCallback_ind	564

9.23.4.10	unpack_sms_SLQSDeleteSMS	564
9.23.4.11	unpack_sms_SLQSGetSMS	565
9.23.4.12	unpack_sms_SLQSGetSMSList	565
9.23.4.13	unpack_sms_SLQSModifySMSStatus	565
9.23.4.14	unpack_sms_SLQSWmsMemoryFullCallBack_ind	566
9.24	SwiDataTypes.h File Reference	566
9.24.1	Detailed Description	567
9.24.2	Macro Definition Documentation	567
9.24.2.1	QMI_NO_LTE_FW_SUPPORT	567
9.24.2.2	QMI_TLV_PLACEHOLDER	567
9.24.2.3	SWI_API	567
9.24.2.4	UNUSEDPARAM	567
9.24.3	Typedef Documentation	567
9.24.3.1	BOOL	567
9.24.3.2	BYTE	567
9.24.3.3	CHAR	567
9.24.3.4	FLOAT	567
9.24.3.5	INT32	567
9.24.3.6	INT8	567
9.24.3.7	LPCSTR	567
9.24.3.8	SHORT	567
9.24.3.9	ULONG	567
9.24.3.10	ULONGLONG	567
9.24.3.11	USHORT	567
9.24.3.12	WORD	567
9.25	swiloc.h File Reference	567
9.25.1	Function Documentation	568
9.25.1.1	pack_swiloc_SwiLocGetAutoStart	568
9.25.1.2	pack_swiloc_SwiLocSetAutoStart	568
9.25.1.3	unpack_swiloc_SwiLocGetAutoStart	568
9.25.1.4	unpack_swiloc_SwiLocSetAutoStart	569
9.26	swioma.h File Reference	569
9.26.1	Macro Definition Documentation	570
9.26.1.1	LIBPACK_MAX_SWIOMA_STR_LEN	570
9.26.2	Function Documentation	570
9.26.2.1	pack_swioma_SLQSOMADMAAlertCallback	570
9.26.2.2	pack_swioma_SLQSOMADMCancelSession	571
9.26.2.3	pack_swioma_SLQSOMADMGetSessionInfo	571
9.26.2.4	pack_swioma_SLQSOMADMGetSettings	572
9.26.2.5	pack_swioma_SLQSOMADMSendSelection	573

9.26.2.6	pack_swima_SLQSOMADMSetSettings	573
9.26.2.7	pack_swima_SLQSOMADMStartSession	574
9.26.2.8	unpack_swima_SLQSOMADMAAlertCallback	575
9.26.2.9	unpack_swima_SLQSOMADMAAlertCallback_ind	575
9.26.2.10	unpack_swima_SLQSOMADMCancelSession	576
9.26.2.11	unpack_swima_SLQSOMADMGetSessionInfo	576
9.26.2.12	unpack_swima_SLQSOMADMGetSettings	577
9.26.2.13	unpack_swima_SLQSOMADMSendSelection	577
9.26.2.14	unpack_swima_SLQSOMADMSetSettings	578
9.26.2.15	unpack_swima_SLQSOMADMStartSession	578
9.27	SWIWWANCMAPI.h File Reference	579
9.28	uim.h File Reference	579
9.28.1	Macro Definition Documentation	580
9.28.1.1	MAX_DESCRIPTION_LENGTH	580
9.28.1.2	MAX_ICCID_LENGTH	580
9.28.1.3	MAX_NO_OF_APPLICATIONS	580
9.28.1.4	MAX_NO_OF_SLOTS	580
9.28.1.5	MAX_SLOTS_STATUS	580
9.28.1.6	UIM_MAX_DESCRIPTION_LENGTH	580
9.28.1.7	UIM_MAX_NO_OF_APPLICATIONS	580
9.28.1.8	UIM_MAX_NO_OF_SLOTS	580
9.28.1.9	UIM_UINT8_MAX_STRING_SZ	580
9.28.2	Function Documentation	581
9.28.2.1	pack_uim_ChangePin	581
9.28.2.2	pack_uim_GetCardStatus	581
9.28.2.3	pack_uim_ReadTransparent	581
9.28.2.4	pack_uim_SetPinProtection	582
9.28.2.5	pack_uim_SLQSUIEventRegister	582
9.28.2.6	pack_uim_SLQSUIMGetSlotsStatus	582
9.28.2.7	pack_uim_SLQSUIMSwitchSlot	583
9.28.2.8	pack_uim_UnblockPin	584
9.28.2.9	pack_uim_VerifyPin	584
9.28.2.10	unpack_uim_ChangePin	584
9.28.2.11	unpack_uim_GetCardStatus	585
9.28.2.12	unpack_uim_ReadTransparent	585
9.28.2.13	unpack_uim_SetPinProtection	585
9.28.2.14	unpack_uim_SetUimSlotStatusChangeCallback_ind	586
9.28.2.15	unpack_uim_SLQSUIEventRegister	586
9.28.2.16	unpack_uim_SLQSUIMGetSlotsStatus	586
9.28.2.17	unpack_uim_SLQSUIMSetStatusChangeCallBack_ind	587

9.28.2.18 unpack_uim_SLQSUIMSwitchSlot	587
9.28.2.19 unpack_uim_UnblockPin	587
9.28.2.20 unpack_uim_VerifyPin	588
9.29 wds.h File Reference	588
9.29.1 Macro Definition Documentation	592
9.29.1.1 IPV6_ADDRESS_ARRAY_SIZE	592
9.29.1.2 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	592
9.29.1.3 PACK_WDS_IPV4	592
9.29.1.4 PACK_WDS_IPV6	592
9.29.2 Typedef Documentation	592
9.29.2.1 UnpackQmiProfileInfo	592
9.29.3 Function Documentation	592
9.29.3.1 pack_wds_GetConnectionRate	592
9.29.3.2 pack_wds_GetDefaultProfile	593
9.29.3.3 pack_wds_GetDefaultProfileNum	593
9.29.3.4 pack_wds_GetDormancyState	594
9.29.3.5 pack_wds_GetLastMobileIPError	595
9.29.3.6 pack_wds_GetMobileIP	595
9.29.3.7 pack_wds_GetMobileIPProfile	596
9.29.3.8 pack_wds_GetPacketStatus	596
9.29.3.9 pack_wds_GetSessionDuration	596
9.29.3.10 pack_wds_GetSessionState	597
9.29.3.11 pack_wds_RMSetTransferStatistics	597
9.29.3.12 pack_wds_SetDefaultProfile	598
9.29.3.13 pack_wds_SetDefaultProfileNum	599
9.29.3.14 pack_wds_SetMobileIPProfile	599
9.29.3.15 pack_wds_SLQSCreateProfile	600
9.29.3.16 pack_wds_SLQSDeleteProfile	600
9.29.3.17 pack_wds_SLQSGet3GPPConfigItem	600
9.29.3.18 pack_wds_SLQSGetCurrDataSystemStat	601
9.29.3.19 pack_wds_SLQSGetDataBearerTechnology	601
9.29.3.20 pack_wds_SLQSGetDUNCallInfo	602
9.29.3.21 pack_wds_SLQSGetProfileSettings	602
9.29.3.22 pack_wds_SLQSGetRuntimeSettings	602
9.29.3.23 pack_wds_SLQSModifyProfile	603
9.29.3.24 pack_wds_SLQSSet3GPPConfigItem	603
9.29.3.25 pack_wds_SLQSSetIPFamilyPreference	604
9.29.3.26 pack_wds_SLQSSetWdsEventCallback	604
9.29.3.27 pack_wds_SLQSSetDHCPv4ClientConfig	604
9.29.3.28 pack_wds_SLQSStartDataSession	605

9.29.3.29 pack_wds_SLQSStopDataSession	605
9.29.3.30 pack_wds_SLQSWdsSwiPDPRuntimeSettings	606
9.29.3.31 unpack_wds_GetConnectionRate	606
9.29.3.32 unpack_wds_GetDefaultProfile	606
9.29.3.33 unpack_wds_GetDefaultProfileNum	607
9.29.3.34 unpack_wds_GetDormancyState	607
9.29.3.35 unpack_wds_GetLastMobileIPError	607
9.29.3.36 unpack_wds_GetMobileIP	608
9.29.3.37 unpack_wds_GetMobileIPProfile	608
9.29.3.38 unpack_wds_GetPacketStatus	608
9.29.3.39 unpack_wds_GetSessionDuration	609
9.29.3.40 unpack_wds_GetSessionState	609
9.29.3.41 unpack_wds_RMSetTransferStatistics	609
9.29.3.42 unpack_wds_SetDefaultProfile	610
9.29.3.43 unpack_wds_SetDefaultProfileNum	610
9.29.3.44 unpack_wds_SetMobileIPProfile	610
9.29.3.45 unpack_wds_SLQSCreateProfile	611
9.29.3.46 unpack_wds_SLQSDeleteProfile	611
9.29.3.47 unpack_wds_SLQSGet3GPPConfigItem	611
9.29.3.48 unpack_wds_SLQSGetCurrDataSystemStat	612
9.29.3.49 unpack_wds_SLQSGetDataBearerTechnology	612
9.29.3.50 unpack_wds_SLQSGetDUNCallInfo	612
9.29.3.51 unpack_wds_SLQSGetProfileSettings	613
9.29.3.52 unpack_wds_SLQSGetRuntimeSettings	613
9.29.3.53 unpack_wds_SLQSModifyProfile	613
9.29.3.54 unpack_wds_SLQSSet3GPPConfigItem	614
9.29.3.55 unpack_wds_SLQSSetIPFamilyPreference	614
9.29.3.56 unpack_wds_SLQSSetPacketSrvStatusCallback	614
9.29.3.57 unpack_wds_SLQSSetWdsEventCallback	615
9.29.3.58 unpack_wds_SLQSSetWdsEventCallback_ind	615
9.29.3.59 unpack_wds_SLQSSetDHCPv4ClientConfig	615
9.29.3.60 unpack_wds_SLQSStartDataSession	616
9.29.3.61 unpack_wds_SLQSStopDataSession	617
9.29.3.62 unpack_wds_SLQSWdsSwiPDPRuntimeSettings	617

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 (pack)	17
----------------------------------	----

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tables	19
----------------------------------	----

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

appStats	21
CarrierImage_t	25
cdmaSSInfo	26
connectionStatus	26
currNetworkInfo	27
dms_ActivationStatusTlv	27
dms_OperatingModeTlv	28
DMScustSettingInfo	29
DMScustSettingList	30
DMSgetCustomFeatureV2	30
DMSgetCustomInput	31
dunchannelRate	31
eTWSPLMNInfoTlv	32
FMSImageElement	32
FMSImageIdElement	33
FMSImageIdEntries	34
FMSImageList	35
FMSPrefImageList	36
hdrSSInfo	36
image_info_t	37
ipv6AddressInfo	37
LibPackGPRSRequestedQoS	38
LibpackProfile3GPP	38
LibpackProfile3GPP2	45
LibPackprofile_3GPP	51
LibPackprofile_3GPP2	57
LibPackQosClassID	63
LibPackTFTIDParams	64
LibPackUMTSQoS	66
LibPackUMTSReqQoSSigInd	69
loc_BdsSV	70
loc_BdsSVInfo	70
loc_CellDb	71
loc_ClkInfo	72
loc_GnssData	74
loc_gpsTime	76
loc_LocApplicationInfo	76
loc_precisionDilution	77

loc_sensorDataUsage	78
loc_SV	79
loc_SVInfo	80
loc_svUsedforFix	81
lteSSInfo	81
messageModeTlv	82
nas_acqOrderPref	82
nas_AddCDMASysInfo	83
nas_AddSysInfo	84
nas_CallBarringSysInfo	84
nas_callBarStatus	85
nas_CDMAECIOThresh	86
nas_CDMAInfo	87
nas_CDMARSSIThresh	88
nas_CDMASysInfo	89
nas_CDMASysInfoExt	93
nas_cellParams	94
nas_CommInfo	95
nas_CSGID	97
nas_currentPLMN	98
nas_dataSrvCapabilities	99
nas_detailSvcInfo	100
nas_ecioListElement	102
nas_errorRateListElement	103
nas_GERANInfo	104
nas_geranInstInfo	106
nas_gsmCellInfo	107
nas_GSMRSSIThresh	108
nas_GSMSrvStatusInfo	109
nas_GSMSysInfo	110
nas_HDRECIOThresh	113
nas_HDRIOThresh	114
nas_HDRRSSIThresh	114
nas_HDRSINRThreshold	115
nas_HDRSysInfo	116
nas_infoInterFreq	119
nas_lteGsmCellInfo	120
nas_LTEInfo	121
nas_LTEInfoInterfreq	124
nas_LTEInfoIntrafreq	124
nas_LTEInfoNeighboringGSM	127
nas_LTEInfoNeighboringWCDMA	128
nas_lteRsrpInformation	128
nas_LTERSRPThresh	129
nas_LTERSQThresh	129
nas_LTERSSIThresh	130
nas_LTESigRptConfig	130
nas_lteSnrinformation	132
nas_LTESNRThreshold	132
nas_LTESysInfo	133
nas_lteWcdmaCellInfo	136
nas_MNRInfo	137
nas_netSelectionPref	138
nas_nmrCellInfo	138
nas_qaQmi3Gpp2TimeZone	141
nas_QmiNas3GppNetworkInfo	141
nas_QmiNas3GppNetworkRAT	142
nas_QmisNasPcsDigit	143

nas_RejectReasonTlv	144
nas_RFInfoTlv	144
nas_roamIndList	145
nas_rsrqInformation	146
nas_rxSignalStrengthListElement	146
nas_servSystem	147
nas_SignalStrengthTlv	149
nas_SLQSSignalStrengthsIndReq	150
nas_SLQSSignalStrengthsInformation	151
nas_SLQSSignalStrengthsTlv	152
nas_SrvStatusInfo	152
nas_sysInfoCommon	153
nas_TDSCDMAECIOThresh	156
nas_TDSCDMARSCPTthresh	157
nas_TDSCDMARSSIThresh	157
nas_TDSCDMASINRThresh	158
nas_UMTSInfo	158
nas_UMTSinstInfo	160
nas_umtsLTENbrCell	162
nas_wcdmaCellInfo	163
nas_WCDMAECIOThresh	164
nas_WCDMAInfoLTENeighborCell	165
nas_WCDMARSSIThresh	166
nas_WCDMASysInfo	166
NASBandPreferenceTlv	170
NASEmergencyModeTlv	171
NasGetLTECphyCaInfo	171
NASGWAcqOrderPrefTlv	171
NASLTEBandPreferenceTlv	172
NASLteNasReleaseInfoTlv	172
NASModePreferenceTlv	172
NASNetSelPreferenceTlv	173
NASOTAMessageTlv	173
NASPhyCaAggPcellInfo	173
NASPhyCaAggScellIDIBw	174
NASPhyCaAggScellIndex	175
NASPhyCaAggScellIndType	175
NASPhyCaAggScellInfo	176
NASPRLPreferenceTlv	177
NASQmiCbkNasSwiOTAMessageInd	177
NASQmiCbkNasSystemSelPrefInd	178
NASRoamPreferenceTlv	178
NASServDomainPrefTlv	179
NASServingSystemInfo	179
NASTimeInfoTlv	181
newMTMessageTlv	181
pack_dms_GetCustFeaturesV2_t	181
pack_dms_SetCustFeature_t	182
pack_dms_SetCustFeaturesV2_t	183
pack_dms_SetEventReport_t	183
pack_dms_SetPower_t	184
pack_dms_SetUSBComp_t	184
pack_dms_SLQSDmsSwiIndicationRegister_t	184
pack_dms_SLQSSwiSetDyingGaspCfg_t	185
pack_dms_UIMGetICCID_t	185
pack_fms_GetImagesPreference_t	185
pack_fms_GetStoredImages_t	186
pack_fms_SetImagesPreference_t	186

pack_loc_Delete_Assist_Data_t	187
pack_loc_EventRegister_t	188
pack_loc_SetExtPowerState_t	190
pack_loc_SetOperationMode_t	190
pack_loc_Start_t	191
pack_loc_Stop_t	193
pack_nas_SetACCOLC_t	193
pack_nas_SetNetworkPreference_t	195
pack_nas_SLQSGetPLMNName_t	196
pack_nas_SLQSInitiateNetworkRegistration_t	197
pack_nas_SLQSNasConfigSigInfo2_t	198
pack_nas_SLQSNasIndicationRegisterExt_t	203
pack_nas_SLQSNasSwiOTAMessageCallback_t	207
pack_nas_SLQSSetSignalStrengthsCallback_t	208
pack_nas_SLQSSetSysSelectionPref_t	209
pack_qmi_t	213
pack_qos_SLQSQosSwiReadApnExtraParams_t	214
pack_qos_SLQSQosSwiReadDataStats_t	214
pack_qos_SLQSSetQosEventCallback_t	214
pack_sms_SendSMS_t	216
pack_sms_SetNewSMSCallback_t	217
pack_sms_SLQSDeleteSMS_t	217
pack_sms_SLQSGetSMS_t	218
pack_sms_SLQSGetSMSList_t	219
pack_sms_SLQSModifySMSStatus_t	220
pack_swiloc_SwiLocSetAutoStart_t	221
pack_swioama_SLQSOMADMCancelSession_t	223
pack_swioama_SLQSOMADMGetSessionInfo_t	223
pack_swioama_SLQSOMADMSelectSendSelection_t	224
pack_swioama_SLQSOMADMSetSettings_t	225
pack_swioama_SLQSOMADMStartSession_t	226
pack_uim_ChangePin_t	226
pack_uim_ReadTransparent_t	227
pack_uim_SetPinProtection_t	229
pack_uim_SLQSUIMEventRegister_t	230
pack_uim_SLQSUIMSwitchSlot_t	230
pack_uim_UnblockPin_t	231
pack_uim_VerifyPin_t	232
pack_wds_GetDefaultProfile_t	234
pack_wds_GetDefaultProfileNum_t	235
pack_wds_GetDormancyState_t	235
pack_wds_GetLastMobileIPError_t	235
pack_wds_GetMobileIP_t	235
pack_wds_GetMobileIPProfile_t	235
pack_wds_GetPacketStatus_t	237
pack_wds_GetSessionDuration_t	237
pack_wds_RMSetTransferStatistics_t	237
pack_wds_SetDefaultProfile_t	237
pack_wds_SetDefaultProfileNum_t	238
pack_wds_SetMobileIPProfile_t	239
pack_wds_SLQSCreateProfile_t	240
pack_wds_SLQSDeleteProfile_t	241
pack_wds_SLQSGetCurrDataSystemStat_t	241
pack_wds_SLQSGetDataBearerTechnology_t	241
pack_wds_SLQSGetDUNCallInfo_t	241
pack_wds_SLQSGetProfileSettings_t	242
pack_wds_SLQSGetRuntimeSettings_t	243
pack_wds_SLQSModifyProfile_t	244

pack_wds_SLQSSet3GPPConfigItem_t	244
pack_wds_SLQSSetIPFamilyPreference_t	245
pack_wds_SLQSSetWdsEventCallback_t	246
pack_wds_SLQSSetDHCPv4ClientConfig_t	247
pack_wds_SLQSSetDataSession_t	247
pack_wds_SLQSSetDataSession_t	248
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	249
PackCreateProfileOut	249
packgetDyingGaspCfg	249
packgetDyingGaspStatistics	250
qmiSmsMessageList	250
qmiWSDDataBearerTechnology	251
RFBandInfoElements	251
rmTransferStaticsReq	252
slot_t	252
slotInf	253
slots_t	256
sMSCAddress	256
sMSCAddressTlv	257
sMSEtwsMessage	257
sMSEtwsMessageTlv	258
sMSEtwsPlmn	258
sMSMessageMode	259
sMSMTMessage	259
sMSOnIMS	260
sMSOnIMSTlv	260
sMSTransferRouteMTMessage	260
tdscdmaSigInfoExt	262
transferRouteMessageTlv	263
transferStatInd	263
uim_appStatus	264
uim_cardResult	267
uim_cardStatus	268
uim_changeUIMPIN	269
uim_encryptedPIN1	270
uim_fileInfo	271
uim_hotSwapStatus	272
uim_readResult	272
uim_readTransparentInfo	273
uim_remainingRetries	273
uim_sessionInformation	274
uim_setPINProtection	275
uim_slotInfo	276
uim_UIMSessionInformation	278
uim_unblockUIMPIN	279
uim_verifyUIMPIN	280
unpack_dms_GetActivationState_t	281
unpack_dms_GetBandCapability_t	281
unpack_dms_GetCrashAction_t	282
unpack_dms_GetCustFeature_t	282
unpack_dms_GetCustFeaturesV2_t	283
unpack_dms_GetDeviceCap_t	283
unpack_dms_GetDeviceCapabilities_t	284
unpack_dms_GetDeviceHardwareRev_t	284
unpack_dms_GetDeviceMfr_t	285
unpack_dms_GetDeviceSerialNumbers_t	285
unpack_dms_GetFirmwareInfo_t	286
unpack_dms_GetFirmwareRevision_t	287

unpack_dms_GetFirmwareRevisions_t	287
unpack_dms_GetFSN_t	288
unpack_dms_GetHardwareRevision_t	288
unpack_dms_GetIMSI_t	288
unpack_dms_GetModelID_t	289
unpack_dms_GetNetworkTime_t	289
unpack_dms_GetPower_t	289
unpack_dms_GetPRLVersion_t	290
unpack_dms_GetSerialNumbers_t	290
unpack_dms_GetUSBComp_t	291
unpack_dms_GetVoiceNumber_t	291
unpack_dms_SetCustFeature_t	292
unpack_dms_SetCustFeaturesV2_t	292
unpack_dms_SetEventReport_ind_t	292
unpack_dms_SetEventReport_t	293
unpack_dms_SetFirmwarePreference_t	293
unpack_dms_SetPower_t	293
unpack_dms_SetUSBComp_t	294
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t	294
unpack_dms_SLQSDmsSwiGetResetInfo_t	295
unpack_dms_SLQSDmsSwiIndicationRegister_t	296
unpack_dms_SLQSGetBandCapability_t	296
unpack_dms_SLQSSwiClearDyingGaspStatistics_t	298
unpack_dms_SLQSSwiGetDyingGaspCfg_t	298
unpack_dms_SLQSSwiGetDyingGaspStatistics_t	300
unpack_dms_SLQSSwiGetFirmwareCurr_t	300
unpack_dms_SLQSSwiSetDyingGaspCfg_t	301
unpack_dms_UIMGetICCID_t	302
unpack_fms_GetImagesPreference_t	302
unpack_fms_GetStoredImages_t	303
unpack_fms_SetImagesPreference_t	304
unpack_loc_Delete_Assist_Data_t	304
unpack_loc_EngineState_Ind_t	305
unpack_loc_EventRegister_t	305
unpack_loc_PositionRpt_Ind_t	306
unpack_loc_SetExtPowerState_t	312
unpack_loc_SetOperationMode_t	312
unpack_loc_Start_t	313
unpack_loc_Stop_t	313
unpack_nas_GetCDMANetworkParameters_t	314
unpack_nas_GetHomeNetwork_t	315
unpack_nas_GetNetworkPreference_t	315
unpack_nas_GetRFInfo_t	317
unpack_nas_GetServingNetwork_t	318
unpack_nas_GetServingNetworkCapabilities_t	319
unpack_nas_GetSignalStrengths_t	320
unpack_nas_PerformNetworkScan_t	320
unpack_nas_SetDataCapabilitiesCallback_ind_t	321
unpack_nas_SetEventReportInd_t	321
unpack_nas_SetNetworkPreference_t	322
unpack_nas_SetRoamingIndicatorCallback_ind_t	323
unpack_nas_SetServingSystemCallback_ind_t	323
unpack_nas_SLqsGetLTECphyCAInfo_t	324
unpack_nas_SLQSGetPLMNName_t	324
unpack_nas_SLQSGetServingSystem_t	325
unpack_nas_SLQSGetSignalStrength_t	328
unpack_nas_SLQSGetSysInfo_t	329
unpack_nas_SLQSGetSysSelectionPref_t	331

unpack_nas_SLQSNasGetCellLocationInfo_t	335
unpack_nas_SLQSNasGetSigInfo_t	337
unpack_nas_SLQSNasSigInfoCallback_t	337
unpack_nas_SLQSNasSwiModemStatus_t	338
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t	339
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t	339
unpack_nas_SLQSSwiGetLteCQI_t	340
unpack_nas_SLQSSysInfoCallback_t	341
unpack_omaDmConfigTlv_t	344
unpack_omaDmFotaTlv_t	345
unpack_omaDmNotificationsTlv_t	347
unpack_qmi_t	348
unpack_qos_dataRate_t	348
unpack_qos_IPv4Addr_t	349
unpack_qos_IPv6Addr_t	349
unpack_qos_IPv6TrafCls_t	349
unpack_qos_pktErrRate_t	350
unpack_qos_Port_t	350
unpack_qos_QosFlowInfo_t	351
unpack_qos_QosFlowInfoState_t	353
unpack_qos_SLQSQosGetNetworkStatus_t	353
unpack_qos_SLQSQosSwiReadApnExtraParams_t	355
unpack_qos_SLQSQosSwiReadDataStats_t	356
unpack_qos_SLQSSetQosEventCallback_ind_t	358
unpack_qos_SLQSSetQosNWStatusCallback_ind_t	358
unpack_qos_SLQSSetQosPriEventCallback_ind_t	360
unpack_qos_SLQSSetQosStatusCallback_ind_t	360
unpack_qos_swiQosFilter_t	362
unpack_qos_swiQosFlow_t	365
unpack_qos_tokenBucket_t	370
unpack_qos_Tos_t	371
unpack_QosFlowStat_t	371
unpack_sms_SendSMS_t	372
unpack_sms_SetNewSMSCallback_ind_t	373
unpack_sms_SetNewSMSCallback_t	374
unpack_sms_SLQSDeleteSMS_t	374
unpack_sms_SLQSGetSMS_t	374
unpack_sms_SLQSGetSMSList_t	375
unpack_sms_SLQSModifySMSStatus_t	376
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	376
unpack_swiloc_SwiLocGetAutoStart_t	376
unpack_swioma_SLQSOMADMAAlertCallback_ind_t	378
unpack_swioma_SLQSOMADMGetSessionInfo_t	379
unpack_swioma_SLQSOMADMGetSettings_t	382
unpack_swioma_SLQSOMADMStartSession_t	384
unpack_uim_ChangePin_t	384
unpack_uim_GetCardStatus_t	385
unpack_uim_ReadTransparent_t	386
unpack_uim_SetPinProtection_t	387
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	388
unpack_uim_SLQSUIMEventRegister_t	388
unpack_uim_SLQSUIMGetSlotsStatus_t	389
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t	389
unpack_uim_UnblockPin_t	390
unpack_uim_VerifyPin_t	390
unpack_wds_GetConnectionRate_t	392
unpack_wds_GetDefaultProfile_t	393
unpack_wds_GetDefaultProfileNum_t	394

unpack_wds_GetDormancyState_t	394
unpack_wds_GetLastMobileIPError_t	395
unpack_wds_GetMobileIP_t	395
unpack_wds_GetMobileIPProfile_t	395
unpack_wds_GetPacketStatus_t	396
unpack_wds_GetSessionDuration_t	398
unpack_wds_GetSessionState_t	398
unpack_wds_RMSetTransferStatistics_t	398
unpack_wds_SetMobileIPProfile_t	398
unpack_wds_SLQSCreateProfile_t	398
unpack_wds_SLQSDeleteProfile_t	399
unpack_wds_SLQSGet3GPPConfigItem_t	399
unpack_wds_SLQSGetCurrDataSystemStat_t	401
unpack_wds_SLQSGetDataBearerTechnology_t	401
unpack_wds_SLQSGetDUNCallInfo_t	402
unpack_wds_SLQSGetProfileSettings_t	403
unpack_wds_SLQSGetRuntimeSettings_t	403
unpack_wds_SLQSModifyProfile_t	405
unpack_wds_SLQSSetIPFamilyPreference_t	405
unpack_wds_SLQSSetPacketSrvStatusCallback_t	406
unpack_wds_SLQSSetWdsEventCallback_ind_t	407
unpack_wds_SLQSSetDHCPv4ClientConfig_t	408
unpack_wds_SLQSStartDataSession_t	409
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	409
UnPackGetProfileSettingOut	411
unpackWdsProfileParam	411
wds_currNetworkInfo	411
wds_Domain	413
wds_DomainNameList	414
wds_GPRSQoS	414
wds_IPV6AddressInfo	416
wds_IPV6GWAddressInfo	417
wds_PCSCFFQDNAddress	417
wds_PCSCFFQDNAddressList	418
wds_PCSCFIPv4ServerAddressList	418
wds_ProfileIdentifier	419
wds_profileInfo	419
wds_UMTSMInQoS	420
wdsDhcpv4HwConfig	423
wdsDhcpv4Option	423
wdsDhcpv4OptionList	424
wdsDhcpv4ProfileId	424

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	427
common.h		427
dms.h		431
fms.h		462
loc.h		465
nas.h		474
qaGobiApiTableBandClasses.h	Network Access Service API Band Classes table	502
qaGobiApiTableCallControlReturnReasons.h	Call Control Return Reasons table	505
qaGobiApiTableCallEndReasons.h	Wireless Data Service Call End Reasons	506
qaGobiApiTableCarrierCodes.h	Carrier Codes table	521
qaGobiApiTableCodingScheme.h	Data Coding Scheme	523
qaGobiApiTableGpsCapabilityCodes.h	Position Determination Service API GPS Capability Codes	526
qaGobiApiTablePowerModes.h	Device Management Service API Power Modes table	526
qaGobiApiTableRadiolInterfaces.h	Network Access Service API Radio Interfaces table	527
qaGobiApiTableRegionCodes.h	Region Codes table	527
qaGobiApiTableServiceOptions.h	Voice Service Options	528
qaGobiApiTableSupServiceInfoClasses.h	Voice Supplementary Service Information Classes	530
qaGobiApiTableSwiAudio.h	Swi Audio related tables	531
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h	Update Complete Status table	531
qaGobiApiTableVoiceCallEndReasons.h	Voice Service Call and supplementary services end reasons	533
qmerrno.h		539
qos.h		546

sms.h	557
SwiDataTypes.h	
SWI data types	566
swiloc.h	567
swioma.h	569
SWIWWANCMAPI.h	579
uim.h	579
wds.h	588

Chapter 6

Module Documentation

6.1 QMI pack/unpack (pack)

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](#)

6.1.1 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 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](#)

8.1.1 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</i> [MAX_DESCRIPTION_LENGTH]	<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.

8.1.2 Field Documentation

8.1.2.1 `uint8_t appStats::aidLength`

8.1.2.2 `uint8_t appStats::aidVal[255]`

8.1.2.3 `uint8_t appStats::appState`

8.1.2.4 `uint8_t appStats::appType`

8.1.2.5 `uint8_t appStats::persoFeature`

8.1.2.6 `uint8_t appStats::persoRetries`

8.1.2.7 `uint8_t appStats::persoState`

8.1.2.8 `uint8_t appStats::persoUnblockRetries`

8.1.2.9 `uint8_t appStats::pin1Retries`

8.1.2.10 `uint8_t appStats::pin1State`

8.1.2.11 `uint8_t appStats::pin2Retries`

8.1.2.12 `uint8_t appStats::pin2State`

8.1.2.13 `uint8_t appStats::puk1Retries`

8.1.2.14 uint8_t appStats::puk2Retries

8.1.2.15 uint8_t appStats::univPin

8.2 CarrierImage_t Struct Reference

Data Fields

- uint32_t [m_nCarrierId](#)
- uint32_t [m_nFolderId](#)
- uint32_t [m_nStorage](#)
- uint8_t [m_FwImageId](#) [16]
- uint8_t [m_FwBuildId](#) [32]
- uint8_t [m_PriImageId](#) [16]
- uint8_t [m_PriBuildId](#) [32]

8.2.1 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

8.2.2 Field Documentation

8.2.2.1 `uint8_t CarrierImage_t::m_FwBuildId[32]`

8.2.2.2 `uint8_t CarrierImage_t::m_FwImageId[16]`

8.2.2.3 `uint32_t CarrierImage_t::m_nCarrierId`

8.2.2.4 `uint32_t CarrierImage_t::m_nFolderId`

8.2.2.5 `uint32_t CarrierImage_t::m_nStorage`

8.2.2.6 `uint8_t CarrierImage_t::m_PriBuildId[32]`

8.2.2.7 `uint8_t CarrierImage_t::m_PrImageId[16]`

8.3 cdmaSSInfo Struct Reference

Data Fields

- `int8_t rssi`
- `int16_t ecio`

8.3.1 Detailed Description

Parameters

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

8.3.2 Field Documentation

8.3.2.1 `int16_t cdmaSSInfo::ecio`

8.3.2.2 `int8_t cdmaSSInfo::rssi`

8.4 connectionStatus Struct Reference

Data Fields

- `uint8_t MDMConnStatus`
- `uint64_t MDMCallDuration`

8.4.1 Detailed Description

Parameters

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

8.4.2 Field Documentation

8.4.2.1 uint64_t connectionStatus::MDMCallDuration

8.4.2.2 uint8_t connectionStatus::MDMConnStatus

8.5 currNetworkInfo Struct Reference

Data Fields

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

8.5.1 Detailed Description

Parameters

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

8.5.2 Field Documentation

8.5.2.1 uint8_t currNetworkInfo::NetworkType

8.5.2.2 uint32_t currNetworkInfo::RATMask

8.5.2.3 uint32_t currNetworkInfo::SOMask

8.6 dms_ActivationStatusTlv Struct Reference

Data Fields

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

8.6.1 Detailed Description

Activation Status Tlv

Parameters

<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
-------------------------	--

8.6.2 Field Documentation

8.6.2.1 uint32_t dms_ActivationStatusTlv::activationStatus

8.6.2.2 uint16_t dms_ActivationStatusTlv::TlvPresent

8.7 dms_OperatingModeTlv Struct Reference

Data Fields

- uint16_t [TlvPresent](#)
- uint32_t [operatingMode](#)

8.7.1 Detailed Description

Operating Mode Tlv

Parameters

<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
----------------------	--

8.7.2 Field Documentation

8.7.2.1 `uint32_t dms_OperatingModeTlv::operatingMode`

8.7.2.2 `uint16_t dms_OperatingModeTlv::TlvPresent`

8.8 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`

8.8.1 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)
<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

8.8.2 Field Documentation

8.8.2.1 `uint16_t DMScustSettingInfo::cust_attr`

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

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

8.8.2.4 `uint16_t DMScustSettingInfo::id_length`

8.8.2.5 `uint16_t DMScustSettingInfo::value_length`

8.9 DMScustSettingList Struct Reference

Data Fields

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

8.9.1 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 custSettingInfo for more information

8.9.2 Field Documentation

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

8.9.2.2 `uint8_t DMScustSettingList::list_type`

8.9.2.3 `uint16_t DMScustSettingList::num_instances`

8.10 DMSgetCustomFeatureV2 Struct Reference

Data Fields

- `DMSgetCustomInput * pGetCustomInput`
- `DMScustSettingInfo * pCustSettingInfo`
- `DMScustSettingList * pCustSettingList`

8.10.1 Detailed Description

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

Parameters

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

8.10.2 Field Documentation

8.10.2.1 DMScustSettingInfo* DMSgetCustomFeatureV2::pCustSettingInfo

8.10.2.2 DMScustSettingList* DMSgetCustomFeatureV2::pCustSettingList

8.10.2.3 DMSgetCustomInput* DMSgetCustomFeatureV2::pGetCustomInput

8.11 DMSgetCustomInput Struct Reference

Data Fields

- uint8_t [cust_id](#) [64+1]
- uint8_t [list_type](#)

8.11.1 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)
<i>list_type</i>	<ul style="list-style-type: none"> • list type requested

8.11.2 Field Documentation

8.11.2.1 uint8_t DMSgetCustomInput::cust_id[64+1]

8.11.2.2 uint8_t DMSgetCustomInput::list_type

8.12 dunchannelRate Struct Reference

Data Fields

- uint32_t [CurrChanTxRate](#)
- uint32_t [CurrChanRxRate](#)

- uint32_t [MaxChanTxRate](#)
- uint32_t [MaxChanRxRate](#)

8.12.1 Detailed Description

Parameters

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

8.12.2 Field Documentation

8.12.2.1 uint32_t dunchannelRate::CurrChanRxRate

8.12.2.2 uint32_t dunchannelRate::CurrChanTxRate

8.12.2.3 uint32_t dunchannelRate::MaxChanRxRate

8.12.2.4 uint32_t dunchannelRate::MaxChanTxRate

8.13 eTWSPLMNInfoTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [sMSEtwsPlmnInfo](#) [ETWSPLMNInfo](#)

8.13.1 Detailed Description

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 sMSEtwsPlmnInfo for more information

8.13.2 Field Documentation

8.13.2.1 [sMSEtwsPlmnInfo](#) [eTWSPLMNInfoTlv::ETWSPLMNInfo](#)

8.13.2.2 uint8_t [eTWSPLMNInfoTlv::TlvPresent](#)

8.14 FMSImageElement Struct Reference

Data Fields

- uint8_t [imageType](#)
- uint8_t [imageId](#) [16]
- uint8_t [buildIdLength](#)
- uint8_t [buildId](#) [100]

8.14.1 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

8.14.2 Field Documentation

8.14.2.1 uint8_t FMSImageElement::buildId[100]

8.14.2.2 uint8_t FMSImageElement::buildIdLength

8.14.2.3 uint8_t FMSImageElement::imageId[16]

8.14.2.4 uint8_t FMSImageElement::imageType

8.15 FMSImageIdElement Struct Reference

Data Fields

- uint8_t [storageIndex](#)
- uint8_t [failureCount](#)
- uint8_t [imageID](#) [16]
- uint8_t [buildIDLength](#)
- uint8_t [buildID](#) [100]

8.15.1 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)

8.15.2 Field Documentation

8.15.2.1 `uint8_t FMSImageIDElement::buildID[100]`8.15.2.2 `uint8_t FMSImageIDElement::buildIDLength`8.15.2.3 `uint8_t FMSImageIDElement::failureCount`8.15.2.4 `uint8_t FMSImageIDElement::imageID[16]`8.15.2.5 `uint8_t FMSImageIDElement::storageIndex`

8.16 FMSImageIDEntries Struct Reference

Data Fields

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

8.16.1 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

8.16.2 Field Documentation

8.16.2.1 `uint8_t FMSImageIDEntries::executingImage`8.16.2.2 `FMSImageIDElement FMSImageIDEntries::imageIDElement[50]`8.16.2.3 `uint8_t FMSImageIDEntries::imageIDSize`8.16.2.4 `uint8_t FMSImageIDEntries::imageType`8.16.2.5 `uint8_t FMSImageIDEntries::maxImages`

8.17 FMSImageList Struct Reference

Data Fields

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

8.17.1 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)
-----------------------	---

8.17.2 Field Documentation

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

8.17.2.2 **uint8_t** FMSImageList::listSize

8.18 FMSPrefImageList Struct Reference

Data Fields

- [uint8_t listSize](#)
- [FMSImageElement listEntries](#) [2]

8.18.1 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

8.18.2 Field Documentation

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

8.18.2.2 **uint8_t** FMSPrefImageList::listSize

8.19 hdrSSInfo Struct Reference

Data Fields

- [int8_t rssi](#)
- [int16_t ecio](#)
- [uint8_t sinr](#)
- [int32_t io](#)

8.19.1 Detailed Description

Parameters

<i>rss</i>	RSSI in dBm.
<i>ecio</i>	ECIO value representing negative 0.5 dBm increment
<i>sinr</i>	SINR level.
<i>io</i>	Received IO in dBm.

8.19.2 Field Documentation

8.19.2.1 int16_t hdrSSInfo::ecio

8.19.2.2 int32_t hdrSSInfo::io

8.19.2.3 int8_t hdrSSInfo::rss

8.19.2.4 uint8_t hdrSSInfo::sinr

8.20 image_info_t Struct Reference

Data Fields

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

8.20.1 Field Documentation

8.20.1.1 uint8_t image_info_t::buildID[255]

8.20.1.2 uint8_t image_info_t::buildIDLen

8.20.1.3 uint8_t image_info_t::imageType

8.20.1.4 uint8_t image_info_t::uniqueID[16]

8.21 ipv6AddressInfo Struct Reference

Data Fields

- uint8_t [IPV6PrefixLen](#)
- uint16_t [IPAddressV6](#) [8]

8.21.1 Detailed Description

Parameters

<i>IPV6PrefixLen</i>	Length of the received IPv6 address
<i>IPAddressV6</i>	IPv6 address(in network byte order)

8.21.2 Field Documentation

8.21.2.1 `uint16_t ipv6AddressInfo::IPAddressV6[8]`

8.21.2.2 `uint8_t ipv6AddressInfo::IPv6PrefixLen`

8.22 LibPackGPRSRequestedQoS Struct Reference

Data Fields

- `uint32_t precedenceClass`
- `uint32_t delayClass`
- `uint32_t reliabilityClass`
- `uint32_t peakThroughputClass`
- `uint32_t meanThroughputClass`

8.22.1 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

8.22.2 Field Documentation

8.22.2.1 `uint32_t LibPackGPRSRequestedQoS::delayClass`

8.22.2.2 `uint32_t LibPackGPRSRequestedQoS::meanThroughputClass`

8.22.2.3 `uint32_t LibPackGPRSRequestedQoS::peakThroughputClass`

8.22.2.4 `uint32_t LibPackGPRSRequestedQoS::precedenceClass`

8.22.2.5 `uint32_t LibPackGPRSRequestedQoS::reliabilityClass`

8.23 LibpackProfile3GPP Struct Reference

Data Fields

- uint8_t * pProfilename
- uint16_t * pProfilenameSize
- uint8_t * pPDPtype
- uint8_t * pPdpHdrCompType
- uint8_t * pPdpDataCompType
- uint8_t * pAPNName
- uint16_t * pAPNnameSize
- uint32_t * pPriDNSIPv4AddPref
- uint32_t * pSecDNSIPv4AddPref
- LibPackUMTSQoS * pUMTSReqQoS
- LibPackUMTSQoS * pUMTSMInQoS
- LibPackGPRSRequestedQoS * pGPRSRequestedQoS
- LibPackGPRSRequestedQoS * pGPRSMinimumQoS
- uint8_t * pUsername
- uint16_t * pUsernameSize
- uint8_t * pPassword
- uint16_t * pPasswordSize
- uint8_t * pAuthenticationPref
- uint32_t * pIPv4AddrPref
- uint8_t * pPcscfAddrUsingPCO
- uint8_t * pPdpAccessConFlag
- uint8_t * pPcscfAddrUsingDhcp
- uint8_t * pImCnFlag
- LibPackTFTIDParams * pTFTID1Params
- LibPackTFTIDParams * pTFTID2Params
- uint8_t * pPdpContext
- uint8_t * pSecondaryFlag
- uint8_t * pPrimaryID
- uint16_t * pIPv6AddPref
- LibPackUMTSReqQoSSigInd * pUMTSReqQoSSigInd
- LibPackUMTSReqQoSSigInd * pUMTSMInQoSsigInd
- uint16_t * pPriDNSIPv6addpref
- uint16_t * pSecDNSIPv6addpref
- uint8_t * pAddrAllocPref
- LibPackQoSClassID * pQoSClassID
- uint8_t * pAPNDisabledFlag
- uint32_t * pPDNInactivTimeout
- uint8_t * pAPNClass

8.23.1 Detailed Description

Parameters

<i>extended</i>	error
<i>profile</i>	<p>info This structure contains Input parameters of SLQSCreateProfile and SLQSModifyProfile and output parameters of SLQSGetProfileSettings</p> <ul style="list-style-type: none"> • Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

<i>pProfileName</i>	<ul style="list-style-type: none"> One or more uint8_ts 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 pProfile-Name field. Size of this parameter is 2 uint8_ts.
<i>pPDPTType</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 pAPN-Name field. Size of this parameter is 2 uint8_ts.
<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 uint8_ts.
<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 uint8_ts.
<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>pSecondaryDN-SIPv6addpref</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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().

8.23.2 Field Documentation

8.23.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

8.23.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

8.23.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

- 8.23.2.4 `uint8_t*` `LibpackProfile3GPP::pAPNName`
- 8.23.2.5 `uint16_t*` `LibpackProfile3GPP::pAPNnameSize`
- 8.23.2.6 `uint8_t*` `LibpackProfile3GPP::pAuthenticationPref`
- 8.23.2.7 `LibPackGPRSRequestedQoS*` `LibpackProfile3GPP::pGPRSMinimumQoS`
- 8.23.2.8 `LibPackGPRSRequestedQoS*` `LibpackProfile3GPP::pGPRSRequestedQoS`
- 8.23.2.9 `uint8_t*` `LibpackProfile3GPP::plmCnFlag`
- 8.23.2.10 `uint32_t*` `LibpackProfile3GPP::pIPv4AddrPref`
- 8.23.2.11 `uint16_t*` `LibpackProfile3GPP::pIPv6AddPref`
- 8.23.2.12 `uint8_t*` `LibpackProfile3GPP::pPassword`
- 8.23.2.13 `uint16_t*` `LibpackProfile3GPP::pPasswordSize`
- 8.23.2.14 `uint8_t*` `LibpackProfile3GPP::pPcscfAddrUsingDhcp`
- 8.23.2.15 `uint8_t*` `LibpackProfile3GPP::pPcscfAddrUsingPCO`
- 8.23.2.16 `uint32_t*` `LibpackProfile3GPP::pPDNInactivTimeout`
- 8.23.2.17 `uint8_t*` `LibpackProfile3GPP::pPdpAccessConFlag`
- 8.23.2.18 `uint8_t*` `LibpackProfile3GPP::pPdpContext`
- 8.23.2.19 `uint8_t*` `LibpackProfile3GPP::pPdpDataCompType`
- 8.23.2.20 `uint8_t*` `LibpackProfile3GPP::pPdpHdrCompType`
- 8.23.2.21 `uint8_t*` `LibpackProfile3GPP::pPDPTtype`
- 8.23.2.22 `uint32_t*` `LibpackProfile3GPP::pPriDNSIPv4AddPref`
- 8.23.2.23 `uint16_t*` `LibpackProfile3GPP::pPriDNSIPv6addpref`
- 8.23.2.24 `uint8_t*` `LibpackProfile3GPP::pPrimaryID`
- 8.23.2.25 `uint8_t*` `LibpackProfile3GPP::pProfilename`
- 8.23.2.26 `uint16_t*` `LibpackProfile3GPP::pProfilenameSize`
- 8.23.2.27 `LibPackQosClassID*` `LibpackProfile3GPP::pQosClassID`
- 8.23.2.28 `uint32_t*` `LibpackProfile3GPP::pSecDNSIPv4AddPref`
- 8.23.2.29 `uint16_t*` `LibpackProfile3GPP::pSecDNSIPv6addpref`
- 8.23.2.30 `uint8_t*` `LibpackProfile3GPP::pSecondaryFlag`
- 8.23.2.31 `LibPackTFTIDParams*` `LibpackProfile3GPP::pTFTID1Params`

8.23.2.32 **LibPackTFTIDParams*** LibpackProfile3GPP::pTFTID2Params

8.23.2.33 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSMinQoS

8.23.2.34 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSMinQoSsigInd

8.23.2.35 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSReqQoS

8.23.2.36 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSReqQoSsigInd

8.23.2.37 **uint8_t*** LibpackProfile3GPP::pUsername

8.23.2.38 **uint16_t*** LibpackProfile3GPP::pUsernameSize

8.24 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](#)

8.24.1 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>plpcpAck-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>plpcpCreqRetry-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 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.
<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.
<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.
<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.
<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 uint8_ts 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 uint8_ts.
<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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().

<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().
-----------------------	---

8.24.2 Field Documentation

8.24.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

8.24.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

8.24.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

8.24.2.4 uint8_t* LibpackProfile3GPP2::pApnString

8.24.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

8.24.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

8.24.2.7 uint32_t* LibpackProfile3GPP2::pAppType

8.24.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

8.24.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

8.24.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

8.24.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

8.24.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

8.24.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

8.24.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

8.24.2.15 uint16_t* LibpackProfile3GPP2::plpcpAckTimeout

8.24.2.16 uint8_t* LibpackProfile3GPP2::plpcpCreqRetryCount

8.24.2.17 uint8_t* LibpackProfile3GPP2::plsPcscfAddressNedded

8.24.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

8.24.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount

8.24.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref

8.24.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

8.24.2.22 uint8_t* LibpackProfile3GPP2::pPdnType

- 8.24.2.23 `uint32_t*` `LibpackProfile3GPP2::pPppSessCloseTimer1x`
- 8.24.2.24 `uint32_t*` `LibpackProfile3GPP2::pPppSessCloseTimerDO`
- 8.24.2.25 `uint32_t*` `LibpackProfile3GPP2::pPrimaryV4DnsAddress`
- 8.24.2.26 `uint16_t*` `LibpackProfile3GPP2::pPriV6DnsAddress`
- 8.24.2.27 `uint8_t*` `LibpackProfile3GPP2::pRATType`
- 8.24.2.28 `uint32_t*` `LibpackProfile3GPP2::pSecondaryV4DnsAddress`
- 8.24.2.29 `uint16_t*` `LibpackProfile3GPP2::pSecV6DnsAddress`
- 8.24.2.30 `uint8_t*` `LibpackProfile3GPP2::pUserId`
- 8.24.2.31 `uint16_t*` `LibpackProfile3GPP2::pUserIdSize`

8.25 LibPackprofile_3GPP 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

8.25.1 Detailed Description

This structure contains Input parameters of SLQSCreateProfile and SLQSModifyProfile and output parameters of SLQSGetProfileSettings

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

Parameters

<i>pProfileName</i>	<ul style="list-style-type: none"> • 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>pPDPTType</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 pAPN-Name 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 mecha-

<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>pPDNInactiv- 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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().

8.25.2 Field Documentation

8.25.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.25.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.25.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.25.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

8.25.2.5 uint16_t* LibPackprofile_3GPP::pAPNnameSize

8.25.2.6 uint8_t* LibPackprofile_3GPP::pAuthenticationPref

8.25.2.7 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSMinimumQoS

8.25.2.8 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSRequestedQoS

8.25.2.9 uint8_t* LibPackprofile_3GPP::pImCnFlag

8.25.2.10 uint32_t* LibPackprofile_3GPP::pIPv4AddrPref

8.25.2.11 uint16_t* LibPackprofile_3GPP::pIPv6AddPref

8.25.2.12 uint8_t* LibPackprofile_3GPP::pPassword

8.25.2.13 uint16_t* LibPackprofile_3GPP::pPasswordSize

8.25.2.14 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingDhcp

8.25.2.15 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingPCO

8.25.2.16 uint32_t* LibPackprofile_3GPP::pPDNInactivTimeout

8.25.2.17 uint8_t* LibPackprofile_3GPP::pPdpAccessConFlag

- 8.25.2.18 uint8_t* LibPackprofile_3GPP::pPdpContext
- 8.25.2.19 uint8_t* LibPackprofile_3GPP::pPdpDataCompType
- 8.25.2.20 uint8_t* LibPackprofile_3GPP::pPdpHdrCompType
- 8.25.2.21 uint8_t* LibPackprofile_3GPP::pPDPTtype
- 8.25.2.22 uint32_t* LibPackprofile_3GPP::pPriDNSIPv4AddPref
- 8.25.2.23 uint16_t* LibPackprofile_3GPP::pPriDNSIPv6addpref
- 8.25.2.24 uint8_t* LibPackprofile_3GPP::pPrimaryID
- 8.25.2.25 uint8_t* LibPackprofile_3GPP::pProfilename
- 8.25.2.26 uint16_t* LibPackprofile_3GPP::pProfilenameSize
- 8.25.2.27 LibPackQosClassID* LibPackprofile_3GPP::pQosClassID
- 8.25.2.28 uint32_t* LibPackprofile_3GPP::pSecDNSIPv4AddPref
- 8.25.2.29 uint16_t* LibPackprofile_3GPP::pSecDNSIPv6addpref
- 8.25.2.30 uint8_t* LibPackprofile_3GPP::pSecondaryFlag
- 8.25.2.31 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID1Params
- 8.25.2.32 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID2Params
- 8.25.2.33 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSMinQoS
- 8.25.2.34 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSMinQoSsigInd
- 8.25.2.35 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSReqQoS
- 8.25.2.36 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSReqQoSSigInd
- 8.25.2.37 uint8_t* LibPackprofile_3GPP::pUsername
- 8.25.2.38 uint16_t* LibPackprofile_3GPP::pUsernameSize

8.26 LibPackprofile_3GPP2 Struct Reference

Data Fields

- uint8_t * [pNegoDnsSrvrPref](#)
- uint32_t * [pPppSessCloseTimerDO](#)
- uint32_t * [pPppSessCloseTimer1x](#)
- uint8_t * [pAllowLinger](#)
- uint16_t * [pLcpAckTimeout](#)
- uint16_t * [pIpccpAckTimeout](#)
- uint16_t * [pAuthTimeout](#)
- uint8_t * [pLcpCreqRetryCount](#)
- uint8_t * [pIpccpCreqRetryCount](#)

- uint8_t * [pAuthRetryCount](#)
- uint8_t * [pAuthProtocol](#)
- uint8_t * [pUserId](#)
- uint16_t * [pUserIdSize](#)
- uint8_t * [pAuthPassword](#)
- uint16_t * [pAuthPassword_tSize](#)
- uint8_t * [pDataRate](#)
- uint32_t * [pAppType](#)
- uint8_t * [pDataMode](#)
- uint8_t * [pAppPriority](#)
- uint8_t * [pApnString](#)
- uint16_t * [pApnStringSize](#)
- uint8_t * [pPdnType](#)
- uint8_t * [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](#)

8.26.1 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>pPppSessCloseTimer1x</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>plpcpAckTimeout</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>pLcpCreqRetryCount</i>	<ul style="list-style-type: none"> • LCP Configuration Request Retry Count
<i>plpcpCreqRetryCount</i>	<ul style="list-style-type: none"> • IPCP Configuration Request Retry Count
<i>pAuthRetryCount</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>plsPcsf-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
Generated on Tue May 31 2016 14:24:35. This page is automatically generated by using the function SLQSGet-ProfileSettings().	

<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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().
<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 SLQSGet-ProfileSettings().

8.26.2 Field Documentation

8.26.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

8.26.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

8.26.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

8.26.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

8.26.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

8.26.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

8.26.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

8.26.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

8.26.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

8.26.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

8.26.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

- 8.26.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout
- 8.26.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode
- 8.26.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate
- 8.26.2.15 uint16_t* LibPackprofile_3GPP2::plpcpAckTimeout
- 8.26.2.16 uint8_t* LibPackprofile_3GPP2::plpcpCreqRetryCount
- 8.26.2.17 uint8_t* LibPackprofile_3GPP2::plsPcscfAddressNedded
- 8.26.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout
- 8.26.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount
- 8.26.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref
- 8.26.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2
- 8.26.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType
- 8.26.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x
- 8.26.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO
- 8.26.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress
- 8.26.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress
- 8.26.2.27 uint8_t* LibPackprofile_3GPP2::pRATType
- 8.26.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress
- 8.26.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress
- 8.26.2.30 uint8_t* LibPackprofile_3GPP2::pUserId
- 8.26.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

8.27 LibPackQosClassID Struct Reference

Data Fields

- uint8_t [QCI](#)
- uint8_t [gDIBitRate](#)
- uint32_t [maxDIBitRate](#)
- uint32_t [gUIBitRate](#)
- uint32_t [maxUIBitRate](#)

8.27.1 Detailed Description

structure contains 3GPP LTE QoS parameters

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

Parameters

<i>QCI</i>	<ul style="list-style-type: none"> 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>gDlBitRate</i>	<ul style="list-style-type: none"> Guaranteed DL bit rate
<i>maxDlBitRate</i>	<ul style="list-style-type: none"> maxDlBitRate
<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

8.27.2 Field Documentation

8.27.2.1 `uint8_t LibPackQosClassID::gDlBitRate`8.27.2.2 `uint32_t LibPackQosClassID::gUlBitRate`8.27.2.3 `uint32_t LibPackQosClassID::maxDlBitRate`8.27.2.4 `uint32_t LibPackQosClassID::maxUlBitRate`8.27.2.5 `uint8_t LibPackQosClassID::QCI`

8.28 LibPackTFTIDParams Struct Reference

Data Fields

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

8.28.1 Detailed Description

structure contains traffic flow template parameters

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

Parameters

<i>filterId</i>	<ul style="list-style-type: none"> • Filter identifier
<i>eValid</i>	<ul style="list-style-type: none"> • Evaluation precedence index
<i>pVersion</i>	<ul style="list-style-type: none"> • IP version number <ul style="list-style-type: none"> – 4 - IPv4 – 6 - IPv6
<i>sourceIP</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

8.28.2 Field Documentation

- 8.28.2.1 `uint16_t LibPackTFTIDParams::destPortRangeEnd`
- 8.28.2.2 `uint32_t LibPackTFTIDParams::destPortRangeStart`
- 8.28.2.3 `uint8_t LibPackTFTIDParams::eValid`
- 8.28.2.4 `uint8_t LibPackTFTIDParams::filterId`
- 8.28.2.5 `uint32_t LibPackTFTIDParams::flowLabel`
- 8.28.2.6 `uint32_t LibPackTFTIDParams::IPSECSPi`
- 8.28.2.7 `uint8_t LibPackTFTIDParams::ipVersion`
- 8.28.2.8 `uint8_t LibPackTFTIDParams::nextHeader`
- 8.28.2.9 `uint16_t* LibPackTFTIDParams::pSourceIP`
- 8.28.2.10 `uint8_t LibPackTFTIDParams::sourceIPMask`
- 8.28.2.11 `uint16_t LibPackTFTIDParams::srcPortRangeEnd`
- 8.28.2.12 `uint16_t LibPackTFTIDParams::srcPortRangeStart`
- 8.28.2.13 `uint16_t LibPackTFTIDParams::tosMask`

8.29 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`

8.29.1 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
<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"> • Guranteed 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>	<p>- Residual bit error ratio</p> <ul style="list-style-type: none"> • Target value for undetected bit error ratio in in the delivered SDUs. • 0x00 - Subscribe • 0x01 - 5×10^{-2} • 0x02 - 1×10^{-2} • 0x03 - 5×10^{-3} • 0x04 - 4×10^{-3} • 0x05 - 1×10^{-3} • 0x06 - 1×10^{-4} • 0x07 - 1×10^{-5} • 0x08 - 1×10^{-6} • 0x09 - 1×10^{-8}
<i>deliveryErrSDU</i>	<p>- Delivery of erroneous SDUs</p> <ul style="list-style-type: none"> • Indicates whether SDUs detected as erroneous shall be delivered or not. • 0x00 - Subscribe • 0x01 - 5×10^{-2} • 0x02 - 1×10^{-2} • 0x03 - 5×10^{-3} • 0x04 - 4×10^{-3} • 0x05 - 1×10^{-3} • 0x06 - 1×10^{-4} • 0x07 - 1×10^{-5} • 0x08 - 1×10^{-6} • 0x09 - 1×10^{-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.

8.29.2 Field Documentation

- 8.29.2.1 `uint8_t LibPackUMTSQoS::deliveryErrSDU`
- 8.29.2.2 `uint32_t LibPackUMTSQoS::grntDownlinkBitrate`
- 8.29.2.3 `uint32_t LibPackUMTSQoS::grntUplinkBitrate`
- 8.29.2.4 `uint32_t LibPackUMTSQoS::maxDownlinkBitrate`
- 8.29.2.5 `uint32_t LibPackUMTSQoS::maxSDUSize`
- 8.29.2.6 `uint32_t LibPackUMTSQoS::maxUplinkBitrate`
- 8.29.2.7 `uint8_t LibPackUMTSQoS::qosDeliveryOrder`
- 8.29.2.8 `uint8_t LibPackUMTSQoS::resBerRatio`
- 8.29.2.9 `uint8_t LibPackUMTSQoS::sduErrorRatio`
- 8.29.2.10 `uint8_t LibPackUMTSQoS::trafficClass`
- 8.29.2.11 `uint32_t LibPackUMTSQoS::trafficPriority`
- 8.29.2.12 `uint32_t LibPackUMTSQoS::transferDelay`

8.30 LibPackUMTSReqQoSsigInd Struct Reference

Data Fields

- [LibPackUMTSQoS UMTSReqQoS](#)
- `uint8_t` [SigInd](#)

8.30.1 Detailed Description

structure contains UMTS requested QoS with Signaling Indication flag

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

Parameters

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

8.30.2 Field Documentation

8.30.2.1 `uint8_t LibPackUMTSReqQoSSigInd::SigInd`8.30.2.2 `LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS`8.31 `loc_BdsSV` Struct Reference

Data Fields

- `uint16_t id`
- `uint8_t mask`

8.31.1 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"> – <code>QMI_LOC_MASK_DELETE_EPHEMERIS (0x01)</code> - Delete ephemeris for the satellite – <code>QMI_LOC_MASK_DELETE_ALMANAC (0x02)</code> - Delete almanac for the satellite

8.31.2 Field Documentation

8.31.2.1 `uint16_t loc_BdsSV::id`8.31.2.2 `uint8_t loc_BdsSV::mask`8.32 `loc_BdsSVInfo` Struct Reference

Data Fields

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

8.32.1 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">– <code>gnssSvId</code>– <code>deleteSvInfoMask</code>
<i>pSV</i>	<ul style="list-style-type: none">• Pointer to struct loc_BdsSV. See loc_BdsSV for more information

8.32.2 Field Documentation

8.32.2.1 `uint8_t` `loc_BdsSVInfo::len`

8.32.2.2 `loc_BdsSV *` `loc_BdsSVInfo::pSV`

8.33 loc_CellDb Struct Reference

Data Fields

- `uint32_t` [mask](#)

8.33.1 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
-------------	--

8.33.2 Field Documentation

8.33.2.1 uint32_t loc_CellDb::mask

8.34 loc_ClkInfo Struct Reference

Data Fields

- uint32_t [mask](#)

8.34.1 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
-------------	--

8.34.2 Field Documentation

8.34.2.1 `uint32_t loc_ClkInfo::mask`

8.35 `loc_GnssData` Struct Reference

Data Fields

- `uint64_t` [mask](#)

8.35.1 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 GLO-NASS SVDIR – QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete GLO-NASS 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
-------------	---

8.35.2 Field Documentation

8.35.2.1 `uint64_t loc_GnssData::mask`

8.36 `loc_gpsTime` Struct Reference

Data Fields

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

8.36.1 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

8.36.2 Field Documentation

8.36.2.1 `uint32_t loc_gpsTime::gpsTimeOfWeekMs`

8.36.2.2 `uint16_t loc_gpsTime::gpsWeek`

8.37 `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](#)

8.37.1 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

8.37.2 Field Documentation

8.37.2.1 uint8_t loc_LocApplicationInfo::appNameLength

8.37.2.2 uint8_t loc_LocApplicationInfo::appProviderLength

8.37.2.3 uint8_t loc_LocApplicationInfo::appVersionLength

8.37.2.4 uint8_t loc_LocApplicationInfo::appVersionValid

8.37.2.5 uint8_t* loc_LocApplicationInfo::pAppName

8.37.2.6 uint8_t* loc_LocApplicationInfo::pAppProvider

8.37.2.7 uint8_t* loc_LocApplicationInfo::pAppVersion

8.38 loc_precisionDilution Struct Reference

Data Fields

- uint32_t [PDOP](#)

- uint32_t [HDOP](#)
- uint32_t [VDOP](#)

8.38.1 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)

8.38.2 Field Documentation

8.38.2.1 uint32_t loc_precisionDilution::HDOP

8.38.2.2 uint32_t loc_precisionDilution::PDOP

8.38.2.3 uint32_t loc_precisionDilution::VDOP

8.39 loc_sensorDataUsage Struct Reference

Data Fields

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

8.39.1 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.
------------------	---

- Value
 - 0x00000001 - Accelerometer used
 - 0x00000002 - Gyroscope used

Parameters

<i>aidingIndicatorMask</i>	
----------------------------	--

- Specifies which results were aided by sensors.

- Value

- 0x00000001 - AIDED_HEADING
- 0x00000002 - AIDED_SPEED
- 0x00000004 - AIDED_POSITION
- 0x00000008 - AIDED_VELOCITY

8.39.2 Field Documentation

8.39.2.1 `uint32_t loc_sensorDataUsage::aidingIndicatorMask`

8.39.2.2 `uint32_t loc_sensorDataUsage::usageMask`

8.40 loc_SV Struct Reference

Data Fields

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

8.40.1 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

8.40.2 Field Documentation

8.40.2.1 `uint16_t loc_SV::id`

8.40.2.2 `uint8_t loc_SV::mask`

8.40.2.3 `uint32_t loc_SV::system`

8.41 loc_SVInfo Struct Reference

Data Fields

- `uint8_t len`
- `loc_SV * pSV`

8.41.1 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 loc_SV: <ul style="list-style-type: none"> gnssSvId system deleteSvInfoMask
------------	--

<i>pSV</i>	<ul style="list-style-type: none"> • Pointer to struct loc_SV. See loc_SV for more information
------------	---

8.41.2 Field Documentation

8.41.2.1 `uint8_t loc_SVInfo::len`

8.41.2.2 `loc_SV* loc_SVInfo::pSV`

8.42 loc_svUsedforFix Struct Reference

Data Fields

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

8.42.1 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 gnssSvUsedList
<i>pGnssSvUsedList</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

8.42.2 Field Documentation

8.42.2.1 `uint16_t loc_svUsedforFix::gnssSvUsedList[255]`

8.42.2.2 `uint8_t loc_svUsedforFix::gnssSvUsedList_len`

8.43 IteSSInfo Struct Reference

Data Fields

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

- `int16_t snr`

8.43.1 Detailed Description

Parameters

<i>rsi</i>	RSSI in dBm.
<i>rsrq</i>	RSRQ value in dB
<i>rsrp</i>	Current RSRP in dBm as measured by L1.
<i>snr</i>	SNR level as a scaled integer in units of 0.1 dB.

8.43.2 Field Documentation

8.43.2.1 `int16_t lteSSInfo::rsrp`

8.43.2.2 `int8_t lteSSInfo::rsrq`

8.43.2.3 `int8_t lteSSInfo::rsi`

8.43.2.4 `int16_t lteSSInfo::snr`

8.44 messageModeTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSMessageModelInfo MessageModelInfo`

8.44.1 Detailed Description

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 <code>sMSMessageModelInfo</code> for more information

8.44.2 Field Documentation

8.44.2.1 `sMSMessageModelInfo messageModeTlv::MessageModelInfo`

8.44.2.2 `uint8_t messageModeTlv::TlvPresent`

8.45 nas_acqOrderPref Struct Reference

Data Fields

- `uint8_t acqOrdeLen`

- uint8_t * [pAcqOrder](#)

8.45.1 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_1xEVDO - 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

8.45.2 Field Documentation

8.45.2.1 uint8_t nas_acqOrderPref::acqOrdeLen

8.45.2.2 uint8_t* nas_acqOrderPref::pAcqOrder

8.46 nas_AddCDMASysInfo Struct Reference

Data Fields

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

8.46.1 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.
---------------	---

8.46.2 Field Documentation

8.46.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

8.46.2.2 uint16_t nas_AddCDMASysInfo::regPrd

8.47 nas_AddSysInfo Struct Reference

Data Fields

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

8.47.1 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

8.47.2 Field Documentation

8.47.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

8.47.2.2 uint16_t nas_AddSysInfo::geoSysIdx

8.48 nas_CallBarringSysInfo Struct Reference

Data Fields

- uint32_t [csBarStatus](#)

- uint32_t [psBarStatus](#)

8.48.1 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

8.48.2 Field Documentation

8.48.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

8.48.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

8.49 nas_callBarStatus Struct Reference

Data Fields

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

8.49.1 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

8.49.2 Field Documentation

8.49.2.1 `uint32_t nas_callBarStatus::csBarStatus`8.49.2.2 `uint32_t nas_callBarStatus::psBarStatus`8.50 `nas_CDMAECIOThresh` Struct Reference

Data Fields

- `uint8_t CDMAECIOThreshListLen`
- `int16_t * pCDMAECIOThreshList`

8.50.1 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)
----------------------------------	--

8.50.2 Field Documentation

8.50.2.1 `uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen`

8.50.2.2 `int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList`

8.51 nas_CDMAInfo Struct Reference

Data Fields

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

8.51.1 Detailed Description

This structure contains information about the CDMA Network.

Parameters

<i>sid</i>	<ul style="list-style-type: none"> • System ID. – 0xFFFF - Not Available
<i>nid</i>	<ul style="list-style-type: none"> • Network ID. – 0xFFFF - Not Available
<i>baselId</i>	<ul style="list-style-type: none"> • Base station ID. – 0xFFFF - Not Available
<i>refpn</i>	<ul style="list-style-type: none"> • Reference PN. – 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

8.51.2 Field Documentation

8.51.2.1 uint16_t nas_CDMAInfo::baseId

8.51.2.2 uint32_t nas_CDMAInfo::baseLat

8.51.2.3 uint32_t nas_CDMAInfo::baseLong

8.51.2.4 uint16_t nas_CDMAInfo::nid

8.51.2.5 uint16_t nas_CDMAInfo::refpn

8.51.2.6 uint16_t nas_CDMAInfo::sid

8.52 nas_CDMARSSIThresh Struct Reference

Data Fields

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

8.52.1 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).

8.52.2 Field Documentation

8.52.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

8.52.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

8.53 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]

8.53.1 Detailed Description

Structure for storing the CDMA System Information.

Parameters

<i>sysInfoCDMA</i>	<ul style="list-style-type: none"> • See 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>ccsSupported-Valid</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>	

8.53.2 Field Documentation

- 8.53.2.1 `uint16_t nas_CDMASysInfo::baseId`
- 8.53.2.2 `uint32_t nas_CDMASysInfo::baseLat`
- 8.53.2.3 `uint32_t nas_CDMASysInfo::baseLong`
- 8.53.2.4 `uint8_t nas_CDMASysInfo::bsInfoValid`
- 8.53.2.5 `uint8_t nas_CDMASysInfo::bsPRev`
- 8.53.2.6 `uint8_t nas_CDMASysInfo::bsPRevValid`
- 8.53.2.7 `uint8_t nas_CDMASysInfo::ccsSupported`
- 8.53.2.8 `uint8_t nas_CDMASysInfo::ccsSupportedValid`
- 8.53.2.9 `uint8_t nas_CDMASysInfo::cdmaSysIdValid`
- 8.53.2.10 `uint8_t nas_CDMASysInfo::isSysPrIMatch`
- 8.53.2.11 `uint8_t nas_CDMASysInfo::isSysPrIMatchValid`
- 8.53.2.12 `uint8_t nas_CDMASysInfo::MCC[3]`
- 8.53.2.13 `uint8_t nas_CDMASysInfo::MNC[3]`
- 8.53.2.14 `uint16_t nas_CDMASysInfo::networkID`
- 8.53.2.15 `uint8_t nas_CDMASysInfo::networkIdValid`
- 8.53.2.16 `uint16_t nas_CDMASysInfo::packetZone`
- 8.53.2.17 `uint8_t nas_CDMASysInfo::packetZoneValid`
- 8.53.2.18 `uint8_t nas_CDMASysInfo::pRevInUse`
- 8.53.2.19 `uint8_t nas_CDMASysInfo::pRevInUseValid`
- 8.53.2.20 `nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA`
- 8.53.2.21 `uint16_t nas_CDMASysInfo::systemID`

8.54 nas_CDMASysInfoExt Struct Reference

Data Fields

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

8.54.1 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

8.54.2 Field Documentation

8.54.2.1 `uint8_t nas_CDMASysInfoExt::imsi_11_12`

8.54.2.2 `uint16_t nas_CDMASysInfoExt::MCC`

8.55 nas_cellParams Struct Reference**Data Fields**

- `uint16_t pci`
- `int16_t rsrq`
- `int16_t rsrp`
- `int16_t rssi`
- `int16_t srxlev`

8.55.1 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>rsi</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.

8.55.2 Field Documentation

8.55.2.1 uint16_t nas_cellParams::pci

8.55.2.2 int16_t nas_cellParams::rsrp

8.55.2.3 int16_t nas_cellParams::rsrq

8.55.2.4 int16_t nas_cellParams::rsi

8.55.2.5 int16_t nas_cellParams::srxlev

8.56 nas_CommInfo Struct Reference

Data Fields

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

8.56.1 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

8.56.2 Field Documentation

8.56.2.1 `uint8_t nas_CommInfo::imsRegState`

8.56.2.2 `uint8_t nas_CommInfo::modemMode`

8.56.2.3 `uint8_t nas_CommInfo::psState`

8.56.2.4 `uint8_t nas_CommInfo::systemMode`

8.56.2.5 `int8_t nas_CommInfo::temperature`

8.57 nas_CSGID Struct Reference

Data Fields

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

8.57.1 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

8.57.2 Field Documentation

8.57.2.1 uint32_t nas_CSgid::id

8.57.2.2 uint16_t nas_CSgid::mcc

8.57.2.3 uint16_t nas_CSgid::mnc

8.57.2.4 uint8_t nas_CSgid::mncPcsDigits

8.57.2.5 uint8_t nas_CSgid::rat

8.58 nas_currentPLMN Struct Reference

Data Fields

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

8.58.1 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

8.58.2 Field Documentation

8.58.2.1 uint16_t nas_currentPLMN::MCC

8.58.2.2 uint16_t nas_currentPLMN::MNC

8.58.2.3 uint8_t nas_currentPLMN::netDescr[255]

8.58.2.4 uint8_t nas_currentPLMN::netDescrLength

8.59 nas_dataSrvCapabilities Struct Reference

Data Fields

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

8.59.1 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+

8.59.2 Field Documentation

8.59.2.1 `uint8_t nas_dataSrvCapabilities::dataCapabilities[32]`

8.59.2.2 `uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen`

8.60 nas_detailSvcInfo Struct Reference

Data Fields

- `uint8_t srvStatus`
- `uint8_t srvCapability`
- `uint8_t hdrSrvStatus`
- `uint8_t hdrHybrid`
- `uint8_t isSysForbidden`

8.60.1 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

8.60.2 Field Documentation

8.60.2.1 `uint8_t nas_detailSvcInfo::hdrHybrid`

8.60.2.2 `uint8_t nas_detailSvcInfo::hdrSrvStatus`

8.60.2.3 `uint8_t nas_detailSvcInfo::isSysForbidden`

8.60.2.4 `uint8_t nas_detailSvcInfo::srvCapability`

8.60.2.5 `uint8_t nas_detailSvcInfo::srvStatus`

8.61 nas_ecioListElement Struct Reference

Data Fields

- `int16_t ecio`
- `uint8_t radiolf`

8.61.1 Detailed Description

This structure contains the ECIO Information

Parameters

<i>ecio</i>	<ul style="list-style-type: none">• ECIO value in dBm
<i>radiolf</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

8.61.2 Field Documentation

8.61.2.1 int16_t nas_ecioListElement::ecio

8.61.2.2 uint8_t nas_ecioListElement::radiolf

8.62 nas_errorRateListElement Struct Reference

Data Fields

- uint16_t [errorRate](#)

- uint8_t [radiolf](#)

8.62.1 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_1xEVDO – cdma2000 HRPD (1xEV-DO) – 0x03 – RADIO_IF_AMPS – AMPS – 0x04 – RADIO_IF_GSM – GSM – 0x05 – RADIO_IF_UMTS – UMTS

8.62.2 Field Documentation

8.62.2.1 uint16_t nas_errorRateListElement::errorRate

8.62.2.2 uint8_t nas_errorRateListElement::radiolf

8.63 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]

8.63.1 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</i> [NAS_PLM-N_LENGTH]	<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>nmrInst</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_DE- SCRIPTION_LE- NGTH]</i>	<ul style="list-style-type: none"> • See nas_nmrCellInfo for more information.

8.63.2 Field Documentation

8.63.2.1 `uint16_t nas_GERANInfo::arfcn`

8.63.2.2 `uint8_t nas_GERANInfo::bsic`

8.63.2.3 `uint32_t nas_GERANInfo::cellID`

8.63.2.4 `nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]`

8.63.2.5 uint16_t nas_GERANInfo::lac

8.63.2.6 uint8_t nas_GERANInfo::nmrInst

8.63.2.7 uint8_t nas_GERANInfo::plmn[3]

8.63.2.8 uint16_t nas_GERANInfo::rxLev

8.63.2.9 uint32_t nas_GERANInfo::timingAdvance

8.64 nas_geranInstInfo Struct Reference

Data Fields

- uint16_t [geranArfcn](#)
- uint8_t [geranBsicNcc](#)
- uint8_t [geranBsicBcc](#)
- int16_t [geranRssi](#)

8.64.1 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.

8.64.2 Field Documentation

8.64.2.1 uint16_t nas_geranInstInfo::geranArfcn

8.64.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

8.64.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

8.64.2.4 int16_t nas_geranInstInfo::geranRssi

8.65 nas_gsmCellInfo Struct Reference

Data Fields

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

8.65.1 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.

8.65.2 Field Documentation

8.65.2.1 uint16_t nas_gsmCellInfo::arfcn

8.65.2.2 uint8_t nas_gsmCellInfo::band1900

8.65.2.3 uint8_t nas_gsmCellInfo::bsicId

8.65.2.4 `uint8_t nas_gsmCellInfo::cellIdValid`

8.65.2.5 `int16_t nas_gsmCellInfo::rssi`

8.65.2.6 `int16_t nas_gsmCellInfo::srxlev`

8.66 nas_GSMRSSIThresh Struct Reference

Data Fields

- `uint8_t GSMRSSIThreshListLen`
- `int16_t * pGSMRSSIThreshList`

8.66.1 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)

8.66.2 Field Documentation

8.66.2.1 `uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen`

8.66.2.2 `int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList`

8.67 nas_GSMSrvStatusInfo Struct Reference

Data Fields

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

8.67.1 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

8.67.2 Field Documentation

8.67.2.1 `uint8_t nas_GSMSrvStatusInfo::isPrefDataPath`

8.67.2.2 `uint8_t nas_GSMSrvStatusInfo::srvStatus`

8.67.2.3 `uint8_t nas_GSMSrvStatusInfo::trueSrvStatus`

8.68 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](#)

8.68.1 Detailed Description

Structure for storing the GSM System Information.

Parameters

<i>sysInfoGSM</i>	<ul style="list-style-type: none"> • See 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[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Country Code. • MCC digits in ASCII characters
<i>MNC[PLMN_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

8.68.2 Field Documentation

8.68.2.1 `uint32_t nas_GSMsystInfo::cellId`

8.68.2.2 `uint8_t nas_GSMsystInfo::cellIdValid`

8.68.2.3 `uint8_t nas_GSMsystInfo::dtmSupp`

8.68.2.4 `uint8_t nas_GSMsystInfo::dtmSuppValid`

8.68.2.5 `uint8_t nas_GSMsystInfo::egprsSupp`

8.68.2.6 `uint8_t nas_GSMsystInfo::egprsSuppValid`

8.68.2.7 `uint16_t nas_GSMsystInfo::lac`

8.68.2.8 `uint8_t nas_GSMsystInfo::lacValid`

8.68.2.9 `uint8_t nas_GSMsystInfo::MCC[3]`

8.68.2.10 `uint8_t nas_GSMsystInfo::MNC[3]`

8.68.2.11 `uint8_t nas_GSMsystInfo::networkIdValid`

8.68.2.12 `uint8_t nas_GSMsystInfo::regRejectInfoValid`

8.68.2.13 `uint8_t nas_GSMsInfo::rejCause`

8.68.2.14 `uint8_t nas_GSMsInfo::rejectSrvDomain`

8.68.2.15 `nas_sysInfoCommon nas_GSMsInfo::sysInfoGSM`

8.69 nas_HDRECIOTresh Struct Reference

Data Fields

- `uint8_t HDRECIOTreshListLen`
- `int16_t * pHRECIOTreshList`

8.69.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

Parameters

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

8.69.2 Field Documentation

8.69.2.1 `uint8_t nas_HDRECIOTresh::HDRECIOTreshListLen`

8.69.2.2 `int16_t* nas_HDRECIOTresh::pHDRECIOTreshList`

8.70 nas_HDRIOTresh Struct Reference

Data Fields

- `uint8_t HDRIOTreshListLen`
- `int16_t * pHRIOTreshList`

8.70.1 Detailed Description

This structure contains HDR IO threshold related parameters.

Parameters

<i>HDRIOTthresh-ListLen</i>	<ul style="list-style-type: none"> Length of the HDR IO threshold list parameter to follow
<i>pHDRIOTthresh-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).

8.70.2 Field Documentation

8.70.2.1 `uint8_t nas_HDRIOTthresh::HDRIOTthreshListLen`8.70.2.2 `int16_t* nas_HDRIOTthresh::pHDRIOTthreshList`

8.71 nas_HDRRSSIThresh Struct Reference

Data Fields

- `uint8_t HDRRSSIthreshListLen`
- `int16_t * pHDRRSSIthreshList`

8.71.1 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).

8.71.2 Field Documentation

8.71.2.1 `uint8_t nas_HDRRSSIthresh::HDRRSSIthreshListLen`8.71.2.2 `int16_t* nas_HDRRSSIthresh::pHDRRSSIthreshList`

8.72 nas_HDRSINRThreshold Struct Reference

Data Fields

- `uint8_t HDRSINRthreshListLen`

- `uint16_t * pHDRSINRThreshList`

8.72.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

Parameters

<i>HDRSINR- ThreshListLen</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

8.72.2 Field Documentation

8.72.2.1 `uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen`

8.72.2.2 `uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList`

8.73 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]`

8.73.1 Detailed Description

Structure for storing the HDR System Information.

Parameters

<i>sysInfoHDR</i>	<ul style="list-style-type: none"> • See 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>hdrActiveProtValid</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.

8.73.2 Field Documentation

8.73.2.1 `uint8_t nas_HDRSysInfo::hdrActiveProt`

8.73.2.2 `uint8_t nas_HDRSysInfo::hdrActiveProtValid`

8.73.2.3 `uint8_t nas_HDRSysInfo::hdrPersonality`

8.73.2.4 `uint8_t nas_HDRSysInfo::hdrPersonalityValid`

8.73.2.5 `uint8_t nas_HDRSysInfo::is856SysId[16]`

8.73.2.6 `uint8_t nas_HDRSysInfo::is856SysIdValid`

8.73.2.7 `uint8_t nas_HDRSysInfo::isSysPrIMatch`

8.73.2.8 `uint8_t nas_HDRSysInfo::isSysPrIMatchValid`

8.73.2.9 `nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR`

8.74 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]

8.74.1 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.

8.74.2 Field Documentation

8.74.2.1 uint8_t nas_infoInterFreq::cell_resel_priority

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

8.74.2.3 `uint8_t` `nas_infoInterFreq::cells_len`

8.74.2.4 `uint16_t` `nas_infoInterFreq::earfcn`

8.74.2.5 `uint8_t` `nas_infoInterFreq::threshXHigh`

8.74.2.6 `uint8_t` `nas_infoInterFreq::threshXLow`

8.75 `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]

8.75.1 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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 ue_in_idle is TRUE.
<i>cells_len</i>	<ul style="list-style-type: none"> • Provides the number of set of gsm cells.
<i>GsmCellInfo[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_gsmCellInfo for more information.

8.75.2 Field Documentation

8.75.2.1 `uint8_t nas_lteGsmCellInfo::cellReselPriority`

8.75.2.2 `uint8_t nas_lteGsmCellInfo::cells_len`

8.75.2.3 `nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]`

8.75.2.4 `uint8_t nas_lteGsmCellInfo::nccPermitted`

8.75.2.5 `uint8_t nas_lteGsmCellInfo::threshGsmHigh`

8.75.2.6 `uint8_t nas_lteGsmCellInfo::threshGsmLow`

8.76 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`

8.76.1 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
---------------------	--

8.76.2 Field Documentation

8.76.2.1 `uint8_t nas_LTEInfo::band`

8.76.2.2 `uint8_t nas_LTEInfo::bandwidth`

8.76.2.3 `uint8_t nas_LTEInfo::emmConnState`

8.76.2.4 `uint8_t nas_LTEInfo::emmState`

8.76.2.5 `uint8_t nas_LTEInfo::emmSubState`

8.76.2.6 `uint16_t nas_LTEInfo::RXChan`

8.76.2.7 `uint16_t nas_LTEInfo::TXChan`

8.77 `nas_LTEInfoInterfreq` Struct Reference

Data Fields

- `uint8_t ueInIdle`
- `uint8_t freqsLen`
- `nas_infoInterFreq InfoInterfreq` [255]

8.77.1 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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_infoInterFreq for more information.

8.77.2 Field Documentation

8.77.2.1 `uint8_t nas_LTEInfoInterfreq::freqsLen`8.77.2.2 `nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]`8.77.2.3 `uint8_t nas_LTEInfoInterfreq::ueInIdle`

8.78 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]`

8.78.1 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 ue_in_idle 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 ue_in_idle 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.

8.78.2 Field Documentation

8.78.2.1 `nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]`

8.78.2.2 `uint8_t nas_LTEInfoIntrafreq::cellReselPriority`

8.78.2.3 `uint8_t nas_LTEInfoIntrafreq::cellsLen`

8.78.2.4 `uint16_t nas_LTEInfoIntrafreq::earfcn`

8.78.2.5 `uint32_t nas_LTEInfoIntrafreq::globalCellId`

8.78.2.6 `uint8_t nas_LTEInfoIntrafreq::plmn[3]`

8.78.2.7 `uint16_t nas_LTEInfoIntrafreq::servingCellId`

8.78.2.8 `uint8_t nas_LTEInfoIntrafreq::sIntraSearch`

8.78.2.9 `uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch`

8.78.2.10 `uint16_t nas_LTEInfoIntrafreq::tac`

8.78.2.11 `uint8_t nas_LTEInfoIntrafreq::threshServingLow`

8.78.2.12 `uint8_t nas_LTEInfoIntrafreq::ueInIdle`

8.79 nas_LTEInfoNeighboringGSM Struct Reference

Data Fields

- `uint8_t ueInIdle`
- `uint8_t freqsLen`
- `nas_LteGsmCellInfo LteGsmCellInfo` [255]

8.79.1 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 <code>nas_LteGsmCellInfo</code> for more information.

8.79.2 Field Documentation

8.79.2.1 `uint8_t nas_LTEInfoNeighboringGSM::freqsLen`

8.79.2.2 `nas_LteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo`[255]

8.79.2.3 `uint8_t nas_LTEInfoNeighboringGSM::ueInIdle`

8.80 nas_LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.80.1 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_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_IteWcdmaCellInfo for more information.

8.80.2 Field Documentation

8.80.2.1 `uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen`

8.80.2.2 `nas_IteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]`

8.80.2.3 `uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle`

8.81 nas_IteRsrpinformation Struct Reference

Data Fields

- `int16_t rsrplevel`

8.81.1 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).
------------------	--

8.81.2 Field Documentation

8.81.2.1 `int16_t nas_IteRsrpinformation::rsrplevel`

8.82 nas_LTERSRPThresh Struct Reference

Data Fields

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

8.82.1 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).

8.82.2 Field Documentation

8.82.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

8.82.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

8.83 nas_LTERSRQThresh Struct Reference

Data Fields

- uint8_t [LTERSRQThreshListLen](#)
- int16_t * [pLTERSRQThreshList](#)

8.83.1 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)

8.83.2 Field Documentation

8.83.2.1 `uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen`

8.83.2.2 `int16_t* nas_LTERSRQThresh::pLTERSRQThreshList`

8.84 nas_LTERSSIThresh Struct Reference

Data Fields

- `uint8_t LTERSSIThreshListLen`
- `int16_t * pLTERSSIThreshList`

8.84.1 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)

8.84.2 Field Documentation

8.84.2.1 `uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen`

8.84.2.2 `int16_t* nas_LTERSSIThresh::pLTERSSIThreshList`

8.85 nas_LTESigRptConfig Struct Reference

Data Fields

- `uint8_t rptRate`
- `uint8_t avgPeriod`

8.85.1 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

8.85.2 Field Documentation

8.85.2.1 `uint8_t nas_LTESigRptConfig::avgPeriod`8.85.2.2 `uint8_t nas_LTESigRptConfig::rptRate`8.86 `nas_IteSnrInformation` Struct Reference

Data Fields

- `int16_t snrlevel`

8.86.1 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.
-----------------	---

8.86.2 Field Documentation

8.86.2.1 int16_t nas_lteSnrinformation::snrlevel

8.87 nas_LTESNRThreshold Struct Reference

Data Fields

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

8.87.1 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).

8.87.2 Field Documentation

8.87.2.1 uint8_t nas_LTESNRThreshold::LTESNRThreshListLen

8.87.2.2 int16_t* nas_LTESNRThreshold::pLTESNRThreshList

8.88 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](#)

8.88.1 Detailed Description

Structure for storing the LTE System Information.

Parameters

<i>sysInfoLTE</i>	<ul style="list-style-type: none"> • See 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[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> • Mobile Country Code. • MCC digits in ASCII characters
<i>MNC[PLMN_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
------------	---

8.88.2 Field Documentation

8.88.2.1 `uint32_t nas_LTESysInfo::cellId`

8.88.2.2 `uint8_t nas_LTESysInfo::cellIdValid`

8.88.2.3 `uint16_t nas_LTESysInfo::lac`

8.88.2.4 `uint8_t nas_LTESysInfo::lacValid`

8.88.2.5 `uint8_t nas_LTESysInfo::MCC[3]`

8.88.2.6 `uint8_t nas_LTESysInfo::MNC[3]`

8.88.2.7 `uint8_t nas_LTESysInfo::networkIdValid`

8.88.2.8 `uint8_t nas_LTESysInfo::regRejectInfoValid`

8.88.2.9 `uint8_t nas_LTESysInfo::rejCause`

8.88.2.10 `uint8_t nas_LTESysInfo::rejectSrvDomain`

8.88.2.11 `nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE`

8.88.2.12 `uint16_t nas_LTESysInfo::tac`

8.88.2.13 `uint8_t nas_LTESysInfo::tacValid`

8.89 nas_lteWcdmaCellInfo Struct Reference

Data Fields

- `uint16_t uarfcn`
- `uint8_t cellReselPriority`
- `uint16_t threshXhigh`
- `uint16_t threshXlow`
- `uint8_t cellsLen`
- `nas_wcdmaCellInfo WCDMACellInfo [255]`

8.89.1 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 ue_in_idle 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 ue_in_idle 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 ue_in_idle 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 wcdmaCellInfo for more information.

8.89.2 Field Documentation

8.89.2.1 uint8_t nas_lteWcdmaCellInfo::cellReselPriority

8.89.2.2 uint8_t nas_lteWcdmaCellInfo::cellsLen

8.89.2.3 uint16_t nas_lteWcdmaCellInfo::threshXhigh

8.89.2.4 uint16_t nas_lteWcdmaCellInfo::threshXlow

8.89.2.5 uint16_t nas_lteWcdmaCellInfo::uarfcn

8.89.2.6 nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]

8.90 nas_MNRInfo Struct Reference

Data Fields

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

8.90.1 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

8.90.2 Field Documentation

8.90.2.1 `uint16_t nas_MNRInfo::mcc`

8.90.2.2 `uint16_t nas_MNRInfo::mnc`

8.90.2.3 `uint32_t nas_MNRInfo::rat`

8.91 `nas_netSelectionPref` Struct Reference

Data Fields

- `uint8_t netReg`
- `uint16_t mcc`
- `uint16_t mnc`

8.91.1 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

8.91.2 Field Documentation

8.91.2.1 uint16_t nas_netSelectionPref::mcc

8.91.2.2 uint16_t nas_netSelectionPref::mnc

8.91.2.3 uint8_t nas_netSelectionPref::netReg

8.92 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](#)

8.92.1 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[NAS_ - PLMN_LEN - H]</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>nmrLac</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>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

8.92.2 Field Documentation

8.92.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

8.92.2.2 uint8_t nas_nmrCellInfo::nmrBsic

8.92.2.3 uint32_t nas_nmrCellInfo::nmrCellID

8.92.2.4 uint16_t nas_nmrCellInfo::nmrLac

8.92.2.5 uint8_t nas_nmrCellInfo::nmrPlmn[3]

8.92.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

8.93 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

- uint8_t [leapSeconds](#)
- uint8_t [localTimeOffset](#)
- uint8_t [daylightSavings](#)

8.93.1 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)

8.93.2 Field Documentation

8.93.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.93.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.93.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

8.94 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]

8.94.1 Detailed Description

Parameters

<i>MCC</i>	Mobile Country Code
<i>MNC</i>	Mobile Network Code
<i>InUse</i>	current network or not
<i>Roaming</i>	Home/Roam Status of the Network
<i>Forbidden</i>	Network in the forbidden network list or not
<i>Preferred</i>	Network in the Preferred network list or not
<i>Description</i>	Network Name/Description

8.94.2 Field Documentation

8.94.2.1 `char nas_QmiNas3GppNetworkInfo::Description[255]`

8.94.2.2 `uint32_t nas_QmiNas3GppNetworkInfo::Forbidden`

8.94.2.3 `uint32_t nas_QmiNas3GppNetworkInfo::InUse`

8.94.2.4 `uint16_t nas_QmiNas3GppNetworkInfo::MCC`

8.94.2.5 `uint16_t nas_QmiNas3GppNetworkInfo::MNC`

8.94.2.6 `uint32_t nas_QmiNas3GppNetworkInfo::Preferred`

8.94.2.7 `uint32_t nas_QmiNas3GppNetworkInfo::Roaming`

8.95 nas_QmiNas3GppNetworkRAT Struct Reference

Data Fields

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

8.95.1 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

8.95.2 Field Documentation

8.95.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.95.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.95.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.96 nas_QmisNasPcsDigit Struct Reference

Data Fields

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

8.96.1 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

8.96.2 Field Documentation

8.96.2.1 `uint8_t nas_QmisNasPcsDigit::includes_pcs_digit`

8.96.2.2 `uint16_t nas_QmisNasPcsDigit::MCC`

8.96.2.3 `uint16_t nas_QmisNasPcsDigit::MNC`

8.97 nas_RejectReasonTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint32_t serviceDomain`
- `uint32_t rejectCause`

8.97.1 Detailed Description

Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>serviceDomain</i>	service domain
<i>rejectCause</i>	cause of reject

8.97.2 Field Documentation

8.97.2.1 `uint32_t nas_RejectReasonTlv::rejectCause`

8.97.2.2 `uint32_t nas_RejectReasonTlv::serviceDomain`

8.97.2.3 `uint8_t nas_RejectReasonTlv::TlvPresent`

8.98 nas_RFInfoTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t radioInterfaceSize`
- `uint32_t radioInterface` [255]
- `uint32_t activeBandClass` [255]
- `uint32_t activeChannel` [255]

8.98.1 Detailed Description

Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>radioInterface</i>	radio interface technology of the signal being measured

<i>activeBandClass</i>	active band class
<i>activeChannel</i>	active channel

8.98.2 Field Documentation

8.98.2.1 `uint32_t nas_RFInfoTlv::activeBandClass[255]`

8.98.2.2 `uint32_t nas_RFInfoTlv::activeChannel[255]`

8.98.2.3 `uint32_t nas_RFInfoTlv::radioInterface[255]`

8.98.2.4 `uint8_t nas_RFInfoTlv::radioInterfaceSize`

8.98.2.5 `uint8_t nas_RFInfoTlv::TlvPresent`

8.99 nas_roamIndList Struct Reference

Data Fields

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

8.99.1 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
----------------------	---

8.99.2 Field Documentation

8.99.2.1 `uint8_t nas_roamIndList::numInstances`

8.99.2.2 `uint8_t nas_roamIndList::radioInterface[32]`

8.99.2.3 `uint8_t nas_roamIndList::roamIndicator[32]`

8.100 nas_rsrqInformation Struct Reference

Data Fields

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

8.100.1 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>radioIf</i>	<ul style="list-style-type: none"> • Radio interface technology of the signal being measured <ul style="list-style-type: none"> – 0x08 – LTE

8.100.2 Field Documentation

8.100.2.1 `uint8_t nas_rsrqInformation::radioIf`

8.100.2.2 `int8_t nas_rsrqInformation::rsrq`

8.101 nas_rxSignalStrengthListElement Struct Reference

Data Fields

- `int16_t` [rxSignalStrength](#)
- `uint8_t` [radioIf](#)

8.101.1 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_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

Note

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

8.101.2 Field Documentation

8.101.2.1 `uint8_t nas_rxSignalStrengthListElement::radioIf`

8.101.2.2 `int16_t nas_rxSignalStrengthListElement::rxSignalStrength`

8.102 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]`

8.102.1 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_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

8.102.2 Field Documentation

8.102.2.1 uint8_t nas_servSystem::csAttachState

8.102.2.2 uint8_t nas_servSystem::numRadioInterfaces

8.102.2.3 uint8_t nas_servSystem::psAttachState

8.102.2.4 uint8_t nas_servSystem::radioInterface[32]

8.102.2.5 uint8_t nas_servSystem::regState

8.102.2.6 uint8_t nas_servSystem::selNetwork

8.103 nas_SignalStrengthTlv Struct Reference

Data Fields

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

8.103.1 Detailed Description

Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
-------------------	---

<i>signalStrength</i>	signal strength
<i>radioInterface</i>	radio interface technology of the signal being measured

8.103.2 Field Documentation

8.103.2.1 `uint32_t nas_SignalStrengthTlv::radioInterface`

8.103.2.2 `int8_t nas_SignalStrengthTlv::signalStrength`

8.103.2.3 `uint8_t nas_SignalStrengthTlv::TlvPresent`

8.104 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`

8.104.1 Detailed Description

Parameters

<i>rxSignalStrengthDelta</i>	RSSI delta(in dBm) at which an event report indication
<i>ecioDelta</i>	ecio delta
<i>ioDelta</i>	io delta
<i>sinrDelta</i>	sinr delta
<i>rsrqDelta</i>	rsrq delta
<i>ecioThresholdListLen</i>	
<i>ecioThresholdList</i>	
<i>sinrThresholdListLen</i>	
<i>sinrThresholdList</i>	
<i>lteSnrDelta</i>	lte snr delta
<i>lteRsrpDelta</i>	lte rsrp delta

8.104.2 Field Documentation

8.104.2.1 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta`

8.104.2.2 int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.104.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.104.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

8.104.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

8.104.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

8.104.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

8.104.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.104.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

8.104.2.10 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.104.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

8.105 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](#)

8.105.1 Detailed Description

Parameters

<i>rxSignalStrengthInfo</i>	signal strength info list
<i>ecioInfo</i>	ecio info list
<i>io</i>	received IO in dBm; IO is only applicable for 1xEV-DO
<i>sinr</i>	SINR level
<i>errorRateInfo</i>	error rate info
<i>rsrqInfo</i>	rsrq info
<i>lteSnrinfo</i>	lte Snr information
<i>lteRsrpinfo</i>	lte rsrp info

8.105.2 Field Documentation

8.105.2.1 nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo

8.105.2.2 nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo

8.105.2.3 uint32_t nas_SLQSSignalStrengthsInformation::io

8.105.2.4 `nas_lteRsrpInformation` `nas_SLQSSignalStrengthsInformation::lteRsrpInfo`

8.105.2.5 `nas_lteSnrinformation` `nas_SLQSSignalStrengthsInformation::lteSnrInfo`

8.105.2.6 `nas_rsrqInformation` `nas_SLQSSignalStrengthsInformation::rsrqInfo`

8.105.2.7 `nas_rxSignalStrengthListElement` `nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo`

8.105.2.8 `uint8_t` `nas_SLQSSignalStrengthsInformation::sinr`

8.106 `nas_SLQSSignalStrengthsTlv` Struct Reference

Data Fields

- `uint8_t` [TlvPresent](#)
- `nas_SLQSSignalStrengthsInformation` [sSLQSSignalStrengthsInfo](#)

8.106.1 Detailed Description

Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>sSLQSSignalStrengthsInfo</i>	signal strength info

8.106.2 Field Documentation

8.106.2.1 `nas_SLQSSignalStrengthsInformation` `nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo`

8.106.2.2 `uint8_t` `nas_SLQSSignalStrengthsTlv::TlvPresent`

8.107 `nas_SrvStatusInfo` Struct Reference

Data Fields

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

8.107.1 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

8.107.2 Field Documentation

8.107.2.1 uint8_t nas_SrvStatusInfo::isPrefDataPath

8.107.2.2 uint8_t nas_SrvStatusInfo::srvStatus

8.108 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](#)

8.108.1 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 - Invalid0x01 - Valid0xFF - Not Available
<i>roamStatus</i>	<ul style="list-style-type: none">Current roaming status.<ul style="list-style-type: none">0x00 - Off0x01 - On0x02 - Blinking0x03 - Out of the neighborhood0x04 - Out of the building0x05 - Preferred system0x06 - Available system0x07 - Alliance partner0x08 - Premium partner0x09 - Full service0x0A - Partial service0x0B - Banner is on0x0C - Banner is off0x0D to 0x3F - Reserved for Standard Enhanced Roaming Indicator Numbers0x40 to 0x7F - Reserved for Non-Standard Enhanced Roaming Indicator Numbers0x40 to 0xFF - Reserved.0xFF - Not AvailableValues 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

8.108.2 Field Documentation

8.108.2.1 `uint8_t nas_sysInfoCommon::isSysForbidden`

8.108.2.2 `uint8_t nas_sysInfoCommon::isSysForbiddenValid`

8.108.2.3 `uint8_t nas_sysInfoCommon::roamStatus`

8.108.2.4 `uint8_t nas_sysInfoCommon::roamStatusValid`

8.108.2.5 `uint8_t nas_sysInfoCommon::srvCapability`

8.108.2.6 `uint8_t nas_sysInfoCommon::srvCapabilityValid`

8.108.2.7 `uint8_t nas_sysInfoCommon::srvDomain`

8.108.2.8 `uint8_t nas_sysInfoCommon::srvDomainValid`

8.109 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

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

8.109.1 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.
-------------------------------	--

8.109.2 Field Documentation

8.109.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.109.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.110 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

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

8.110.1 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).

8.110.2 Field Documentation

8.110.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.110.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.111 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

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

8.111.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

Parameters

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

8.111.2 Field Documentation

8.111.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.111.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.112 nas_TDSCDMASINRThresh Struct Reference

Data Fields

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

8.112.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

Parameters

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

8.112.2 Field Documentation

8.112.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.112.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.113 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]

8.113.1 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[N-AS_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.

8.113.2 Field Documentation

8.113.2.1 `uint16_t nas_UMTSInfo::cellID`

8.113.2.2 `int16_t nas_UMTSInfo::ecio`

8.113.2.3 `uint8_t nas_UMTSInfo::geranInst`

8.113.2.4 `nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]`

8.113.2.5 `uint16_t nas_UMTSInfo::lac`

8.113.2.6 `uint8_t nas_UMTSInfo::plmn[3]`

8.113.2.7 `uint16_t nas_UMTSInfo::psc`

8.113.2.8 `int16_t nas_UMTSInfo::rscf`

8.113.2.9 `uint16_t nas_UMTSInfo::uarfcn`

8.113.2.10 `uint8_t nas_UMTSInfo::umtsInst`

8.113.2.11 `nas_UMTSInstInfo nas_UMTSInfo::UMTSInstInfo[255]`

8.114 nas_UMTSInstInfo Struct Reference

Data Fields

- `uint16_t umtsUarfcn`
- `uint16_t umtsPsc`
- `int16_t umtsRscf`
- `int16_t umtsEcio`

8.114.1 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).

8.114.2 Field Documentation

8.114.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.114.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.114.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.114.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.115 nas_umtsLTENbrCell Struct Reference

Data Fields

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

8.115.1 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_CELL_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.

8.115.2 Field Documentation

8.115.2.1 uint8_t nas_umtsLTENbrCell::cellsTDD

8.115.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.115.2.3 uint16_t nas_umtsLTENbrCell::pci

8.115.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.115.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.115.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.116 nas_wcdmaCellInfo Struct Reference

Data Fields

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

8.116.1 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.

8.116.2 Field Documentation

8.116.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.116.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.116.2.3 uint16_t nas_wcdmaCellInfo::psc

8.116.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.117 nas_WCDMAECIOThresh Struct Reference

Data Fields

- uint8_t [WCDMAECIOThreshListLen](#)
- int16_t * [pWCDMAECIOThreshList](#)

8.117.1 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)

8.117.2 Field Documentation

8.117.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.117.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

8.118 nas_WCDMAInfoLTENeighborCell Struct Reference

Data Fields

- uint32_t [wcdmaRRState](#)
- uint8_t [umtsLTENbrCellLen](#)
- [nas_umtsLTENbrCell](#) [UMTSLTENbrCell](#) [255]

8.118.1 Detailed Description

This structure contains information about the WCDMA - LTE Neighboring Cell Info Set.

Parameters

<i>wcdmaRR- 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.

8.118.2 Field Documentation

8.118.2.1 `nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]`

8.118.2.2 `uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen`

8.118.2.3 `uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCTest`

8.119 nas_WCDMARSSIThresh Struct Reference

Data Fields

- `uint8_t` [WCDMARSSIThreshListLen](#)
- `int16_t *` [pWCDMARSSIThreshList](#)

8.119.1 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)

8.119.2 Field Documentation

8.119.2.1 `int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList`

8.119.2.2 `uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen`

8.120 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](#)

8.120.1 Detailed Description

Structure for storing the WCDMA System Information.

Parameters

<i>sysInfoWCDMA</i>	<ul style="list-style-type: none"> • See 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]. <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[PLMN_LENGTH]</i>	<ul style="list-style-type: none"> Mobile Country Code. MCC digits in ASCII characters
<i>MNC[PLMN_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

8.120.2 Field Documentation

- 8.120.2.1 `uint32_t nas_WCDMA SysInfo::cellId`
- 8.120.2.2 `uint8_t nas_WCDMA SysInfo::cellIdValid`
- 8.120.2.3 `uint8_t nas_WCDMA SysInfo::hsCallStatus`
- 8.120.2.4 `uint8_t nas_WCDMA SysInfo::hsCallStatusValid`
- 8.120.2.5 `uint8_t nas_WCDMA SysInfo::hsInd`
- 8.120.2.6 `uint8_t nas_WCDMA SysInfo::hsIndValid`
- 8.120.2.7 `uint16_t nas_WCDMA SysInfo::lac`
- 8.120.2.8 `uint8_t nas_WCDMA SysInfo::lacValid`
- 8.120.2.9 `uint8_t nas_WCDMA SysInfo::MCC[3]`
- 8.120.2.10 `uint8_t nas_WCDMA SysInfo::MNC[3]`
- 8.120.2.11 `uint8_t nas_WCDMA SysInfo::networkIdValid`
- 8.120.2.12 `uint16_t nas_WCDMA SysInfo::psc`
- 8.120.2.13 `uint8_t nas_WCDMA SysInfo::pscValid`
- 8.120.2.14 `uint8_t nas_WCDMA SysInfo::regRejectInfoValid`
- 8.120.2.15 `uint8_t nas_WCDMA SysInfo::rejCause`
- 8.120.2.16 `uint8_t nas_WCDMA SysInfo::rejectSrvDomain`
- 8.120.2.17 `nas_sysInfoCommon nas_WCDMA SysInfo::sysInfoWCDMA`

8.121 NASBandPreferenceTlv Struct Reference

Data Fields

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

8.121.1 Field Documentation

8.121.1.1 uint64_t NASBandPreferenceTlv::band_pref

8.121.1.2 uint8_t NASBandPreferenceTlv::TlvPresent

8.122 NASEmergencyModeTlv Struct Reference

Data Fields

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

8.122.1 Field Documentation

8.122.1.1 uint8_t NASEmergencyModeTlv::EmerMode

8.122.1.2 uint8_t NASEmergencyModeTlv::TlvPresent

8.123 NasGetLTECphyCalInfo Struct Reference

Data Fields

- [NASPhyCaAggScellIndType](#) [PhyCaAggScellIndType](#)
- [NASPhyCaAggScellIDIBw](#) [PhyCaAggScellIDIBw](#)
- [NASPhyCaAggScellInfo](#) [PhyCaAggScellInfo](#)
- [NASPhyCaAggPcellInfo](#) [PhyCaAggPcellInfo](#)
- [NASPhyCaAggScellIndex](#) [PhyCaAggScellIndex](#)

8.123.1 Field Documentation

8.123.1.1 NASPhyCaAggPcellInfo NasGetLTECphyCalInfo::PhyCaAggPcellInfo

8.123.1.2 NASPhyCaAggScellIDIBw NasGetLTECphyCalInfo::PhyCaAggScellIDIBw

8.123.1.3 NASPhyCaAggScellIndex NasGetLTECphyCalInfo::PhyCaAggScellIndex

8.123.1.4 NASPhyCaAggScellIndType NasGetLTECphyCalInfo::PhyCaAggScellIndType

8.123.1.5 NASPhyCaAggScellInfo NasGetLTECphyCalInfo::PhyCaAggScellInfo

8.124 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

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

8.124.1 Field Documentation

8.124.1.1 `uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref`

8.124.1.2 `uint8_t NASGWAcqOrderPrefTlv::TlvPresent`

8.125 NASLTEBandPreferenceTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint64_t LTEBandPref`

8.125.1 Field Documentation

8.125.1.1 `uint64_t NASLTEBandPreferenceTlv::LTEBandPref`

8.125.1.2 `uint8_t NASLTEBandPreferenceTlv::TlvPresent`

8.126 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t nas_release`
- `uint8_t nas_major`
- `uint8_t nas_minor`

8.126.1 Field Documentation

8.126.1.1 `uint8_t NASLteNasReleaseInfoTlv::nas_major`

8.126.1.2 `uint8_t NASLteNasReleaseInfoTlv::nas_minor`

8.126.1.3 `uint8_t NASLteNasReleaseInfoTlv::nas_release`

8.126.1.4 `uint8_t NASLteNasReleaseInfoTlv::TlvPresent`

8.127 NASModePreferenceTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint16_t ModePref`

8.127.1 Field Documentation

8.127.1.1 `uint16_t NASModePreferenceTlv::ModePref`

8.127.1.2 `uint8_t NASModePreferenceTlv::TlvPresent`

8.128 NASNetSelPreferenceTlv Struct Reference

Data Fields

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

8.128.1 Field Documentation

8.128.1.1 uint8_t NASNetSelPreferenceTlv::NetSelPref

8.128.1.2 uint8_t NASNetSelPreferenceTlv::TlvPresent

8.129 NASOTAMessageTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint32_t [message_type](#)
- uint16_t [data_len](#)
- uint8_t [data_buf](#) [2048]

8.129.1 Field Documentation

8.129.1.1 uint8_t NASOTAMessageTlv::data_buf[2048]

8.129.1.2 uint16_t NASOTAMessageTlv::data_len

8.129.1.3 uint32_t NASOTAMessageTlv::message_type

8.129.1.4 uint8_t NASOTAMessageTlv::TlvPresent

8.130 NASPhyCaAggPcellInfo Struct Reference

Data Fields

- uint32_t [pci](#)
- uint32_t [freq](#)
- [LIBPACK_NAS_LTE_CPHY_CA_BW_NRB](#) dl_bw_value
- uint32_t [iLTEbandValue](#)
- uint8_t [TlvPresent](#)

8.130.1 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 for more information.
<i>scell_state</i>	<ul style="list-style-type: none"> Scell state Values. See NAS_LTE_CPHY_SCELL_STATE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present.

8.130.2 Field Documentation

8.130.2.1 LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

8.130.2.2 uint32_t NASPhyCaAggPcellInfo::freq

8.130.2.3 uint32_t NASPhyCaAggPcellInfo::iLTEbandValue

8.130.2.4 uint32_t NASPhyCaAggPcellInfo::pci

8.130.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

8.131 NASPhyCaAggScellIDIBw Struct Reference

Data Fields

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

8.131.1 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 for more information.
--------------------	--

8.131.2 Field Documentation

8.131.2.1 `LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellIDBw::dl_bw_value`

8.131.2.2 `uint8_t NASPhyCaAggScellIDBw::TlvPresent`

8.132 NASPhyCaAggScellIndex Struct Reference

Data Fields

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

8.132.1 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.

8.132.2 Field Documentation

8.132.2.1 `uint8_t NASPhyCaAggScellIndex::scell_idx`

8.132.2.2 `uint8_t NASPhyCaAggScellIndex::TlvPresent`

8.133 NASPhyCaAggScellIndType Struct Reference

Data Fields

- `uint32_t` [pci](#)
- `uint32_t` [freq](#)
- [LIBPACK_NAS_LTE_CPHY_SCELL_STATE](#) `scell_state`
- `uint8_t` [TlvPresent](#)

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

8.133.2 Field Documentation

8.133.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.133.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.133.2.3 LIBPACK_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.133.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.134 NASPhyCaAggScellInfo Struct Reference

Data Fields

- uint32_t [pci](#)
- uint32_t [freq](#)
- [LIBPACK_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value](#)
- uint32_t [iLTEbandValue](#)
- [LIBPACK_NAS_LTE_CPHY_SCELL_STATE scell_state](#)
- uint8_t [TlvPresent](#)

8.134.1 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 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 NAS_LTE_CPHY_SCELL_STATE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> Tlv Present.

8.134.2 Field Documentation

8.134.2.1 **LIBPACK_NAS_LTE_CPHY_CA_BW_NRB** NASPhyCaAggScellInfo::dl_bw_value

8.134.2.2 **uint32_t** NASPhyCaAggScellInfo::freq

8.134.2.3 **uint32_t** NASPhyCaAggScellInfo::iLTEbandValue

8.134.2.4 **uint32_t** NASPhyCaAggScellInfo::pci

8.134.2.5 **LIBPACK_NAS_LTE_CPHY_SCELL_STATE** NASPhyCaAggScellInfo::scell_state

8.134.2.6 **uint8_t** NASPhyCaAggScellInfo::TlvPresent

8.135 NASPRLPreferenceTlv Struct Reference

Data Fields

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

8.135.1 Field Documentation

8.135.1.1 **uint16_t** NASPRLPreferenceTlv::PRLPref

8.135.1.2 **uint8_t** NASPRLPreferenceTlv::TlvPresent

8.136 NASQmiCbkJnasSwtOTAMessageInd Struct Reference

Data Fields

- [NASOTAMessageTlv otaMsgTlv](#)
- [NASLteNasReleaseInfoTlv nasRelInfoTlv](#)
- [NASTimeInfoTlv timeTlv](#)

8.136.1 Field Documentation

8.136.1.1 **NASLteNasReleaseInfoTlv** NASQmiCbkNasSwtOTAMessageInd::nasRelInfoTlv

8.136.1.2 **NASOTAMessageTlv** NASQmiCbkNasSwtOTAMessageInd::otaMsgTlv

8.136.1.3 **NASTimeInfoTlv** NASQmiCbkNasSwtOTAMessageInd::timeTlv

8.137 NASQmiCbkNasSystemSelPrefInd Struct Reference

Data Fields

- [NASEmergencyModeTlv EMTlv](#)
- [NASModePreferenceTlv MPTlv](#)
- [NASBandPreferenceTlv BPTlv](#)
- [NASPRLPreferenceTlv PRLPTlv](#)
- [NASRoamPreferenceTlv RPTlv](#)
- [NASLTEBandPreferenceTlv LBPTlv](#)
- [NASNetSelPreferenceTlv NSPTlv](#)
- [NASServDomainPrefTlv SDPTlv](#)
- [NASGWAcqOrderPrefTlv GWAOPTlv](#)

8.137.1 Field Documentation

8.137.1.1 **NASBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::BPTlv

8.137.1.2 **NASEmergencyModeTlv** NASQmiCbkNasSystemSelPrefInd::EMTlv

8.137.1.3 **NASGWAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

8.137.1.4 **NASLTEBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::LBPTlv

8.137.1.5 **NASModePreferenceTlv** NASQmiCbkNasSystemSelPrefInd::MPTlv

8.137.1.6 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.137.1.7 **NASPRLPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.137.1.8 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv

8.137.1.9 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.138 NASRoamPreferenceTlv Struct Reference

Data Fields

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

8.138.1 Field Documentation

8.138.1.1 uint16_t NASRoamPreferenceTlv::RoamPref

8.138.1.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.139 NAServDomainPrefTlv Struct Reference

Data Fields

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

8.139.1 Field Documentation

8.139.1.1 uint32_t NAServDomainPrefTlv::SrvDomainPref

8.139.1.2 uint8_t NAServDomainPrefTlv::TlvPresent

8.140 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](#)

8.140.1 Detailed Description

This structure will hold the serving system parameters information

Parameters

<i>registrationState</i>	<div>- 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</div>
--------------------------	--

<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>	- HDR personality information (valid only for EVDO) <ul style="list-style-type: none"> • 0x00 - Unknown • 0x01 - HRPD • 0x02 - eHRPD

Note: None

8.140.2 Field Documentation

8.140.2.1 `uint8_t NAServingSystemInfo::csAttachState`

8.140.2.2 `uint8_t NAServingSystemInfo::hdrPersonality`

8.140.2.3 `uint8_t NAServingSystemInfo::psAttachState`

8.140.2.4 `uint8_t NAServingSystemInfo::radioInterfaceList[255]`

8.140.2.5 uint8_t NAServingSystemInfo::radioInterfaceNo

8.140.2.6 uint8_t NAServingSystemInfo::registrationState

8.140.2.7 uint8_t NAServingSystemInfo::selectedNetwork

8.141 NASTimeInfoTlv Struct Reference

Data Fields

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

8.141.1 Field Documentation

8.141.1.1 uint64_t NASTimeInfoTlv::time

8.141.1.2 uint8_t NASTimeInfoTlv::TlvPresent

8.142 newMTMessageTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [sMSMTMessageInfo](#) [MTMessageInfo](#)

8.142.1 Detailed Description

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

8.142.2 Field Documentation

8.142.2.1 [sMSMTMessageInfo](#) newMTMessageTlv::MTMessageInfo

8.142.2.2 uint8_t newMTMessageTlv::TlvPresent

8.143 pack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- uint8_t [cust_id](#) [64+1]
- uint8_t [list_type](#)
- uint16_t [Tlvresult](#)

8.143.1 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)
<i>list_type</i>	<ul style="list-style-type: none"> list type requested
<i>Tlvresult</i>	<ul style="list-style-type: none"> Pack Result

8.143.2 Field Documentation

8.143.2.1 `uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]`

8.143.2.2 `uint8_t pack_dms_GetCustFeaturesV2_t::list_type`

8.143.2.3 `uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult`

8.144 `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`

8.144.1 Field Documentation

8.144.1.1 `uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled`

8.144.1.2 `uint8_t pack_dms_SetCustFeature_t::DisableIMSI`

8.144.1.3 `uint32_t pack_dms_SetCustFeature_t::GpsEnable`

8.144.1.4 `uint8_t pack_dms_SetCustFeature_t::GPSPMP`

8.144.1.5 `uint8_t pack_dms_SetCustFeature_t::GPSSel`

8.144.1.6 `uint16_t pack_dms_SetCustFeature_t::IPFamSupport`

8.144.1.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

8.144.1.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

8.144.1.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

8.145 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](#)

8.145.1 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)
<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

8.145.2 Field Documentation

8.145.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.145.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.145.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.145.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.146 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t [mode](#)

8.146.1 Field Documentation

8.146.1.1 `uint8_t pack_dms_SetEventReport_t::mode`

8.147 `pack_dms_SetPower_t` Struct Reference

Data Fields

- `uint32_t mode`
- `uint16_t Tlvresult`

8.147.1 Field Documentation

8.147.1.1 `uint32_t pack_dms_SetPower_t::mode`

8.147.1.2 `uint16_t pack_dms_SetPower_t::Tlvresult`

8.148 `pack_dms_SetUSBComp_t` Struct Reference

Data Fields

- `uint8_t USBComp`
- `uint16_t Tlvresult`

8.148.1 Field Documentation

8.148.1.1 `uint16_t pack_dms_SetUSBComp_t::Tlvresult`

8.148.1.2 `uint8_t pack_dms_SetUSBComp_t::USBComp`

8.149 `pack_dms_SLQSDmsSwilIndicationRegister_t` Struct Reference

Data Fields

- `uint8_t resetInfoInd`

8.149.1 Detailed Description

Parameters

<code>resetInfoInd[IN]</code>	<ul style="list-style-type: none">• Values<ul style="list-style-type: none">– 0 - Disable– 1 - Enable
-------------------------------	--

8.149.2 Field Documentation

8.149.2.1 `uint8_t pack_dms_SLQSDmsSwilIndicationRegister_t::resetInfoInd`

8.150 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint8_t * [pDestSMSNum](#)
- uint8_t * [pDestSMSContent](#)

8.150.1 Detailed Description

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>pDestSMSContent</i> [IN]	<ul style="list-style-type: none"> • SMS Content as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.150.2 Field Documentation

8.150.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.150.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.151 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.151.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Pack result.
------------------	--

8.151.2 Field Documentation

8.151.2.1 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

8.152 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.152.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result
------------------	---

8.152.2 Field Documentation

8.152.2.1 `uint16_t pack_fms_GetImagesPreference_t::Tlvresult`

8.153 `pack_fms_GetStoredImages_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.153.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result
------------------	---

8.153.2 Field Documentation

8.153.2.1 `uint16_t pack_fms_GetStoredImages_t::Tlvresult`

8.154 `pack_fms_SetImagesPreference_t` Struct Reference

Data Fields

- `uint32_t` [imageListSize](#)
- `FMSPrefImageList * pImageList`
- `uint32_t` [bForceDownload](#)
- `uint8_t` [modemindex](#)
- `uint16_t` [Tlvresult](#)

8.154.1 Detailed Description

This structure contains the Set Images Preference pack

Parameters

<i>imageListSize</i>	<ul style="list-style-type: none"> • Image List Size
<i>pImageList</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 - Force Download.
<i>modemindex</i>	<ul style="list-style-type: none"> • Modem Index.
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result

8.154.2 Field Documentation

8.154.2.1 `uint32_t pack_fms_SetImagesPreference_t::bForceDownload`

8.154.2.2 `uint32_t pack_fms_SetImagesPreference_t::imageListSize`

8.154.2.3 `uint8_t pack_fms_SetImagesPreference_t::modemindex`

8.154.2.4 `FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList`

8.154.2.5 `uint16_t pack_fms_SetImagesPreference_t::Tlvresult`

8.155 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`

8.155.1 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.

8.155.2 Field Documentation

8.155.2.1 `loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo`8.155.2.2 `loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb`8.155.2.3 `loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo`8.155.2.4 `loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData`8.155.2.5 `loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo`8.155.2.6 `uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult`8.156 `pack_loc_EventRegister_t` Struct Reference

Data Fields

- `uint64_t` [eventRegister](#)
- `uint16_t` [Tlvresult](#)

8.156.1 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.).
Generated on Tue May 31 2016 14:24:35 for Linux QMI SDK by Doxygen	<ul style="list-style-type: none"> – 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 Q-

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.
------------------	--

8.156.2 Field Documentation

8.156.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

8.156.2.2 uint16_t pack_loc_EventRegister_t::Tlvresult

8.157 pack_loc_SetExtPowerState_t Struct Reference**Data Fields**

- uint32_t [extPowerState](#)
- uint16_t [Tlvresult](#)

8.157.1 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.

8.157.2 Field Documentation

8.157.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

8.157.2.2 uint16_t pack_loc_SetExtPowerState_t::Tlvresult

8.158 pack_loc_SetOperationMode_t Struct Reference**Data Fields**

- uint32_t [mode](#)

- uint16_t [Tlvresult](#)

8.158.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

<i>mode</i>	<ul style="list-style-type: none">• 0 - Default Mode.
<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result.

8.158.2 Field Documentation

8.158.2.1 uint32_t pack_loc_SetOperationMode_t::mode

8.158.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

8.159 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](#)

8.159.1 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"> • 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"> • 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"> • 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>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>LocApplication-Info</i>	<ul style="list-style-type: none"> • LOC Application Parameters • See loc_LocApplicationInfo for more information
<i>pConfigAltitude-Assumed</i>	<ul style="list-style-type: none"> • 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

8.159.2 Field Documentation

8.159.2.1 loc_LocApplicationInfo* pack_loc_Start_t::pApplicationInfo

8.159.2.2 uint32_t* pack_loc_Start_t::pConfigAltitudeAssumed

8.159.2.3 uint32_t* pack_loc_Start_t::pHorizontalAccuracyLvl

8.159.2.4 uint32_t* pack_loc_Start_t::pIntermediateReportState

8.159.2.5 uint32_t* pack_loc_Start_t::pMinIntervalTime

8.159.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

8.159.2.7 uint8_t pack_loc_Start_t::SessionId

8.159.2.8 uint16_t pack_loc_Start_t::Tlvresult

8.160 pack_loc_Stop_t Struct Reference

Data Fields

- uint8_t [SessionId](#)
- uint16_t [Tlvresult](#)

8.160.1 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.

8.160.2 Field Documentation

8.160.2.1 uint8_t pack_loc_Stop_t::SessionId

8.160.2.2 uint16_t pack_loc_Stop_t::Tlvresult

8.161 pack_nas_SetACCOLC_t Struct Reference

Data Fields

- int8_t [spc](#) [6]

- `uint8_t` [accolc](#)

8.161.1 Detailed Description

Parameters

<i>spc</i>	servcie programming code
<i>accolc</i>	accolc

8.161.2 Field Documentation

8.161.2.1 `uint8_t` `pack_nas_SetACCOLC_t::accolc`

8.161.2.2 `int8_t` `pack_nas_SetACCOLC_t::spc[6]`

8.162 `pack_nas_SetNetworkPreference_t` Struct Reference

Data Fields

- `uint32_t` [TechnologyPref](#)
- `uint32_t` [Duration](#)
- `uint16_t` [Tlvresult](#)

8.162.1 Detailed Description

Parameters

<i>TechnologyPref</i> [IN]	<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</i> [!N]	<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

8.162.2 Field Documentation

8.162.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

8.162.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

8.162.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

8.163 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint16_t [mcc](#)
- uint16_t [mnc](#)
- uint8_t * [pMncPcsStatus](#)

8.163.1 Detailed Description

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.

8.163.2 Field Documentation

8.163.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.163.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.163.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.164 pack_nas_SLQSIInitiateNetworkRegistration_t Struct Reference

Data Fields

- uint32_t [regAction](#)
- [nas_MNRInfo](#) * [pMNRInfo](#)
- uint32_t * [pChangeDuration](#)
- uint8_t * [pMncPcsDigitStatus](#)

8.164.1 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.
----------------------------	--

8.164.2 Field Documentation

8.164.2.1 uint32_t* pack_nas_SLQSIInitiateNetworkRegistration_t::pChangeDuration

8.164.2.2 uint8_t* pack_nas_SLQSIInitiateNetworkRegistration_t::pMncPcsDigitStatus

8.164.2.3 nas_MNRInfo* pack_nas_SLQSIInitiateNetworkRegistration_t::pMNRInfo

8.164.2.4 uint32_t pack_nas_SLQSIInitiateNetworkRegistration_t::regAction

8.165 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference

Data Fields

- nas_CDMARSSIThresh * pCDMARSSIThresh
- uint16_t * pCDMARSSIDelta
- nas_CDMAECIOThresh * pCDMAECIOThresh
- uint16_t * pCDMAECIODelta
- nas_HDRRSSIThresh * pHDRRSSIThresh
- uint16_t * pHDRRSSIDelta
- nas_HDRECIOThresh * pHDRECIOThresh
- uint16_t * pHDRECIODelta
- nas_HDRSINRThreshold * pHDRSINRThreshold
- uint16_t * pHDRSINRDelta
- nas_HDRIOThresh * pHDRIOThresh
- uint16_t * pHDRIODelta
- nas_GSMRSSIThresh * pGSMRSSIThresh
- uint16_t * pGSMRSSIDelta
- nas_WCDMARSSIThresh * pWCDMARSSIThresh
- uint16_t * pWCDMARSSIDelta
- nas_WCDMAECIOThresh * pWCDMAECIOThresh
- uint16_t * pWCDMAECIODelta
- nas_LTERSSIThresh * pLTERSSIThresh
- uint16_t * pLTERSSIDelta
- nas_LTESNRThreshold * pLTESNRThreshold
- uint16_t * pLTESNRDelta
- nas_LTERSRQThresh * pLTERSRQThresh
- uint16_t * pLTERSRQDelta
- nas_LTERSRPThresh * pLTERSRPThresh
- uint16_t * pLTERSRPDelta
- nas_LTESigRptConfig * pLTESigRptConfig
- nas_TDSCDMARSCPTThresh * pTDSCDMARSCPTThresh
- uint16_t * pTDSCDMARSCPDelta
- nas_TDSCDMARSSIThresh * pTDSCDMARSSIThresh
- float * pTDSCDMARSSIDelta
- nas_TDSCDMAECIOThresh * pTDSCDMAECIOThresh

- float * [pTDSCDMAECIODelta](#)
- [nas_TDSCDMASINRThresh](#) * [pTDSCDMASINRThresh](#)
- float * [pTDSCDMASINRDelta](#)

8.165.1 Detailed Description

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>pHDRRSSI- Thresh</i>	<ul style="list-style-type: none"> HDR RSSI Threshold List
<i>pHDRRSSIDelta</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>pHDRECI- 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 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 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>pLTERSNR- Thresh</i>	<ul style="list-style-type: none"> • LTE SNR Threshold List
<i>pLTERSNRDelta</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-CPThresh</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-OThresh</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.

8.165.2 Field Documentation

8.165.2.1 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta`

8.165.2.2 `nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh`

8.165.2.3 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta`

8.165.2.4 `nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh`

8.165.2.5 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta`

8.165.2.6 `nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh`

8.165.2.7 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta`

8.165.2.8 `nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh`

- 8.165.2.9 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta`
- 8.165.2.10 `nas_HDRIOTthresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOTthresh`
- 8.165.2.11 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIDelta`
- 8.165.2.12 `nas_HRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIThresh`
- 8.165.2.13 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRDelta`
- 8.165.2.14 `nas_HRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRThresh`
- 8.165.2.15 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta`
- 8.165.2.16 `nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh`
- 8.165.2.17 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta`
- 8.165.2.18 `nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh`
- 8.165.2.19 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta`
- 8.165.2.20 `nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh`
- 8.165.2.21 `nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig`
- 8.165.2.22 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta`
- 8.165.2.23 `nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh`
- 8.165.2.24 `float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta`
- 8.165.2.25 `nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh`
- 8.165.2.26 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta`
- 8.165.2.27 `nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh`
- 8.165.2.28 `float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta`
- 8.165.2.29 `nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh`
- 8.165.2.30 `float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta`
- 8.165.2.31 `nas_TDSCDMASINRThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh`
- 8.165.2.32 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta`
- 8.165.2.33 `nas_WCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh`
- 8.165.2.34 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta`
- 8.165.2.35 `nas_WCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh`

8.166 `pack_nas_SLQSNasIndicationRegisterExt_t` Struct Reference

Data Fields

- uint8_t * [pSystemSelectionInd](#)
- uint8_t * [pDDTMInd](#)
- uint8_t * [pServingSystemInd](#)
- uint8_t * [pDualStandByPrefInd](#)
- uint8_t * [pSubscriptionInfoInd](#)
- uint8_t * [pNetworkTimeInd](#)
- uint8_t * [pSysInfoInd](#)
- uint8_t * [pSignalStrengthInd](#)
- uint8_t * [pErrorRateInd](#)
- uint8_t * [pHDRNewUATIAssInd](#)
- uint8_t * [pHDRSessionCloseInd](#)
- uint8_t * [pManagedRoamingInd](#)
- uint8_t * [pLTECphyCa](#)

8.166.1 Detailed Description

Parameters

<i>pSystem- SelectionInd</i>	[Optional] <ul style="list-style-type: none"> • System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled. tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pDDTMInd</i>	[Optional] <ul style="list-style-type: none"> • DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled. tFNDDTM <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pServing- SystemInd</i>	[Optional] <ul style="list-style-type: none"> • Serving System indication registration. The following callbacks would not be invoked if the indication is disabled. tFNBandPreference <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 following callbacks would not be invoked if the indication is disabled. tFNDualStandByPref <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 following callbacks would not be invoked if the indication is disabled. tFNSubscriptionInfo <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 following callbacks would not be invoked if the indication is disabled. tFNNetworkTime <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 following callbacks would not be invoked if the indication is disabled. tFNSysInfo <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 following callbacks would not be invoked if the indication is disabled. tFNSigInfo <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 following callbacks would not be invoked if the indication is disabled. tFNErrRate <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 following callbacks would not be invoked if the indication is disabled. tFNHDRUATIUpdate <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pHDRSession-CloseInd</i>	[Optional] <ul style="list-style-type: none"> HDR Session Closed indication registration. The following callbacks would not be invoked if the indication is disabled. tFNHDRSessionClose <ul style="list-style-type: none"> – 0x00 - Disable – 0x01 - Enable
<i>pManaged-RoamingInd</i>	[Optional] <ul style="list-style-type: none"> Managed Roaming indication registration. The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming <ul style="list-style-type: none"> – 0x00 - Disable – 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. tFNManagedRoaming <ul style="list-style-type: none"> – 0x00 - Disable (default value) – 0x01 - Enable

8.166.2 Field Documentation

8.166.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.166.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.166.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.166.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

- 8.166.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd
- 8.166.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa
- 8.166.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd
- 8.166.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd
- 8.166.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd
- 8.166.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd
- 8.166.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd
- 8.166.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd
- 8.166.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.167 pack_nas_SLQSNasSwiOTAMessageCallback_t Struct Reference

Data Fields

- uint8_t [lteEsmUI](#)
- uint8_t [lteEsmDI](#)
- uint8_t [lteEmmUI](#)
- uint8_t [lteEmmDI](#)
- uint8_t [gsmUmtsUI](#)
- uint8_t [gsmUmtsDI](#)
- uint8_t * [pRankIndicatorInd](#)

8.167.1 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>pRankIndicatorInd</i>	<ul style="list-style-type: none"> • 0 - do not report • 1 - report Rank Indicator messages

8.167.2 Field Documentation

8.167.2.1 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsDI

8.167.2.2 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsUI

8.167.2.3 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmDI

8.167.2.4 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmUI

8.167.2.5 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmDI

8.167.2.6 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmUI

8.167.2.7 uint8_t* pack_nas_SLQSNasSwiOTAMessageCallback_t::pRankIndicatorInd

8.168 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

Data Fields

- uint8_t [bEnable](#)
- [nas_SLQSSignalStrengthsIndReq](#) * [pSigIndReq](#)

8.168.1 Detailed Description

Parameters

<i>bEnable</i>	0/1 to disable/enable RSSI signal strength indication
<i>pSigIndReq</i>	parameters to control signal strength indication

8.168.2 Field Documentation

8.168.2.1 `uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable`

8.168.2.2 `nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq`

8.169 `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`

8.169.1 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

<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
	<ul style="list-style-type: none"> – 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

<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 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

8.169.2 Field Documentation

8.169.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.169.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

- 8.169.2.3 `uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration`
- 8.169.2.4 `struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID`
- 8.169.2.5 `uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode`
- 8.169.2.6 `uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref`
- 8.169.2.7 `uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref`
- 8.169.2.8 `uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat`
- 8.169.2.9 `uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref`
- 8.169.2.10 `struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref`
- 8.169.2.11 `uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref`
- 8.169.2.12 `unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT`
- 8.169.2.13 `uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref`
- 8.169.2.14 `uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref`
- 8.169.2.15 `uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction`
- 8.169.2.16 `uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdsdmaBandPref`

8.170 pack_qmi_t Struct Reference

Data Fields

- `uint16_t xid`
- `int timeout`
- `uint16_t msgid`
- `uint8_t svc`

8.170.1 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

8.170.2 Field Documentation

- 8.170.2.1 `uint16_t pack_qmi_t::msgid`
- 8.170.2.2 `uint8_t pack_qmi_t::svc`

8.170.2.3 int pack_qmi_t::timeout

8.170.2.4 uint16_t pack_qmi_t::xid

8.171 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.171.1 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
-------------------	--

8.171.2 Field Documentation

8.171.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.172 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.172.1 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
-------------------	--

8.172.2 Field Documentation

8.172.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.173 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t [enable](#)

8.173.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

Parameters

<i>enable</i> [IN]	<ul style="list-style-type: none"> • 1 - Enable QoS event reporting • 0 - Disable QoS event reporting
--------------------	---

8.173.2 Field Documentation

8.173.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.174 pack_sms_SendSMS_t Struct Reference

Data Fields

- uint32_t [messageFormat](#)
- uint32_t [messageSize](#)
- uint8_t * [pMessage](#)
- uint8_t * [pLinktimer](#)

8.174.1 Detailed Description

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

8.174.2 Field Documentation

8.174.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.174.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.174.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.174.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.175 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum [eqmiCbkJSetStatus](#) *status*

8.175.1 Detailed Description

Parameters

<i>status</i>	callback parameter
---------------	--------------------

8.175.2 Field Documentation

8.175.2.1 enum [eqmiCbkJSetStatus](#) *pack_sms_SetNewSMSCallback_t::status*

8.176 pack_sms_SLQSDDeleteSMS_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t * [pMessageIndex](#)
- uint32_t * [pMessageTag](#)
- uint8_t * [pMessageMode](#)

8.176.1 Detailed Description

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)
---------------------	--

8.176.2 Field Documentation

8.176.2.1 uint32_t* pack_sms_SLQSDelateSMS_t::pMessageIndex

8.176.2.2 uint8_t* pack_sms_SLQSDelateSMS_t::pMessageMode

8.176.2.3 uint32_t* pack_sms_SLQSDelateSMS_t::pMessageTag

8.176.2.4 uint32_t pack_sms_SLQSDelateSMS_t::storageType

8.177 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t [messageIndex](#)
- uint8_t * [pMessageMode](#)

8.177.1 Detailed Description

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)

8.177.2 Field Documentation

8.177.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.177.2.2 `uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode`

8.177.2.3 `uint32_t pack_sms_SLQSGetSMS_t::storageType`

8.178 `pack_sms_SLQSGetSMSList_t` Struct Reference

Data Fields

- `uint32_t storageType`
- `uint32_t * pRequestedTag`
- `uint8_t * pMessageMode`

8.178.1 Detailed Description

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)

8.178.2 Field Documentation

8.178.2.1 `uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode`

8.178.2.2 `uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag`

8.178.2.3 `uint32_t pack_sms_SLQSGetSMSList_t::storageType`

8.179 `pack_sms_SLQSModifySMSStatus_t` Struct Reference

Data Fields

- `uint32_t storageType`
- `uint32_t messageIndex`
- `uint32_t messageTag`

- `uint8_t * pMessageMode`

8.179.1 Detailed Description

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)

8.179.2 Field Documentation

8.179.2.1 `uint32_t pack_sms_SLQSMModifySMSStatus_t::messageIndex`

8.179.2.2 `uint32_t pack_sms_SLQSMModifySMSStatus_t::messageTag`

8.179.2.3 `uint8_t* pack_sms_SLQSMModifySMSStatus_t::pMessageMode`

8.179.2.4 `uint32_t pack_sms_SLQSMModifySMSStatus_t::storageType`

8.180 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`

8.180.1 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
---------------------	---

8.180.2 Field Documentation

8.180.2.1 uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate

8.180.2.2 uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type

8.180.2.3 uint8_t pack_swiloc_SwiLocSetAutoStart_t::function

8.180.2.4 uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist

8.180.2.5 uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time

8.180.2.6 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate

8.180.2.7 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type

8.180.2.8 int pack_swiloc_SwiLocSetAutoStart_t::set_function

8.180.2.9 int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist

8.180.2.10 int pack_swiloc_SwiLocSetAutoStart_t::set_max_time

8.181 pack_swioama_SLQSOMADMCancelSession_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.181.1 Detailed Description

Structure that contains the session type for OMA cancel session command

Parameters

<i>sessionType[IN]</i>	<ul style="list-style-type: none"> • Session type <ul style="list-style-type: none"> – 0x01 - FOTA, to check availability of FW Update – 0xFF - Cancel any active OMADM session
------------------------	---

8.181.2 Field Documentation

8.181.2.1 uint32_t pack_swioama_SLQSOMADMCancelSession_t::sessionType

8.182 pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t [SessionType](#)

8.182.1 Detailed Description

Structure that contains the session type for OMA get session info command

Parameters

<i>SessionType</i> [IN]	<ul style="list-style-type: none">• 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
-------------------------	--

8.182.2 Field Documentation

8.182.2.1 uint32_t pack_swima_SLQSOMADMGetSessionInfo_t::SessionType

8.183 pack_swima_SLQSOMADMSelectSelection_t Struct Reference

Data Fields

- uint32_t [selection](#)
- uint32_t * [pDeferTime](#)
- uint32_t * [pRejectReason](#)

8.183.1 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.

8.183.2 Field Documentation

8.183.2.1 uint32_t* pack_swima_SLQSOMADMSendSelection_t::pDeferTime

8.183.2.2 uint32_t* pack_swima_SLQSOMADMSendSelection_t::pRejectReason

8.183.2.3 uint32_t pack_swima_SLQSOMADMSendSelection_t::selection

8.184 pack_swima_SLQSOMADMSetSettings_t Struct Reference

Data Fields

- uint8_t [FOTAdownload](#)
- uint8_t [FOTAUpdate](#)
- uint8_t * [pAutosdm](#)
- uint8_t * [pFwAutoCheck](#)

8.184.1 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> [IN]	<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> [-IN]	<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

8.184.2 Field Documentation

8.184.2.1 uint8_t pack_swisma_SLQSOMADMSetSettings_t::FOTAdownload

8.184.2.2 uint8_t pack_swisma_SLQSOMADMSetSettings_t::FOTAUpdate

8.184.2.3 uint8_t* pack_swisma_SLQSOMADMSetSettings_t::pAutosdm

8.184.2.4 uint8_t* pack_swisma_SLQSOMADMSetSettings_t::pFwAutoCheck

8.185 pack_swisma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.185.1 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
-------------------------	---

8.185.2 Field Documentation

8.185.2.1 uint32_t pack_swisma_SLQSOMADMStartSession_t::sessionType

8.186 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](#)

8.186.1 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 UIMSessionInformation for more information.
<i>changePIN</i>	<ul style="list-style-type: none"> • See 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.

8.186.2 Field Documentation

8.186.2.1 [uim_changeUIMPIN](#) `pack_uim_ChangePin_t::changePIN`

8.186.2.2 [uim_encryptedPIN1](#) `pack_uim_ChangePin_t::EncryptedPIN1`

8.186.2.3 [uint32_t*](#) `pack_uim_ChangePin_t::pIndicationToken`

8.186.2.4 uint8_t* pack_uim_ChangePin_t::pKeyReferenceID

8.186.2.5 uim_sessionInformation pack_uim_ChangePin_t::sessionInfo

8.186.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

8.187 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](#)

8.187.1 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 UIMSessionInformation for more information.
<i>fileIndex</i>	<ul style="list-style-type: none"> • See fileInfo for more information.
<i>readTransparent</i>	<ul style="list-style-type: none"> • See 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.

8.187.2 Field Documentation

8.187.2.1 `uim_fileInfo` `pack_uim_ReadTransparent_t::fileIndex`

8.187.2.2 `uint8_t*` `pack_uim_ReadTransparent_t::pEncryptData`

8.187.2.3 `uint32_t*` `pack_uim_ReadTransparent_t::pIndicationToken`

8.187.2.4 `uim_readTransparentInfo` `pack_uim_ReadTransparent_t::readTransparent`

8.187.2.5 `uim_sessionInformation` `pack_uim_ReadTransparent_t::sessionInfo`

8.187.2.6 `uint16_t` `pack_uim_ReadTransparent_t::Tlvresult`

8.188 `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`

8.188.1 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.

8.188.2 Field Documentation

8.188.2.1 `uim_encryptedPIN1` `pack_uim_SetPinProtection_t::EncryptedPIN1`

8.188.2.2 `uint32_t*` `pack_uim_SetPinProtection_t::pIndicationToken`

8.188.2.3 `uim_setPINProtection` `pack_uim_SetPinProtection_t::pinProtection`

8.188.2.4 `uint8_t*` `pack_uim_SetPinProtection_t::pKeyReferenceID`

8.188.2.5 `uim_sessionInformation` `pack_uim_SetPinProtection_t::sessionInfo`

8.188.2.6 `uint16_t` `pack_uim_SetPinProtection_t::Tlvresult`

8.189 pack_uim_SLQSUIEventRegister_t Struct Reference

Data Fields

- `uint32_t` [eventMask](#)

8.189.1 Detailed Description

Parameters

<i>eventMask</i>	<ul style="list-style-type: none"> - bit 1 - card status • bit 4 - physical slot status
------------------	---

8.189.2 Field Documentation

8.189.2.1 uint32_t pack_uim_SLQSUIEventRegister_t::eventMask

8.190 pack_uim_SLQSUISSwitchSlot_t Struct Reference

Data Fields

- uint8_t [bLogicalSlot](#)
- uint32_t [ulPhysicalSlot](#)

8.190.1 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

8.190.2 Field Documentation

8.190.2.1 uint8_t pack_uim_SLQSUISSwitchSlot_t::bLogicalSlot

8.190.2.2 uint32_t pack_uim_SLQSUISSwitchSlot_t::ulPhysicalSlot

8.191 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

8.191.1 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

8.191.2 Field Documentation

8.191.2.1 `uim_encryptedPIN1` `pack_uim_UnblockPin_t::EncryptedPIN1`

8.191.2.2 `uint32_t*` `pack_uim_UnblockPin_t::pIndicationToken`

8.191.2.3 `uim_unblockUIMPIN` `pack_uim_UnblockPin_t::pinProtection`

8.191.2.4 `uint8_t*` `pack_uim_UnblockPin_t::pKeyReferenceID`

8.191.2.5 `uim_sessionInformation` `pack_uim_UnblockPin_t::sessionInfo`

8.191.2.6 `uint16_t` `pack_uim_UnblockPin_t::Tlvresult`

8.192 `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`

8.192.1 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 UIMSessionInformation for more information.
<i>verifyPIN</i>	<ul style="list-style-type: none"> • See verifyUIMPIN for more information.
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See 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.

8.192.2 Field Documentation

8.192.2.1 uim_encryptedPIN1* pack_uim_VerifyPin_t::pEncryptedPIN1

8.192.2.2 uint32_t* pack_uim_VerifyPin_t::pIndicationToken

8.192.2.3 uint8_t* pack_uim_VerifyPin_t::pKeyReferenceID

8.192.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

8.192.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

8.192.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

8.193 pack_wds_GetDefaultProfile_t Struct Reference

Data Fields

- uint32_t [profiletype](#)

8.193.1 Detailed Description

Parameters

<i>profiletype</i>	profile type
--------------------	--------------

8.193.2 Field Documentation

8.193.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.194 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [family](#)

8.194.1 Detailed Description

Parameters

<i>type</i>	profile type <ul style="list-style-type: none"> • 0 - 3GPP • 1 - 3GPP2
<i>type</i>	profile family <ul style="list-style-type: none"> • 0 - Embedded • 1 - Tethered

8.194.2 Field Documentation

8.194.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

8.194.2.2 uint8_t pack_wds_GetDefaultProfileNum_t::type

8.195 pack_wds_GetDormancyState_t Struct Reference

8.196 pack_wds_GetLastMobileIPError_t Struct Reference

8.197 pack_wds_GetMobileIP_t Struct Reference

8.198 pack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t [index](#)

8.198.1 Detailed Description

Parameters

<i>index</i>	mobile ip profile identifier
--------------	------------------------------

8.198.2 Field Documentation

8.198.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

8.199 pack_wds_GetPacketStatus_t Struct Reference

Data Fields

- uint32_t [statmask](#)

8.199.1 Detailed Description

Parameters

<i>statmask</i>	packet statistics mask
-----------------	------------------------

8.199.2 Field Documentation

8.199.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.200 pack_wds_GetSessionDuration_t Struct Reference

8.201 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- [rmTrasnferStaticsReq](#) RmTrasnferStaticsReq

8.201.1 Detailed Description

Parameters

rmTrasnferStaticsReq	RM Transfer Statistics Indicator
--------------------------------------	----------------------------------

8.201.2 Field Documentation

8.201.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

8.202 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](#)

8.202.1 Detailed Description

Parameters

<i>profileType</i>	profile type
<i>pdpType</i>	Packet Data Protocol (PDP) type
<i>ipAddress</i>	ip address
<i>primaryDNS</i>	primary dns
<i>secondaryDNS</i>	secondry dns
<i>authentication</i>	authentication type
<i>name</i>	name of the profile
<i>apnname</i>	apn name
<i>username</i>	username of the profile
<i>password</i>	password of profile

8.202.2 Field Documentation

8.202.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

8.202.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

8.202.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

8.202.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

8.202.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

8.202.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

8.202.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

8.202.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

8.202.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

8.202.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

8.203 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [family](#)
- uint8_t [index](#)

8.203.1 Field Documentation

8.203.1.1 `uint8_t pack_wds_SetDefaultProfileNum_t::family`

8.203.1.2 `uint8_t pack_wds_SetDefaultProfileNum_t::index`

8.203.1.3 `uint8_t pack_wds_SetDefaultProfileNum_t::type`

8.204 `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`

8.204.1 Detailed Description

Parameters

<i>spc</i>	service programming code string
<i>index</i>	Index of the profile to modify
<i>pEnabled</i>	Enable profile 0-disable nonzero enable
<i>pAddress</i>	Home IPv4 address
<i>pPrimaryHA</i>	Primary home agent IPv4 address
<i>pSecondaryHA</i>	secondary home agent IPv4 address
<i>pRevTunneling</i>	Enable reverse tunneling 0-disable nonzero enable
<i>pNAI</i>	Network access identifier string
<i>pHASPI</i>	Home agent security parameter index
<i>pAAASPI</i>	AAA server security parameter index
<i>pMNHA</i>	MN-HA key string
<i>pMNAAA</i>	MN-AAA key string

8.204.2 Field Documentation

8.204.2.1 `uint8_t pack_wds_SetMobileIPProfile_t::index`

8.204.2.2 `uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI`

8.204.2.3 `uint32_t* pack_wds_SetMobileIPProfile_t::pAddress`

8.204.2.4 `uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled`

8.204.2.5 `uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI`

8.204.2.6 int8_t* pack_wds_SetMobilePPProfile_t::pMNAAA

8.204.2.7 int8_t* pack_wds_SetMobilePPProfile_t::pMNHA

8.204.2.8 int8_t* pack_wds_SetMobilePPProfile_t::pNAI

8.204.2.9 uint32_t* pack_wds_SetMobilePPProfile_t::pPrimaryHA

8.204.2.10 uint8_t* pack_wds_SetMobilePPProfile_t::pRevTunneling

8.204.2.11 uint32_t* pack_wds_SetMobilePPProfile_t::pSecondaryHA

8.204.2.12 int8_t pack_wds_SetMobilePPProfile_t::spc[10]

8.205 pack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

- uint8_t * [pProfileId](#)
- uint8_t * [pProfileType](#)
- [wds_profileInfo](#) * [pCurProfile](#)

8.205.1 Detailed Description

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile • 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 – NULL is not allowed
<i>curProfile</i>	<ul style="list-style-type: none"> • union of 3GPP and 3GPP2 profile

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

8.205.2 Field Documentation

8.205.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

8.205.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

8.205.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

8.206 pack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint8_t [profileType](#)
- uint8_t [profileIndex](#)

8.206.1 Detailed Description

Parameters

<i>profileType</i>	profile type
<i>profileIndex</i>	profile index

8.206.2 Field Documentation

8.206.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.206.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.207 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.208 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.209 pack_wds_SLQSGetDUNCallInfo_t Struct Reference

Data Fields

- uint32_t [Mask](#)
- uint8_t * [pReportConnStatus](#)
- [transferStatInd](#) * [pTransferStatInd](#)
- uint8_t * [pReportDormStatus](#)
- uint8_t * [pReportDataBearerTech](#)
- uint8_t * [pReportChannelRate](#)

8.209.1 Detailed Description

Parameters

<i>Mask</i>	mask bits corresponding to the information requested to 1
<i>pReportConn- Status</i>	Connect Status Indicator
<i>pTransferStatInd</i>	Transfer Statistics Indicator
<i>pReportDorm- Status</i>	Dormancy Status Indicator

<i>pReportData-BearerTech</i>	Current Data Bearer Technology Indicator
<i>pReport-ChannelRate</i>	Channel Rate Indicator

8.209.2 Field Documentation

8.209.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.209.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.209.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.209.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.209.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.209.2.6 transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

8.210 pack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

- uint8_t [ProfileId](#)
- uint8_t [ProfileType](#)

8.210.1 Detailed Description

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile • 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

8.210.2 Field Documentation

8.210.2.1 `uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId`

8.210.2.2 `uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType`

8.211 `pack_wds_SLQSGetRuntimeSettings_t` Struct Reference

Data Fields

- `uint32_t * pReqSettings`

8.211.1 Detailed Description

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
---------------------	--

8.211.2 Field Documentation

8.211.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

8.212 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint8_t * pProfileId
- uint8_t * pProfileType
- wds_profileInfo curProfile

8.212.1 Detailed Description

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile • 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 – NULL is not allowed
<i>curProfile</i>	<ul style="list-style-type: none"> • union of 3GPP and 3GPP2 profile

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

8.212.2 Field Documentation

8.212.2.1 wds_profileInfo pack_wds_SLQSModifyProfile_t::curProfile

8.212.2.2 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileId

8.212.2.3 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileType

8.213 pack_wds_SLQSSet3GPPConfigItem_t Struct Reference

Data Fields

- uint16_t profileList [5]
- uint8_t defaultPDNEnabled
- uint8_t _3gppRelease

- uint16_t [LTEAttachProfileList](#) [24]
- uint16_t [LTEAttachProfileListLen](#)

8.213.1 Detailed Description

Parameters

<i>profileList</i>	Profile List
<i>defaultPDN-Enabled</i>	<ul style="list-style-type: none"> • 0 - disabled • 1 - enabled
<i>_3gppRelease</i>	3GPP release <ul style="list-style-type: none"> • 0 - Release_99 • 1 - Release_5 • 2 - Release_6 • 3 - Release_7 • 4 - Release_8
<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.
<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

8.213.2 Field Documentation

8.213.2.1 uint8_t pack_wds_SLQSSet3GPPConfigItem_t::_3gppRelease

8.213.2.2 uint8_t pack_wds_SLQSSet3GPPConfigItem_t::defaultPDNEnabled

8.213.2.3 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileList[24]

8.213.2.4 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.213.2.5 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::profileList[5]

8.214 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint8_t [IPFamilyPreference](#)

8.214.1 Detailed Description

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
----------------------------	---

8.214.2 Field Documentation

8.214.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

8.215 pack_wds_SLQSSetWdsEventCallback_t Struct Reference

Data Fields

- uint8_t [dataBearer](#)
- uint8_t [dormancyStatus](#)
- uint8_t [mobileIP](#)
- uint8_t [transferStats](#)
- uint8_t [currentDataBearer](#)
- uint8_t [dataSystemStatus](#)
- uint8_t [interval](#)

8.215.1 Detailed Description

Parameters

<i>dataBearer</i>	data bearer
<i>dormancyStatus</i>	dormancy status
<i>mobileIP</i>	mobile IP
<i>currentData-Bearer</i>	current data bearer
<i>dataSystem-Status</i>	data system status
<i>interval</i>	interval

8.215.2 Field Documentation

8.215.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

8.215.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

8.215.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

8.215.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

8.215.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

8.215.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

8.215.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

8.216 pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

- [wdsDhcpv4ProfileId](#) * [pProfileId](#)

8.216.1 Detailed Description

Parameters

<i>pProfileId</i>	pointer to Profile Id structure
-------------------	---------------------------------

8.216.2 Field Documentation

8.216.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSetDHCPv4ClientConfig_t::pProfileId

8.217 pack_wds_SLQSSetDataSession_t Struct Reference

Data Fields

- uint8_t * [pTech](#)
- uint32_t * [pprofileid3gpp](#)
- uint32_t * [pprofileid3gpp2](#)
- uint32_t * [pAuth](#)
- char * [pUser](#)
- char * [pPass](#)

8.217.1 Detailed Description

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 3GPPs 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

8.217.2 Field Documentation

8.217.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

8.217.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

8.217.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

8.217.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

8.217.2.5 uint8_t* pack_wds_SLQSStartDataSession_t::pTech

8.217.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

8.218 pack_wds_SLQSStopDataSession_t Struct Reference

Data Fields

- uint32_t * [psid](#)

8.218.1 Detailed Description

Parameters

<i>sid</i>	session id
------------	------------

8.218.2 Field Documentation

8.218.2.1 `uint32_t*` `pack_wds_SLQSSStopDataSession_t::psid`

8.219 `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t` Struct Reference

Data Fields

- `uint8_t` [contextId](#)
- `uint8_t` [contextType](#)

8.219.1 Detailed Description

Parameters

<i>contextId</i>	Context Identifier
<i>contextType</i>	Context Type 0-3GPP 1-3GPP2

8.219.2 Field Documentation

8.219.2.1 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.219.2.2 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType`

8.220 `PackCreateProfileOut` Struct Reference

Data Fields

- `uint8_t` [ProfileType](#)
- `uint8_t` [ProfileIndex](#)
- `uint16_t` [ExtErrorCode](#)

8.220.1 Field Documentation

8.220.1.1 `uint16_t` `PackCreateProfileOut::ExtErrorCode`

8.220.1.2 `uint8_t` `PackCreateProfileOut::ProfileIndex`

8.220.1.3 `uint8_t` `PackCreateProfileOut::ProfileType`

8.221 `packgetDyingGaspCfg` Struct Reference

Data Fields

- `uint8_t *` [pDestSMSNum](#)
- `uint8_t *` [pDestSMSCContent](#)

8.221.1 Detailed Description

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>pDestSMS-Content</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.

8.221.2 Field Documentation

8.221.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.221.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.222 packgetDyingGaspStatistics Struct Reference

Data Fields

- uint32_t * [pTimeStamp](#)
- uint8_t * [pSMSAttemptedFlag](#)

8.222.1 Detailed Description

Parameters

<i>TimeStamp</i> [<i>OUT</i>]	<ul style="list-style-type: none"> • Time Stamp.
<i>SMSAttempted-Flag</i> [<i>OUT</i>]	<ul style="list-style-type: none"> • SMS Attempted Flag.

8.222.2 Field Documentation

8.222.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.222.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

8.223 qmiSmsMessageList Struct Reference

Data Fields

- uint32_t [messageIndex](#)
- uint32_t [messageTag](#)

8.223.1 Detailed Description

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

8.223.2 Field Documentation

8.223.2.1 uint32_t qmiSmsMessageList::messageIndex

8.223.2.2 uint32_t qmiSmsMessageList::messageTag

8.224 qmiWSDDataBearerTechnology Struct Reference

Data Fields

- uint8_t [currentNetwork](#)
- uint32_t [ratMask](#)
- uint32_t [soMask](#)

8.224.1 Detailed Description

Parameters

<i>currentNetwork</i>	current selected network
<i>Radio</i>	Access Technology (RAT) mask
<i>soMask</i>	Service Option (SO) mask

8.224.2 Field Documentation

8.224.2.1 uint8_t qmiWSDDataBearerTechnology::currentNetwork

8.224.2.2 uint32_t qmiWSDDataBearerTechnology::ratMask

8.224.2.3 uint32_t qmiWSDDataBearerTechnology::soMask

8.225 RFBandInfoElements Struct Reference

Data Fields

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

8.225.1 Detailed Description

Parameters

<i>radioInterface</i>	radio interface technology
<i>activeBandClass</i>	active band class
<i>activeChannel</i>	active channel

8.225.2 Field Documentation

8.225.2.1 uint16_t RFBandInfoElements::activeBandClass

8.225.2.2 uint16_t RFBandInfoElements::activeChannel

8.225.2.3 uint8_t RFBandInfoElements::radioInterface

8.226 rmTrasnferStaticsReq Struct Reference

Data Fields

- uint8_t [bResetStatistics](#)
- uint32_t [ulMask](#)

8.226.1 Detailed Description

Parameters

<i>bResetStatistics</i>	Clear RM statistics
<i>ulMask</i>	Requested statistic bit mask

8.226.2 Field Documentation

8.226.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

8.226.2.2 uint32_t rmTrasnferStaticsReq::ulMask

8.227 slot_t Struct Reference

Data Fields

- uint32_t [uPhyCardStatus](#)
- uint32_t [uPhySlotStatus](#)
- uint8_t [bLogicalSlot](#)
- uint8_t [bICCIDLength](#)
- uint8_t [bICCID](#) [255]

8.227.1 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 ICCCID
<i>bICCID[MAX_ICCID_LENGTH]</i>	<ul style="list-style-type: none"> • Contains the ICCID of the card in the physical slot.

8.227.2 Field Documentation

8.227.2.1 `uint8_t slot_t::bICCID[255]`8.227.2.2 `uint8_t slot_t::bICCIDLength`8.227.2.3 `uint8_t slot_t::bLogicalSlot`8.227.2.4 `uint32_t slot_t::uPhyCardStatus`8.227.2.5 `uint32_t slot_t::uPhySlotStatus`

8.228 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]

8.228.1 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> [MAX_ _NO_OF_APPL- ICATIONS]	<ul style="list-style-type: none">• See appStats for more information.
---	--

8.228.2 Field Documentation

8.228.2.1 `appStats slotInf::AppStatus[10]`

8.228.2.2 `uint8_t slotInf::cardState`

8.228.2.3 `uint8_t slotInf::errorState`

8.228.2.4 `uint8_t slotInf::numApp`

8.228.2.5 `uint8_t slotInf::upinRetries`

8.228.2.6 `uint8_t slotInf::upinState`

8.228.2.7 `uint8_t slotInf::upukRetries`

8.229 slots_t Struct Reference

Data Fields

- [slot_t uimSlotStatus](#) [255]

8.229.1 Field Documentation

8.229.1.1 `slot_t slots_t::uimSlotStatus[255]`

8.230 sMSCAddress Struct Reference

Data Fields

- `uint8_t` [length](#)
- `uint8_t` [data](#) [256]

8.230.1 Detailed Description

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

8.230.2 Field Documentation

8.230.2.1 `uint8_t sMSCAddress::data[256]`

8.230.2.2 `uint8_t sMSCAddress::length`

8.231 sMSCAddressTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSCAddressInfo SMSCInfo`

8.231.1 Detailed Description

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

8.231.2 Field Documentation

8.231.2.1 `sMSCAddressInfo sMSCAddressTlv::SMSCInfo`

8.231.2.2 `uint8_t sMSCAddressTlv::TlvPresent`

8.232 sMSEtwsMessage Struct Reference

Data Fields

- `uint8_t notificationType`
- `uint16_t length`
- `uint8_t data [1254]`

8.232.1 Detailed Description

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
-------------	--

8.232.2 Field Documentation

8.232.2.1 `uint8_t sMSEtwsMessage::data[1254]`

8.232.2.2 `uint16_t sMSEtwsMessage::length`

8.232.2.3 `uint8_t sMSEtwsMessage::notificationType`

8.233 sMSEtwsMessageTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSEtwsMessageInfo EtwsMessageInfo`

8.233.1 Detailed Description

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 <code>sMSEtwsMessageInfo</code> for more information

8.233.2 Field Documentation

8.233.2.1 `sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo`

8.233.2.2 `uint8_t sMSEtwsMessageTlv::TlvPresent`

8.234 sMSEtwsPlmn Struct Reference

Data Fields

- `uint16_t mobileCountryCode`
- `uint16_t mobileNetworkCode`

8.234.1 Detailed Description

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

8.234.2 Field Documentation

8.234.2.1 uint16_t sMSEtwsPlmn::mobileCountryCode

8.234.2.2 uint16_t sMSEtwsPlmn::mobileNetworkCode

8.235 sMSMessageMode Struct Reference

Data Fields

- uint8_t [messageMode](#)

8.235.1 Detailed Description

Parameters

<i>messageMode</i>	Message Mode
--------------------	--------------

8.235.2 Field Documentation

8.235.2.1 uint8_t sMSMessageMode::messageMode

8.236 sMSMTMessage Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t [messageIndex](#)

8.236.1 Detailed Description

Parameters

<i>storageType</i>	memory storage 0x00-UIM 0x01-NV
<i>messageIndex</i>	MT Message index

8.236.2 Field Documentation

8.236.2.1 uint32_t sMSMTMessage::messageIndex

8.236.2.2 uint32_t sMSMTMessage::storageType

8.237 sMSOnIMS Struct Reference

Data Fields

- `uint8_t` [sMSOnIMS](#)

8.237.1 Detailed Description

Parameters

<i>sMSOnIMS</i>	SMS on IMS
-----------------	------------

8.237.2 Field Documentation

8.237.2.1 `uint8_t` `sMSOnIMS::sMSOnIMS`

8.238 sMSOnIMSTlv Struct Reference

Data Fields

- `uint8_t` [TlvPresent](#)
- [sMSOnIMSInfo](#) `IMSInfo`

8.238.1 Detailed Description

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

8.238.2 Field Documentation

8.238.2.1 `sMSOnIMSInfo` `sMSOnIMSTlv::IMSInfo`

8.238.2.2 `uint8_t` `sMSOnIMSTlv::TlvPresent`

8.239 sMSTransferRouteMTMessage Struct Reference

Data Fields

- `uint8_t` [ackIndicator](#)
- `uint32_t` [transactionID](#)
- `uint8_t` [format](#)
- `uint16_t` [length](#)
- `uint8_t` [data](#) [256]

8.239.1 Detailed Description

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

8.239.2 Field Documentation

8.239.2.1 uint8_t sMSTransferRouteMTMessage::ackIndicator

8.239.2.2 uint8_t sMSTransferRouteMTMessage::data[256]

8.239.2.3 uint8_t sMSTransferRouteMTMessage::format

8.239.2.4 uint16_t sMSTransferRouteMTMessage::length

8.239.2.5 uint32_t sMSTransferRouteMTMessage::transactionID

8.240 tdscdmaSigInfoExt Struct Reference

Data Fields

- float [rssi](#)
- float [rscp](#)
- float [ecio](#)
- float [sinr](#)

8.240.1 Detailed Description

Parameters

<i>rssi</i>	RSSI in dBm.
<i>rsrq</i>	RSRQ value in dB
<i>rsrp</i>	Current RSRP in dBm as measured by L1.
<i>snr</i>	SNR level as a scaled integer in units of 0.1 dB.

8.240.2 Field Documentation

8.240.2.1 float tdscdmaSigInfoExt::ecio

8.240.2.2 float tdscdmaSigInfoExt::rscp

8.240.2.3 float tdscdmaSigInfoExt::rssi

8.240.2.4 float tdscdmaSigInfoExt::sinr

8.241 transferRouteMessageTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [sMSTransferRouteMTMessageInfo](#) [TransferRouteMTMessageInfo](#)

8.241.1 Detailed Description

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

8.241.2 Field Documentation

8.241.2.1 uint8_t transferRouteMessageTlv::TlvPresent

8.241.2.2 [sMSTransferRouteMTMessageInfo](#) transferRouteMessageTlv::TransferRouteMTMessageInfo

8.242 transferStatInd Struct Reference

Data Fields

- uint8_t [StatsPeriod](#)
- uint32_t [StatsMask](#)

8.242.1 Detailed Description

Parameters

<i>StatsPeriod</i>	Field Period between transfer statistic reports.
<i>StatsMask</i>	requested statistic bit mask.

8.242.2 Field Documentation

8.242.2.1 uint32_t transferStatInd::StatsMask

8.242.2.2 uint8_t transferStatInd::StatsPeriod

8.243 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](#)

8.243.1 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</i> [MAX_DESCRIPTION_LENGTH]	<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.

8.243.2 Field Documentation

8.243.2.1 `uint8_t uim_appStatus::aidLength`

8.243.2.2 `uint8_t uim_appStatus::aidVal[255]`

8.243.2.3 `uint8_t uim_appStatus::appState`

8.243.2.4 `uint8_t uim_appStatus::appType`

8.243.2.5 `uint8_t uim_appStatus::persoFeature`

8.243.2.6 `uint8_t uim_appStatus::persoRetries`

8.243.2.7 `uint8_t uim_appStatus::persoState`

8.243.2.8 `uint8_t uim_appStatus::persoUnblockRetries`

8.243.2.9 `uint8_t uim_appStatus::pin1Retries`

8.243.2.10 `uint8_t uim_appStatus::pin1State`

8.243.2.11 `uint8_t uim_appStatus::pin2Retries`

8.243.2.12 `uint8_t uim_appStatus::pin2State`

8.243.2.13 `uint8_t uim_appStatus::puk1Retries`

8.243.2.14 `uint8_t uim_appStatus::puk2Retries`

8.243.2.15 `uint8_t uim_appStatus::univPin`

8.244 uim_cardResult Struct Reference

Data Fields

- `uint8_t sw1`
- `uint8_t sw2`

8.244.1 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.

8.244.2 Field Documentation

8.244.2.1 uint8_t uim_cardResult::sw1

8.244.2.2 uint8_t uim_cardResult::sw2

8.245 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]

8.245.1 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.

8.245.2 Field Documentation

8.245.2.1 `uint16_t uim_cardStatus::index1xPri`

8.245.2.2 `uint16_t uim_cardStatus::index1xSec`

8.245.2.3 `uint16_t uim_cardStatus::indexGwPri`

8.245.2.4 `uint16_t uim_cardStatus::indexGwSec`

8.245.2.5 `uint8_t uim_cardStatus::numSlot`

8.245.2.6 `uim_slotInfo uim_cardStatus::SlotInfo[5]`

8.246 uim_changeUIMPIN Struct Reference

Data Fields

- `uint8_t pinID`
- `uint8_t oldPINLen`
- `uint8_t oldPINVal [255]`
- `uint8_t pinLen`
- `uint8_t pinVal [255]`

8.246.1 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_DESCRIPTOR_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_DESCRIPTOR_LENGTH]	<ul style="list-style-type: none"> New PIN value. This value is a sequence of ASCII characters.

8.246.2 Field Documentation

8.246.2.1 uint8_t uim_changeUIMPIN::oldPINLen

8.246.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

8.246.2.3 uint8_t uim_changeUIMPIN::pinID

8.246.2.4 uint8_t uim_changeUIMPIN::pinLen

8.246.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

8.247 uim_encryptedPIN1 Struct Reference

Data Fields

- uint8_t [pin1Len](#)
- uint8_t [pin1Val](#) [255]

8.247.1 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.

8.247.2 Field Documentation

8.247.2.1 `uint8_t uim_encryptedPIN1::pin1Len`

8.247.2.2 `uint8_t uim_encryptedPIN1::pin1Val[255]`

8.248 `uim_fileInfo` Struct Reference

Data Fields

- `uint16_t fileID`
- `uint8_t pathLen`
- `uint16_t path` [255]

8.248.1 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).

8.248.2 Field Documentation

8.248.2.1 `uint16_t uim_fileInfo::fileID`

8.248.2.2 `uint16_t uim_fileInfo::path[255]`

8.248.2.3 `uint8_t uim_fileInfo::pathLen`

8.249 uim_hotSwapStatus Struct Reference

Data Fields

- uint8_t [hotSwapLength](#)
- uint8_t [hotSwap](#) [255]

8.249.1 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

8.249.2 Field Documentation

8.249.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.249.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.250 uim_readResult Struct Reference

Data Fields

- uint16_t [contentLen](#)
- uint8_t [content](#) [255]

8.250.1 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[255]</i>	<ul style="list-style-type: none"> • Read content. • The content is the sequence of bytes as read from the card.
---------------------	--

8.250.2 Field Documentation

8.250.2.1 `uint8_t uim_readResult::content[255]`

8.250.2.2 `uint16_t uim_readResult::contentLen`

8.251 `uim_readTransparentInfo` Struct Reference

Data Fields

- `uint16_t` [offset](#)
- `uint16_t` [length](#)

8.251.1 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.

8.251.2 Field Documentation

8.251.2.1 `uint16_t uim_readTransparentInfo::length`

8.251.2.2 `uint16_t uim_readTransparentInfo::offset`

8.252 `uim_remainingRetries` Struct Reference

Data Fields

- `uint8_t` [verifyLeft](#)
- `uint8_t` [unblockLeft](#)

8.252.1 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.

8.252.2 Field Documentation

8.252.2.1 `uint8_t uim_remainingRetries::unblockLeft`

8.252.2.2 `uint8_t uim_remainingRetries::verifyLeft`

8.253 uim_sessionInformation Struct Reference

Data Fields

- `uint8_t sessionType`
- `uint8_t aidLength`
- `uint8_t aid [255]`

8.253.1 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.

8.253.2 Field Documentation

8.253.2.1 `uint8_t uim_sessionInformation::aid[255]`8.253.2.2 `uint8_t uim_sessionInformation::aidLength`8.253.2.3 `uint8_t uim_sessionInformation::sessionType`8.254 `uim_setPINProtection` Struct Reference

Data Fields

- `uint8_t pinID`
- `uint8_t pinOperation`
- `uint8_t pinLength`
- `uint8_t pinValue [255]`

8.254.1 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.

8.254.2 Field Documentation

8.254.2.1 uint8_t uim_setPINProtection::pinID

8.254.2.2 uint8_t uim_setPINProtection::pinLength

8.254.2.3 uint8_t uim_setPINProtection::pinOperation

8.254.2.4 uint8_t uim_setPINProtection::pinValue[255]

8.255 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]

8.255.1 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.
------------------	---

8.255.2 Field Documentation

8.255.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.255.2.2 `uint8_t uim_slotInfo::cardState`

8.255.2.3 `uint8_t uim_slotInfo::errorState`

8.255.2.4 `uint8_t uim_slotInfo::numApp`

8.255.2.5 `uint8_t uim_slotInfo::upinRetries`

8.255.2.6 `uint8_t uim_slotInfo::upinState`

8.255.2.7 `uint8_t uim_slotInfo::upukRetries`

8.256 uim_UIMSessionInformation Struct Reference

Data Fields

- `uint8_t sessionType`
- `uint8_t aidLength`
- `uint8_t aid [255]`

8.256.1 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.

8.256.2 Field Documentation

8.256.2.1 `uint8_t uim_UIMSessionInformation::aid[255]`

8.256.2.2 `uint8_t uim_UIMSessionInformation::aidLength`

8.256.2.3 `uint8_t uim_UIMSessionInformation::sessionType`

8.257 uim_unblockUIMPIN Struct Reference

Data Fields

- `uint8_t pinID`
- `uint8_t pukLen`
- `uint8_t pukVal [255]`
- `uint8_t newPINLen`
- `uint8_t newPINVal [255]`

8.257.1 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_DESCRIPTOR_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_DESCR- PTION_LEN- GTH]</i>	<ul style="list-style-type: none"> New PIN value. This value is a sequence of ASCII characters.

8.257.2 Field Documentation

8.257.2.1 `uint8_t uim_unblockUIMPIN::newPINLen`

8.257.2.2 `uint8_t uim_unblockUIMPIN::newPINVal[255]`

8.257.2.3 `uint8_t uim_unblockUIMPIN::pinID`

8.257.2.4 `uint8_t uim_unblockUIMPIN::pukLen`

8.257.2.5 `uint8_t uim_unblockUIMPIN::pukVal[255]`

8.258 uim_verifyUIMPIN Struct Reference

Data Fields

- `uint8_t pinID`
- `uint8_t pinLen`
- `uint8_t pinVal [255]`

8.258.1 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> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> PIN value. This value is a sequence of ASCII characters.

8.258.2 Field Documentation

8.258.2.1 uint8_t uim_verifyUIMPIN::pinID

8.258.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.258.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.259 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- uint8_t [state](#)

8.259.1 Detailed Description

Parameters

<i>pActivationState</i> [OUT]	<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
-------------------------------	--

8.259.2 Field Documentation

8.259.2.1 uint8_t unpack_dms_GetActivationState_t::state

8.260 unpack_dms_GetBandCapability_t Struct Reference

Data Fields

- uint32_t [BandCapability](#)
- uint16_t [Tlvresult](#)

8.260.1 Field Documentation

8.260.1.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

8.260.1.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.261 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

- uint8_t [DevCrashState](#)
- uint16_t [Tlvresult](#)

8.261.1 Field Documentation

8.261.1.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.261.1.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.262 unpack_dms_GetCustFeature_t Struct Reference

Data Fields

- uint32_t [GpsEnable](#)
- uint8_t [DisableIMSI](#)
- uint16_t [IPFamSupport](#)
- uint8_t [RMAutoConnect](#)
- uint8_t [GPSSel](#)
- uint8_t [SMSSupport](#)
- uint8_t [IsVoiceEnabled](#)
- uint8_t [DHCPRelayEnabled](#)
- uint8_t [GPSLPM](#)
- uint16_t [Tlvresult](#)

8.262.1 Field Documentation

8.262.1.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.262.1.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.262.1.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.262.1.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.262.1.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.262.1.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.262.1.7 `uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled`

8.262.1.8 `uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect`

8.262.1.9 `uint8_t unpack_dms_GetCustFeature_t::SMSSupport`

8.262.1.10 `uint16_t unpack_dms_GetCustFeature_t::Tlvresult`

8.263 `unpack_dms_GetCustFeaturesV2_t` Struct Reference

Data Fields

- [DMSgetCustomFeatureV2](#) `GetCustomFeatureV2`
- `uint16_t` [Tlvresult](#)

8.263.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
------------------	---

8.263.2 Field Documentation

8.263.2.1 `DMSgetCustomFeatureV2` `unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2`

8.263.2.2 `uint16_t` `unpack_dms_GetCustFeaturesV2_t::Tlvresult`

8.264 `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](#) [64]
- `uint16_t` [Tlvresult](#)

8.264.1 Field Documentation

8.264.1.1 `uint32_t` `unpack_dms_GetDeviceCap_t::DataServiceCapability`

8.264.1.2 `uint32_t` `unpack_dms_GetDeviceCap_t::MaxRXChannelRate`

8.264.1.3 `uint32_t` `unpack_dms_GetDeviceCap_t::MaxTXChannelRate`

8.264.1.4 `uint8_t` `unpack_dms_GetDeviceCap_t::Radiofaces`[64]

8.264.1.5 uint32_t unpack_dms_GetDeviceCap_t::RadiolfacesSize

8.264.1.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.264.1.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.265 unpack_dms_GetDeviceCapabilities_t Struct Reference

Data Fields

- uint32_t [maxTxChannelRate](#)
- uint32_t [maxRxChannelRate](#)
- uint32_t [dataServiceCaCapability](#)
- uint32_t [simCapability](#)
- uint32_t [radiolfacesSize](#)
- uint8_t [Radiolfaces](#) [255]

8.265.1 Detailed Description

Parameters

<i>maxTxChannelRate</i>	Maximum Tx transmission rate in bits per second.
<i>maxRxChannelRate</i>	Maximum Rx transmission rate in bits per second
<i>dataServiceCaCapability</i>	data service capability
<i>simCapability</i>	SIM Capability
<i>radiolfacesSize</i>	radio interface length
<i>Radiolfaces</i>	radio interfaces

8.265.2 Field Documentation

8.265.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

8.265.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

8.265.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

8.265.2.4 uint8_t unpack_dms_GetDeviceCapabilities_t::Radiolfaces[255]

8.265.2.5 uint32_t unpack_dms_GetDeviceCapabilities_t::radiolfacesSize

8.265.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

8.266 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

- uint8_t [stringSize](#)
- char [String](#) [255]
- uint16_t [Tlvresult](#)

8.266.1 Field Documentation

8.266.1.1 char unpack_dms_GetDeviceHardwareRev_t::String[255]

8.266.1.2 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

8.266.1.3 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

8.267 unpack_dms_GetDeviceMfr_t Struct Reference

Data Fields

- uint8_t [stringSize](#)
- char [String](#) [255]
- uint16_t [Tlvresult](#)

8.267.1 Field Documentation

8.267.1.1 char unpack_dms_GetDeviceMfr_t::String[255]

8.267.1.2 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

8.267.1.3 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

8.268 unpack_dms_GetDeviceSerialNumbers_t Struct Reference

Data Fields

- uint8_t [esnSize](#)
- char [ESNString](#) [255]
- uint8_t [imeiSize](#)
- char [IMEIString](#) [255]
- uint8_t [meidSize](#)
- char [MEIDString](#) [255]
- uint8_t [imeiSvnSize](#)
- char [ImeiSvnString](#) [255]
- uint16_t [Tlvresult](#)

8.268.1 Field Documentation

8.268.1.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

8.268.1.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

8.268.1.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

8.268.1.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

8.268.1.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

8.268.1.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

8.268.1.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

8.268.1.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

8.268.1.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

8.269 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](#) [10]
- char [cur_carr_name](#) [17]
- char [cur_carr_rev](#) [13]
- uint16_t [Tlvresult](#)

8.269.1 Detailed Description

Parameters

<i>modelid_str</i>	Mode ID String.
<i>bootversion_str</i>	Boot Version.
<i>appversion_str</i>	Application Version String.
<i>sku_str</i>	SKU String.
<i>packageid_str</i>	<ul style="list-style-type: none"> • Package ID String. • deprecated on EM/MC74xx(9x30) devices
<i>carrier_str</i>	Carrier String.
<i>priversion_str</i>	PRI Version String.
<i>priversion_str</i>	PRI Version String.
<i>cur_carr_name</i>	Current Carrier Name String.
<i>cur_carr_rev</i>	Current Carrier Revision String.
<i>Tlvresult</i>	Tlv Result.

8.269.2 Field Documentation

8.269.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

8.269.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

8.269.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

8.269.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

8.269.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

8.269.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

8.269.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

8.269.2.8 char unpack_dms_GetFirmwareInfo_t::priversion_str[10]

8.269.2.9 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

8.269.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.270 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

- uint8_t [amssSize](#)
- char [AMSSString](#) [255]
- uint16_t [Tlvresult](#)

8.270.1 Field Documentation

8.270.1.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.270.1.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.270.1.3 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

8.271 unpack_dms_GetFirmwareRevisions_t Struct Reference

Data Fields

- uint8_t [amssSize](#)
- char [AMSSString](#) [255]
- uint8_t [bootSize](#)
- char [BootString](#) [255]
- uint8_t [priSize](#)
- char [PRIString](#) [255]
- uint16_t [Tlvresult](#)

8.271.1 Detailed Description

Parameters

<i>amssstring</i>	AMSS revision string
<i>bootstring</i>	boot code revision string
<i>pristring</i>	PRI revision string

8.271.2 Field Documentation

8.271.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

8.271.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

8.271.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

8.271.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

8.271.2.5 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

8.271.2.6 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

8.271.2.7 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

8.272 unpack_dms_GetFSN_t Struct Reference

Data Fields

- char [String](#) [255]
- uint16_t [Tlvresult](#)

8.272.1 Field Documentation

8.272.1.1 char unpack_dms_GetFSN_t::String[255]

8.272.1.2 uint16_t unpack_dms_GetFSN_t::Tlvresult

8.273 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char [hwVer](#) [255]

8.273.1 Detailed Description

Parameters

<i>hwVer</i>	hardware vesion
--------------	-----------------

8.273.2 Field Documentation

8.273.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.274 unpack_dms_GetIMSI_t Struct Reference

Data Fields

- char [imsi](#) [255]
- uint16_t [Tlvresult](#)

8.274.1 Field Documentation

8.274.1.1 char unpack_dms_GetIMSI_t::imsi[255]

8.274.1.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.275 unpack_dms_GetModelID_t Struct Reference

Data Fields

- char [modelid](#) [255]
- uint16_t [Tlvresult](#)

8.275.1 Detailed Description

Parameters

<i>modelid</i>	device model id
----------------	-----------------

8.275.2 Field Documentation

8.275.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.275.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.276 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

- uint16_t [source](#)
- uint64_t [timestamp](#)
- uint16_t [Tlvresult](#)

8.276.1 Detailed Description

Parameters

<i>source</i>	<ul style="list-style-type: none"> • Source of timestamp <ul style="list-style-type: none"> 0 - 32 kHz device clock 1 - CDMA network 2 - cdma2000 1xEV-DO network
<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)

8.276.2 Field Documentation

8.276.2.1 uint16_t unpack_dms_GetNetworkTime_t::source

8.276.2.2 uint64_t unpack_dms_GetNetworkTime_t::timestamp

8.276.2.3 uint16_t unpack_dms_GetNetworkTime_t::Tlvresult

8.277 unpack_dms_GetPower_t Struct Reference

Data Fields

- uint32_t [OperationMode](#)
- uint32_t [OfflineReason](#)
- uint32_t [HardwareControlledMode](#)
- uint16_t [Tlvresult](#)

8.277.1 Detailed Description

Parameters

<i>OperationMode</i>	operating mode
<i>OfflineReason</i>	offline reason
<i>Hardware-ControlledMode</i>	hardware restricted mode

8.277.2 Field Documentation

8.277.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

8.277.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

8.277.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

8.277.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

8.278 unpack_dms_GetPRLVersion_t Struct Reference

Data Fields

- uint8_t [u8PRLPreference](#)
- uint16_t [u16PRLVersion](#)
- uint16_t [Tlvresult](#)

8.278.1 Field Documentation

8.278.1.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.278.1.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.278.1.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.279 unpack_dms_GetSerialNumbers_t Struct Reference

Data Fields

- char [esn](#) [255]
- char [imei_no](#) [255]
- char [meid](#) [255]
- char [imeisv_svn](#) [255]

8.279.1 Detailed Description

Parameters

<i>esn</i>	Electronic Serial Number of the device
<i>imei_no</i>	International Mobile Equipment Identity of the device.
<i>meid</i>	Mobile Equipment Identifier of the device.
<i>imeisv_svn</i>	imei software version revision

8.279.2 Field Documentation

8.279.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

8.279.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

8.279.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

8.279.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

8.280 unpack_dms_GetUSBComp_t Struct Reference

Data Fields

- uint8_t [USBComp](#) [255]
- uint8_t [NumSupUSBComps](#)
- uint8_t [SupUSBComps](#)
- uint16_t [Tlvresult](#)

8.280.1 Field Documentation

8.280.1.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.280.1.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

8.280.1.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.280.1.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

8.281 unpack_dms_GetVoiceNumber_t Struct Reference

Data Fields

- uint8_t [voiceNumberSize](#)
- char [VoiceNumber](#) [255]
- uint8_t [minSize](#)
- char [MIN](#) [255]
- uint16_t [Tlvresult](#)

8.281.1 Field Documentation

8.281.1.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.281.1.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.281.1.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.281.1.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.281.1.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.282 unpack_dms_SetCustFeature_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.282.1 Field Documentation

8.282.1.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

8.283 unpack_dms_SetCustFeaturesV2_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.283.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result
------------------	---

8.283.2 Field Documentation

8.283.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

8.284 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

- [dms_ActivationStatusTlv](#) ActivationStatusTlv
- [dms_OperatingModeTlv](#) OperatingModeTlv
- uint16_t [Tlvresult](#)

8.284.1 Detailed Description

DMS Event Report indication structure

Parameters

<i>ActivationStatus-Tlv</i>	<ul style="list-style-type: none"> • See dms_ActivationStatusTlv
<i>OperatingMode-Tlv</i>	<ul style="list-style-type: none"> • See dms_OperatingModeTlv
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result

8.284.2 Field Documentation

8.284.2.1 `dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv`

8.284.2.2 `dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv`

8.284.2.3 `uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult`

8.285 unpack_dms_SetEventReport_t Struct Reference**Data Fields**

- `uint16_t` [Tlvresult](#)

8.285.1 Field Documentation

8.285.1.1 `uint16_t unpack_dms_SetEventReport_t::Tlvresult`

8.286 unpack_dms_SetFirmwarePreference_t Struct Reference**Data Fields**

- `uint16_t` [Tlvresult](#)

8.286.1 Field Documentation

8.286.1.1 `uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult`

8.287 unpack_dms_SetPower_t Struct Reference**Data Fields**

- `uint16_t` [Tlvresult](#)

8.287.1 Field Documentation

8.287.1.1 `uint16_t unpack_dms_SetPower_t::Tlvresult`

8.288 unpack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.288.1 Field Documentation

8.288.1.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

8.289 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [source](#)
- uint16_t [Tlvresult](#)

8.289.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

<i>OUT]</i>	type[OUT] <ul style="list-style-type: none"> • 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
<i>OUT]</i>	source[OUT] <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
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result

8.289.2 Field Documentation

8.289.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

8.289.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

8.289.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

8.290 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

Data Fields

- uint8_t [type](#)
- uint8_t [source](#)
- uint16_t [Tlvresult](#)

8.290.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

Parameters

<i>OUT]</i>	type[OUT] <ul style="list-style-type: none"> • 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
<i>OUT]</i>	source[OUT] <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

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result
------------------	---

8.290.2 Field Documentation

8.290.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.290.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.290.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.291 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.291.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result
------------------	---

8.291.2 Field Documentation

8.291.2.1 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::Tlvresult

8.292 unpack_dms_SLQSGetBandCapability_t Struct Reference

Data Fields

- uint64_t [bandCapability](#)
- int [is_LteBandCapability_Available](#)
- uint64_t [LteBandCapability](#)
- int [is_TdsBandCapability_Available](#)
- uint64_t [TdsBandCapability](#)

8.292.1 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
	<ul style="list-style-type: none"> • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved

<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
---------------------------------	--

8.292.2 Field Documentation

8.292.2.1 `uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability`

8.292.2.2 `int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available`

8.292.2.3 `int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available`

8.292.2.4 `uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability`

8.292.2.5 `uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability`

8.293 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.293.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
------------------	---

8.293.2 Field Documentation

8.293.2.1 `uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult`

8.294 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

- `packgetDyingGaspCfg * pGetDyingGaspCfg`
- `uint16_t` [Tlvresult](#)

8.294.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result
------------------	---

8.294.2 Field Documentation

8.294.2.1 `packgetDyingGaspCfg*` `unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg`

8.294.2.2 `uint16_t` `unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult`

8.295 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

Data Fields

- `packgetDyingGaspStatistics*` `pGetDyingGaspStatistics`
- `uint16_t` `Tlvresult`

8.295.1 Detailed Description

This structure contains Get Dying GASP Statistics.

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result
------------------	---

8.295.2 Field Documentation

8.295.2.1 `packgetDyingGaspStatistics*` `unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics`

8.295.2.2 `uint16_t` `unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

8.296 unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference

Data Fields

- `uint8_t` `numEntries`
- `image_info_t*` `pCurrImgInfo`
- `char` `priver` [16]
- `char` `pkgver` [16]
- `char` `fwvers` [16]
- `char` `carrier` [16]

8.296.1 Detailed Description

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
<i>pCurrImgInfo</i> [OUT]	<ul style="list-style-type: none"> • Currently Active Image List
<i>priver</i> [OUT]	<ul style="list-style-type: none"> • PRI version of the currently running firmware
<i>pkgver</i> [OUT]	<ul style="list-style-type: none"> • Package version of the currently running firmware
<i>fwvers</i> [OUT]	<ul style="list-style-type: none"> • firmware version of the currently running firmware
<i>carrier</i> [OUT]	<ul style="list-style-type: none"> • Carrier string of the currently running firmware

8.296.2 Field Documentation

8.296.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.296.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.296.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.296.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.296.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.296.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.297 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.297.1 Detailed Description

This structure contains set Dying GASP Config unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack Result
------------------	---

8.297.2 Field Documentation

8.297.2.1 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

8.298 unpack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint8_t [stringSize](#)
- uint8_t [String](#) [255]
- uint16_t [Tlvresult](#)

8.298.1 Detailed Description

This structure contains Get ICCID pack

Parameters

<i>stringSize</i>	<ul style="list-style-type: none">• Size of String.
<i>String</i>	<ul style="list-style-type: none">• ICCID String.
<i>Tlvresult</i>	<ul style="list-style-type: none">• Pack result.

8.298.2 Field Documentation

8.298.2.1 uint8_t unpack_dms_UIMGetICCID_t::String[255]

8.298.2.2 uint8_t unpack_dms_UIMGetICCID_t::stringSize

8.298.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.299 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint32_t [ImageListSize](#)
- [FMSPrefImageList](#) * [pImageList](#)
- uint16_t [Tlvresult](#)

8.299.1 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
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack result

8.299.2 Field Documentation

8.299.2.1 `uint32_t unpack_fms_GetImagesPreference_t::ImageListSize`

8.299.2.2 `FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList`

8.299.2.3 `uint16_t unpack_fms_GetImagesPreference_t::Tlvresult`

8.300 `unpack_fms_GetStoredImages_t` Struct Reference

Data Fields

- `uint32_t` [imagelistSize](#)
- [FMSImageList](#) [imageList](#)
- `uint16_t` [Tlvresult](#)

8.300.1 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
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack result

8.300.2 Field Documentation

8.300.2.1 `FMSImageList unpack_fms_GetStoredImages_t::imageList`

8.300.2.2 `uint32_t unpack_fms_GetStoredImages_t::imagelistSize`

8.300.2.3 uint16_t unpack_fms_GetStoredImages_t::Tlvresult

8.301 unpack_fms_SetImagesPreference_t Struct Reference

Data Fields

- uint32_t [ImageTypesSize](#)
- uint8_t [ImageTypes](#) [255]
- uint16_t [Tlvresult](#)

8.301.1 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
<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result

8.301.2 Field Documentation

8.301.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

8.301.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

8.301.2.3 uint16_t unpack_fms_SetImagesPreference_t::Tlvresult

8.302 unpack_loc_Delete_Assist_Data_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.302.1 Detailed Description

This structure contains LOC delete assist data unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.
------------------	--

8.302.2 Field Documentation

8.302.2.1 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

8.303 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

- uint32_t [engineState](#)
- uint16_t [Tlvresult](#)

8.303.1 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
<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack result

8.303.2 Field Documentation

8.303.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.303.2.2 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

8.304 unpack_loc_EventRegister_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.304.1 Detailed Description

This structure contains Event Register unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.
------------------	--

8.304.2 Field Documentation

8.304.2.1 uint16_t unpack_loc_EventRegister_t::Tlvresult

8.305 unpack_loc_PositionRpt_Ind_t Struct Reference

Data Fields

- uint32_t [sessionStatus](#)
- uint8_t [sessionId](#)
- uint64_t * [pLatitude](#)
- uint64_t * [pLongitude](#)
- uint32_t * [pHorUncCircular](#)
- uint32_t * [pHorUncEllipseSemiMinor](#)
- uint32_t * [pHorUncEllipseSemiMajor](#)
- uint32_t * [pHorUncEllipseOrientAzimuth](#)
- uint8_t * [pHorConfidence](#)
- uint32_t * [pHorReliability](#)
- uint32_t * [pSpeedHorizontal](#)
- uint32_t * [pSpeedUnc](#)
- uint32_t * [pAltitudeWrtEllipsoid](#)
- uint32_t * [pAltitudeWrtMeanSeaLevel](#)
- uint32_t * [pVertUnc](#)
- uint8_t * [pVertConfidence](#)
- uint32_t * [pVertReliability](#)
- uint32_t * [pSpeedVertical](#)
- uint32_t * [pHeading](#)
- uint32_t * [pHeadingUnc](#)
- uint32_t * [pMagneticDeviation](#)
- uint32_t * [pTechnologyMask](#)
- [loc_precisionDilution](#) * [pPrecisionDilution](#)
- uint64_t * [pTimestampUtc](#)
- uint8_t * [pLeapSeconds](#)
- [loc_gpsTime](#) * [pGpsTime](#)
- uint32_t * [pTimeUnc](#)
- uint32_t * [pTimeSrc](#)
- [loc_sensorDataUsage](#) * [pSensorDataUsage](#)
- uint32_t * [pFixId](#)
- [loc_svUsedforFix](#) * [pSvUsedforFix](#)
- uint8_t * [pAltitudeAssumed](#)
- uint16_t [Tlvresult](#)

8.305.1 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
<i>sessionId</i>	<ul style="list-style-type: none"> • ID of the session that was specified in the Start request • Range - 0 to 255
<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
<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

<i>pHorUncCircular</i>	<ul style="list-style-type: none"> • Horizontal position uncertainty. • Units - Meters
<i>pHorUncEllipse-SemiMinor</i>	<ul style="list-style-type: none"> • Semi-minor axis of horizontal elliptical uncertainty. • Units - Meters
<i>pHorUncEllipse-SemiMajor</i>	<ul style="list-style-type: none"> • Semi-major axis of horizontal elliptical uncertainty. • Units: Meters
<i>pHorUncEllipse-OrientAzimuth</i>	<ul style="list-style-type: none"> • Elliptical horizontal uncertainty azimuth of orientation. • Units - Decimal degrees • Range - 0 to 180
<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
<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
<i>pSpeed-Horizontal</i>	<ul style="list-style-type: none"> • Horizontal speed. • Units - Meters/second
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> • 3-D Speed uncertainty. • Units - Meters/second.

<i>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> • Altitude With Respect to WGS84 Ellipsoid. • Units - Meters • Range -500 to 15883
<i>pAltitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> • Altitude With Respect to Sea Level. • Units - Meters
<i>pVertUnc</i>	<ul style="list-style-type: none"> • Vertical uncertainty. • Units - Meters
<i>pVertConfidence</i>	<ul style="list-style-type: none"> • Vertical uncertainty confidence. • Units - Percentage • Range 0 to 99
<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
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> • Vertical speed. • Units - Meters/second
<i>pHeading</i>	<ul style="list-style-type: none"> • Heading. • Units - Degree • Range 0 to 359.999
<i>pHeadingUnc</i>	<ul style="list-style-type: none"> • Heading uncertainty. • Units - Degree • Range 0 to 359.999

<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.
<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
<i>-pPrecision-Dilution</i>	<ul style="list-style-type: none"> • See loc_precisionDilution for more information
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> • UTC timestamp • Units - Milliseconds since Jan. 1, 1970
<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
<i>-pGpsTime</i>	<ul style="list-style-type: none"> • See loc_gpsTime for more information
<i>pTimeUnc</i>	<ul style="list-style-type: none"> • Time uncertainty. • Units - Milliseconds
<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
Generated on Tue May 31 2016 14:24:35 for Linux QMISDK by Doxygen	
	<ul style="list-style-type: none"> – 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

<i>-pSensorData-Usage</i>	<ul style="list-style-type: none"> • See loc_sensorDataUsage for more information
<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.
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> • See loc_svUsedforFix for more information
<i>pAltitude-Assumed</i>	<ul style="list-style-type: none"> • Indicates whether altitude is assumed or calculated.

- Value
 - 0x00 - Altitude is calculated
 - 0x01 - Altitude is assumed

8.305.2 Field Documentation

- 8.305.2.1 `uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed`
- 8.305.2.2 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid`
- 8.305.2.3 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel`
- 8.305.2.4 `uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId`
- 8.305.2.5 `loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime`
- 8.305.2.6 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading`
- 8.305.2.7 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc`
- 8.305.2.8 `uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence`
- 8.305.2.9 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability`
- 8.305.2.10 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular`
- 8.305.2.11 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.305.2.12 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor`
- 8.305.2.13 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor`
- 8.305.2.14 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude`
- 8.305.2.15 `uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds`
- 8.305.2.16 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude`
- 8.305.2.17 `uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation`

- 8.305.2.18 `loc_precisionDilution*` `unpack_loc_PositionRpt_Ind_t::pPrecisionDilution`
- 8.305.2.19 `loc_sensorDataUsage*` `unpack_loc_PositionRpt_Ind_t::pSensorDataUsage`
- 8.305.2.20 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal`
- 8.305.2.21 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pSpeedUnc`
- 8.305.2.22 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pSpeedVertical`
- 8.305.2.23 `loc_svUsedforFix*` `unpack_loc_PositionRpt_Ind_t::pSvUsedforFix`
- 8.305.2.24 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pTechnologyMask`
- 8.305.2.25 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pTimeSrc`
- 8.305.2.26 `uint64_t*` `unpack_loc_PositionRpt_Ind_t::pTimestampUtc`
- 8.305.2.27 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pTimeUnc`
- 8.305.2.28 `uint8_t*` `unpack_loc_PositionRpt_Ind_t::pVertConfidence`
- 8.305.2.29 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pVertReliability`
- 8.305.2.30 `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pVertUnc`
- 8.305.2.31 `uint8_t` `unpack_loc_PositionRpt_Ind_t::sessionId`
- 8.305.2.32 `uint32_t` `unpack_loc_PositionRpt_Ind_t::sessionStatus`
- 8.305.2.33 `uint16_t` `unpack_loc_PositionRpt_Ind_t::Tlvresult`

8.306 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.306.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result.
------------------	--

8.306.2 Field Documentation

- 8.306.2.1 `uint16_t` `unpack_loc_SetExtPowerState_t::Tlvresult`

8.307 unpack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.307.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.
------------------	--

8.307.2 Field Documentation

8.307.2.1 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

8.308 unpack_loc_Start_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.308.1 Detailed Description

This structure contains Start LOC unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result.
------------------	--

8.308.2 Field Documentation

8.308.2.1 uint16_t unpack_loc_Start_t::Tlvresult

8.309 unpack_loc_Stop_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.309.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack result.
------------------	--

8.309.2 Field Documentation

8.309.2.1 uint16_t unpack_loc_Stop_t::Tlvresult

8.310 unpack_nas_GetCDMANetworkParameters_t Struct Reference

Data Fields

- uint8_t [SCI](#)
- uint8_t [SCM](#)
- uint8_t [RegHomeSID](#)
- uint8_t [RegForeignSID](#)
- uint8_t [RegForeignNID](#)
- uint8_t [ForceRev0](#)
- uint8_t [CustomSCP](#)
- uint32_t [Protocol](#)
- uint32_t [Broadcast](#)
- uint32_t [Application](#)
- uint32_t [Roaming](#)

8.310.1 Detailed Description

Parameters

<i>SCI</i>	slot cycle index
<i>SCM</i>	station class mark
<i>RegHomeSID</i>	register on home sid
<i>RegForeignSID</i>	register on foreign sid
<i>RegForeignNID</i>	register on foreign nid
<i>ForceRev0</i>	force header revision
<i>CustomSCP</i>	custom SCP
<i>Protocol</i>	protocol
<i>Booadcast</i>	broadcast
<i>Application</i>	application
<i>Roaming</i>	roaming

8.310.2 Field Documentation

8.310.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.310.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.310.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.310.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.310.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.310.2.6 `uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID`

8.310.2.7 `uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID`

8.310.2.8 `uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID`

8.310.2.9 `uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming`

8.310.2.10 `uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI`

8.310.2.11 `uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM`

8.311 `unpack_nas_GetHomeNetwork_t` Struct Reference

Data Fields

- `uint16_t` [mcc](#)
- `uint16_t` [mnc](#)
- `char` [name](#) [255]
- `uint16_t` [sid](#)
- `uint16_t` [nid](#)

8.311.1 Detailed Description

Parameters

<i>mcc</i>	mobile country code
<i>mnc</i>	mobile network code
<i>name</i>	network name or description
<i>sid</i>	home network system id only applies to cdma2000
<i>nid</i>	home network id Only applies to cdma2000

8.311.2 Field Documentation

8.311.2.1 `uint16_t unpack_nas_GetHomeNetwork_t::mcc`

8.311.2.2 `uint16_t unpack_nas_GetHomeNetwork_t::mnc`

8.311.2.3 `char unpack_nas_GetHomeNetwork_t::name[255]`

8.311.2.4 `uint16_t unpack_nas_GetHomeNetwork_t::nid`

8.311.2.5 `uint16_t unpack_nas_GetHomeNetwork_t::sid`

8.312 `unpack_nas_GetNetworkPreference_t` Struct Reference

Data Fields

- `uint32_t` [ActiveTechPref](#)
- `uint32_t` [Duration](#)
- `uint32_t` [PersistentTechPref](#)
- `uint16_t` [Tlvresult](#)

8.312.1 Detailed Description

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
<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
<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
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.312.2 Field Documentation

8.312.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.312.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.312.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.312.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.313 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

- uint8_t [instancesSize](#)
- [RFBandInfoElements](#) [RFBandInfoElements](#) [255]

8.313.1 Detailed Description

Parameters

<i>instancesSize</i>	number of elements in RF info instances array.
<i>RFBandInfo-Elements</i>	RF info instances array

8.313.2 Field Documentation

8.313.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

8.313.2.2 [RFBandInfoElements](#) unpack_nas_GetRFInfo_t::RFBandInfoElements[255]

8.314 unpack_nas_GetServingNetwork_t Struct Reference

Data Fields

- uint32_t [RegistrationState](#)
- uint32_t [CSDomain](#)
- uint32_t [PSDomain](#)
- uint32_t [RAN](#)
- uint8_t [RadiolfacesSize](#)
- uint8_t [Radiolfaces](#) [255]
- uint32_t [Roaming](#)
- uint16_t [MCC](#)
- uint16_t [MNC](#)
- uint8_t [nameSize](#)
- uint8_t [Name](#) [255]
- uint8_t [DataCapsLen](#)
- uint8_t [DataCaps](#) [255]

8.314.1 Detailed Description

Parameters

<i>Registration-State</i>	registration state
<i>CSDomain</i>	CS domain
<i>PSDomain</i>	PS domain
<i>RAN</i>	radio access network
<i>RadiolfacesSize</i>	radio interface size
<i>Radiolfaces</i>	radio interface list
<i>Roaming</i>	romaing indicator

<i>MCC</i>	Mobile country code
<i>MNC</i>	Mobile network code
<i>nameSize</i>	network name size
<i>Name</i>	network name
<i>DataCapsLen</i>	data capabilities len
<i>DataCap</i>	data capabilities

8.314.2 Field Documentation

8.314.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.314.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.314.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.314.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.314.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

8.314.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

8.314.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

8.314.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

8.314.2.9 uint8_t unpack_nas_GetServingNetwork_t::Radiolfaces[255]

8.314.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadiolfacesSize

8.314.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

8.314.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

8.314.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

8.315 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

Data Fields

- uint8_t [DataCapsLen](#)
- uint8_t [DataCaps](#) [255]

8.315.1 Detailed Description

Parameters

<i>DataCapsLen</i>	data capabilities len
<i>DataCap</i>	data capabilities

8.315.2 Field Documentation

8.315.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

8.315.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

8.316 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

- uint32_t [len](#)
- signed char [rssi](#) [8]
- uint32_t [radio](#) [8]

8.316.1 Detailed Description

Parameters

<i>len</i>	number of rssi & radio items following
<i>rssi</i>	signal strength array
<i>radio</i>	radio interface array

8.316.2 Field Documentation

8.316.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

8.316.2.2 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

8.316.2.3 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

8.317 unpack_nas_PerformNetworkScan_t Struct Reference

Data Fields

- uint8_t * [p3GppNetworkInstanceSize](#)
- [nas_QmiNas3GppNetworkInfo](#) * [p3GppNetworkInfoInstances](#)
- uint8_t * [pRATInstanceSize](#)
- [nas_QmiNas3GppNetworkRAT](#) * [pRATInstance](#)
- uint8_t * [pPCSInstanceSize](#)
- [nas_QmisNasPcsDigit](#) * [pPCSInstance](#)
- uint32_t * [pScanResult](#)

8.317.1 Detailed Description

Parameters

<i>InstanceSize</i>	total instances
<i>Instances</i>	info for instances

8.317.2 Field Documentation

8.317.2.1 [nas_QmiNas3GppNetworkInfo](#)* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

8.317.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

8.317.2.3 `nas_QmisNasPcsDigit*` `unpack_nas_PerformNetworkScan_t::pPCSInstance`

8.317.2.4 `uint8_t*` `unpack_nas_PerformNetworkScan_t::pPCSInstanceSize`

8.317.2.5 `nas_QmiNas3GppNetworkRAT*` `unpack_nas_PerformNetworkScan_t::pRATInstance`

8.317.2.6 `uint8_t*` `unpack_nas_PerformNetworkScan_t::pRATInstanceSize`

8.317.2.7 `uint32_t*` `unpack_nas_PerformNetworkScan_t::pScanResult`

8.318 `unpack_nas_SetDataCapabilitiesCallback_ind_t` Struct Reference

Data Fields

- `uint8_t` [dataCapsSize](#)
- `uint8_t` [dataCaps](#) [255]

8.318.1 Detailed Description

Parameters

<i>dataCapsSize</i>	Number of Data Capabilities
<i>dataCaps</i>	Data Capabilities

8.318.2 Field Documentation

8.318.2.1 `uint8_t` `unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]`

8.318.2.2 `uint8_t` `unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize`

8.319 `unpack_nas_SetEventReportInd_t` Struct Reference

Data Fields

- [nas_SignalStrengthTlv](#) SSTlv
- [nas_RFInfoTlv](#) RFTlv
- [nas_RejectReasonTlv](#) RRTlv
- [nas_SLQSSignalStrengthsTlv](#) SLQSSSTlv

8.319.1 Detailed Description

Parameters

<i>SSTlv</i>	signal strength tlv
<i>RFTlv</i>	RF tlv
<i>RRTlv</i>	RR tlv
<i>SLQSSSTlv</i>	signal strength complete info tlv

8.319.2 Field Documentation

8.319.2.1 `nas_RFInfoTlv` `unpack_nas_SetEventReportInd_t::RFTlv`

8.319.2.2 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

8.319.2.3 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

8.319.2.4 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

8.320 unpack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.320.1 Detailed Description

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
----------------------------	--

<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
<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
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.320.2 Field Documentation

8.320.2.1 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

8.321 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- uint8_t [roaming](#)

8.321.1 Detailed Description

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
----------------	---

8.321.2 Field Documentation

8.321.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.322 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

- [NASServingSystemInfo SSInfo](#)
- uint16_t [Tlvresult](#)

8.322.1 Detailed Description

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
<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack result

8.322.2 Field Documentation

8.322.2.1 [NASServingSystemInfo](#) unpack_nas_SetServingSystemCallback_ind_t::SSInfo

8.322.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

8.323 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- [NasGetLTECphyCAInfo LTECphyCAInfo](#)
- uint16_t [Tlvresult](#)

8.323.1 Detailed Description

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

8.323.2 Field Documentation

8.323.2.1 [NasGetLTECphyCAInfo](#) unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCAInfo

8.323.2.2 uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult

8.324 unpack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

- uint8_t [spnEncoding](#)
- uint8_t [spnLength](#)
- char [spn](#) [255]
- uint8_t [shortNameEn](#)
- uint8_t [shortNameCI](#)
- uint8_t [shortNameSB](#)
- char [shortNameLen](#)
- uint8_t [shortName](#) [255]
- uint8_t [longNameEn](#)
- uint8_t [longNameCI](#)
- uint8_t [longNameSB](#)
- uint8_t [longNameLen](#)
- char [longName](#) [255]

8.324.1 Field Documentation

- 8.324.1.1 char [unpack_nas_SLQSGetPLMNName_t::longName](#)[255]
- 8.324.1.2 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameCI](#)
- 8.324.1.3 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameEn](#)
- 8.324.1.4 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameLen](#)
- 8.324.1.5 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameSB](#)
- 8.324.1.6 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortName](#)[255]
- 8.324.1.7 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameCI](#)
- 8.324.1.8 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameEn](#)
- 8.324.1.9 char [unpack_nas_SLQSGetPLMNName_t::shortNameLen](#)
- 8.324.1.10 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameSB](#)
- 8.324.1.11 char [unpack_nas_SLQSGetPLMNName_t::spn](#)[255]
- 8.324.1.12 uint8_t [unpack_nas_SLQSGetPLMNName_t::spnEncoding](#)
- 8.324.1.13 uint8_t [unpack_nas_SLQSGetPLMNName_t::spnLength](#)

8.325 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](#)

8.325.1 Detailed Description

Parameters

<i>ServingSystem</i>	serving system info
<i>RoamIndicator-Val</i>	roaming indicator value
<i>DataSrv-Capabilities</i>	data servcie capabilities
<i>CurrentPLMN</i>	current PLMN info
<i>SystemID</i>	system id
<i>NetworkID</i>	network id
<i>BasestationID</i>	base station id
<i>Basestation-Latitude</i>	base station latitude
<i>Basestation-Longitude</i>	base station longitude
<i>Roaming-IndicatorList</i>	roaming indicator list
<i>DefaultRoamInd</i>	default roaming indicator
<i>3Gpp2TimeZone</i>	3Gpp2 time zone
<i>pCDMA_P_Rev</i>	cdma P_Rev in use
<i>3GppTimeZone</i>	3Gpp time zone
<i>GppNetworkDS-TAdjustment</i>	3GPP network daylight saving adjustment
<i>Lac</i>	location area code
<i>CellID</i>	3GPP cell id
<i>ConcSvcInfo</i>	3GPP2 concurrent servcie info
<i>PRLInd</i>	3GPP2 PRL indicator

<i>DTMInd</i>	DTM indicator(GSM)
<i>DetailedSvcInfo</i>	detail servcie info
<i>CDMASystem-InfoExt</i>	extra cdma system info
<i>HdrPersonality</i>	hdr personality
<i>TrackAreaCode</i>	track area code
<i>CallBarStatus</i>	call barring status

8.325.2 Field Documentation

- 8.325.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID
- 8.325.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude
- 8.325.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude
- 8.325.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus
- 8.325.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev
- 8.325.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt
- 8.325.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID
- 8.325.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo
- 8.325.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN
- 8.325.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities
- 8.325.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd
- 8.325.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo
- 8.325.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd
- 8.325.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone
- 8.325.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment
- 8.325.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone
- 8.325.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality
- 8.325.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac
- 8.325.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID
- 8.325.2.20 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd
- 8.325.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal
- 8.325.2.22 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList
- 8.325.2.23 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem

8.325.2.24 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID

8.325.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

8.326 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 [lte snr](#)
- int16_t [lte srp](#)

8.326.1 Detailed Description

Parameters

<i>rxSignalStrengthListLen</i>	number of elements in Receive Signal Strength List
<i>rxSignalStrengthList</i>	signal strength list
<i>ecioListLen</i>	number of elements in ECIO List
<i>ecioList</i>	ecio list
<i>lo</i>	received lo in dBm; IO is only applicable for 1xEV-DO
<i>sinr</i>	SINR level; SINR is only applicable for 1xEV-DO
<i>errorRateListLen</i>	number of elements in Error Rate List
<i>errorRateList</i>	error rate list
<i>rsrqInfo</i>	rsrq info
<i>lte snr</i>	lte snr info
<i>lte srp</i>	lte srp info

8.326.2 Field Documentation

8.326.2.1 [nas_ecioListElement](#) [unpack_nas_SLQSGetSignalStrength_t::ecioList](#)[18]

8.326.2.2 uint16_t [unpack_nas_SLQSGetSignalStrength_t::ecioListLen](#)

8.326.2.3 [nas_errorRateListElement](#) [unpack_nas_SLQSGetSignalStrength_t::errorRateList](#)[18]

8.326.2.4 uint16_t [unpack_nas_SLQSGetSignalStrength_t::errorRateListLen](#)

8.326.2.5 int32_t [unpack_nas_SLQSGetSignalStrength_t::lo](#)

8.326.2.6 int16_t [unpack_nas_SLQSGetSignalStrength_t::lte srp](#)

8.326.2.7 `int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr`

8.326.2.8 `nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo`

8.326.2.9 `nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]`

8.326.2.10 `uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen`

8.326.2.11 `uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask`

8.326.2.12 `uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr`

8.327 `unpack_nas_SLQSGetSysInfo_t` Struct Reference

Data Fields

- `nas_SrvStatusInfo` * `pCDMASrvStatusInfo`
- `nas_SrvStatusInfo` * `pHDRSrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pGSMSrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pWCDMASrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pLTESrvStatusInfo`
- `nas_CDMASysInfo` * `pCDMASysInfo`
- `nas_HDRSysInfo` * `pHDRSysInfo`
- `nas_GSMSysInfo` * `pGSMSysInfo`
- `nas_WCDMASysInfo` * `pWCDMASysInfo`
- `nas_LTESysInfo` * `pLTESysInfo`
- `nas_AddCDMASysInfo` * `pAddCDMASysInfo`
- `uint16_t` * `pAddHDRSysInfo`
- `nas_AddSysInfo` * `pAddGSMSysInfo`
- `nas_AddSysInfo` * `pAddWCDMASysInfo`
- `uint16_t` * `pAddLTESysInfo`
- `nas_CallBarringSysInfo` * `pGSMCallBarringSysInfo`
- `nas_CallBarringSysInfo` * `pWCDMACallBarringSysInfo`
- `uint8_t` * `pLTEVoiceSupportSysInfo`
- `uint8_t` * `pGSMCipherDomainSysInfo`
- `uint8_t` * `pWCDMACipherDomainSysInfo`

8.327.1 Detailed Description

Parameters

<i>pCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> • See <code>SrvStatusInfo</code> for more information.
<i>pHDRSrvStatusInfo</i>	<ul style="list-style-type: none"> • See <code>SrvStatusInfo</code> for more information.
<i>pGSMSrvStatusInfo</i>	<ul style="list-style-type: none"> • See <code>GSMSrvStatusInfo</code> for more information.

<i>pWCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pLTESrvStatus-Info</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> • See CDMASysInfo for more information.
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> • See HDRSysInfo for more information.
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> • See GSMSysInfo for more information.
<i>pWCDMASys-Info</i>	<ul style="list-style-type: none"> • See WCDMASysInfo for more information.
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> • See LTESysInfo for more information.
<i>pAddCDMASys-Info</i>	<ul style="list-style-type: none"> • See AddCDMASysInfo for more information.
<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.
<i>pAddGSMSys-Info</i>	<ul style="list-style-type: none"> • See AddSysInfo for more information.
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> • See AddSysInfo for more information.
<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.
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See CallBarringSysInfo for more information.
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See CallBarringSysInfo for more information.
<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

<i>pGSMCipherDomainSysInfo</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
<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

8.327.2 Field Documentation

8.327.2.1 **nas_AddCDMASysInfo*** `unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo`

8.327.2.2 **nas_AddSysInfo*** `unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo`

8.327.2.3 **uint16_t*** `unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo`

8.327.2.4 **uint16_t*** `unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo`

8.327.2.5 **nas_AddSysInfo*** `unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo`

8.327.2.6 **nas_SrvStatusInfo*** `unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo`

8.327.2.7 **nas_CDMASysInfo*** `unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo`

8.327.2.8 **nas_CallBarringSysInfo*** `unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo`

8.327.2.9 **uint8_t*** `unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo`

8.327.2.10 **nas_GSMSrvStatusInfo*** `unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo`

8.327.2.11 **nas_GSMSysInfo*** `unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo`

8.327.2.12 **nas_SrvStatusInfo*** `unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo`

8.327.2.13 **nas_HDRSysInfo*** `unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo`

8.327.2.14 **nas_GSMSrvStatusInfo*** `unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo`

8.327.2.15 **nas_LTESysInfo*** `unpack_nas_SLQSGetSysInfo_t::pLTESysInfo`

8.327.2.16 **uint8_t*** `unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo`

8.327.2.17 **nas_CallBarringSysInfo*** `unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo`

8.327.2.18 uint8_t* unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

8.327.2.19 nas_GSMsSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

8.327.2.20 nas_WCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

8.328 unpack_nas_SLQSGetSysSelectionPref_t Struct Reference

Data Fields

- uint8_t * [pEmerMode](#)
- uint16_t * [pModePref](#)
- uint64_t * [pBandPref](#)
- uint16_t * [pPRLPref](#)
- uint16_t * [pRoamPref](#)
- uint64_t * [pLTEBandPref](#)
- uint8_t * [pNetSelPref](#)
- uint32_t * [pSrvDomainPref](#)
- uint32_t * [pGWAcqOrderPref](#)

8.328.1 Detailed Description

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
	<ul style="list-style-type: none"> – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved <p style="text-align: right;">Generated on Tue May 31 2016 14:24:35 for LinuxQMSDK by Doxygen</p>

<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

8.328.2 Field Documentation

8.328.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

8.328.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

8.328.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

8.328.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

8.328.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

8.328.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

8.328.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

8.328.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

8.328.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

8.329 unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference

Data Fields

- [nas_GERANInfo](#) * [pGERANInfo](#)
- [nas_UMTSInfo](#) * [pUMTSInfo](#)
- [nas_CDMAInfo](#) * [pCDMAInfo](#)
- [nas_LTEInfoIntrafreq](#) * [pLTEInfoIntrafreq](#)
- [nas_LTEInfoInterfreq](#) * [pLTEInfoInterfreq](#)
- [nas_LTEInfoNeighboringGSM](#) * [pLTEInfoNeighboringGSM](#)
- [nas_LTEInfoNeighboringWCDMA](#) * [pLTEInfoNeighboringWCDMA](#)
- uint32_t * [pUMTSCellID](#)
- [nas_WCDMAInfoLTENeighborCell](#) * [pWCDMAInfoLTENeighborCell](#)

8.329.1 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.
<i>pUMTSInfo</i>	<ul style="list-style-type: none"> See nas_UMTSInfo for more information.
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> See nas_CDMAInfo for more information.
<i>pLTEInfo-Intrafreq</i>	<ul style="list-style-type: none"> See nas_LTEInfoIntrafreq for more information.
<i>pLTEInfo-Interfreq</i>	<ul style="list-style-type: none"> See nas_LTEInfoInterfreq for more information.
<i>pLTEInfo-NeighboringGSM</i>	<ul style="list-style-type: none"> See nas_LTEInfoNeighboringGSM for more information.
<i>pLTEInfo-NeighboringWCDMA</i>	<ul style="list-style-type: none"> See nas_LTEInfoNeighboringWCDMA for more information.
<i>pUMTSCellID</i>	<ul style="list-style-type: none"> Cell ID. 0xFFFFFFFF indicates cell ID information is not present.
<i>pWCDMAInfoLT-ENeighborCell</i>	<ul style="list-style-type: none"> See nas_WCDMAInfoLTENeighborCell for more information.

8.329.2 Field Documentation

8.329.2.1 `nas_CDMAInfo*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo`8.329.2.2 `nas_GERANInfo*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo`8.329.2.3 `nas_LTEInfoInterfreq*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq`8.329.2.4 `nas_LTEInfoIntrafreq*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq`8.329.2.5 `nas_LTEInfoNeighboringGSM*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM`8.329.2.6 `nas_LTEInfoNeighboringWCDMA*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA`8.329.2.7 `uint32_t*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID`8.329.2.8 `nas_UMTSInfo*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo`8.329.2.9 `nas_WCDMAInfoLTENeighborCell*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell`

8.330 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

- [cdmaSSInfo](#) [CDMASSInfo](#)
- [hdrSSInfo](#) [HDRSSInfo](#)
- [int8_t](#) [GSMSSInfo](#)
- [cdmaSSInfo](#) [WCDMASSInfo](#)
- [lteSSInfo](#) [LTESSInfo](#)

8.330.1 Detailed Description

Parameters

<i>CDMASSInfo</i>	CDMA Signal Strength Information
<i>HDRSSInfo</i>	HDR Signal Strength Information
<i>GSMSSInfo</i>	GSM signal strength is the RSSI in dBm.
<i>WCDMASSInfo</i>	WCDMA Signal Strength Information
<i>LTESSInfo</i>	LTE Signal Strength Information

8.330.2 Field Documentation

8.330.2.1 [cdmaSSInfo](#) [unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo](#)

8.330.2.2 [int8_t](#) [unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo](#)

8.330.2.3 [hdrSSInfo](#) [unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo](#)

8.330.2.4 [lteSSInfo](#) [unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo](#)

8.330.2.5 [cdmaSSInfo](#) [unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo](#)

8.331 unpack_nas_SLQSNasSigInfoCallback_t Struct Reference

Data Fields

- [cdmaSSInfo](#) * [pCDMASigInfo](#)
- [hdrSSInfo](#) * [pHDRSigInfo](#)
- [int8_t](#) * [pGSMSigInfo](#)
- [cdmaSSInfo](#) * [pWCDMASigInfo](#)
- [lteSSInfo](#) * [pLTESigInfo](#)
- [int8_t](#) * [pRscp](#)
- [tdscdmaSigInfoExt](#) * [pTDSCDMASigInfoExt](#)

8.331.1 Detailed Description

Parameters

<i>pCDMASigInfo</i>	CDMA SS info
<i>pHDRSigInfo</i>	HDR SS info
<i>pGSMSigInfo</i>	GSM signal info
<i>pWCDMASigInfo</i>	WCDMA signal info
<i>pLTESigInfo</i>	LTE signal info
<i>pRscp</i>	RSCP of the Primary Common Control Physical Channel
<i>pTDSCDMASig-InfoExt</i>	extra CDMA sig info

8.331.2 Field Documentation

8.331.2.1 **cdmaSSInfo*** unpack_nas_SLQSNasSigInfoCallback_t::pCDMASigInfo

8.331.2.2 **int8_t*** unpack_nas_SLQSNasSigInfoCallback_t::pGSMSigInfo

8.331.2.3 **hdrSSInfo*** unpack_nas_SLQSNasSigInfoCallback_t::pHDRSigInfo

8.331.2.4 **lteSSInfo*** unpack_nas_SLQSNasSigInfoCallback_t::pLTESigInfo

8.331.2.5 **int8_t*** unpack_nas_SLQSNasSigInfoCallback_t::pRscp

8.331.2.6 **tdscdmaSigInfoExt*** unpack_nas_SLQSNasSigInfoCallback_t::pTDSCDMASigInfoExt

8.331.2.7 **cdmaSSInfo*** unpack_nas_SLQSNasSigInfoCallback_t::pWCDMASigInfo

8.332 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

Data Fields

- [nas_CommlInfo](#) [commonInfo](#)
- [nas_LTEInfo](#) * [pLTEInfo](#)

8.332.1 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 CommlInfo for more information
<i>pLTEInfo</i>	(optional) <ul style="list-style-type: none"> • See LTEInfo for more information

8.332.2 Field Documentation

8.332.2.1 `nas_CommInfo` `unpack_nas_SLQSNasSwiModemStatus_t::commonInfo`

8.332.2.2 `nas_LTEInfo*` `unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo`

8.333 `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t` Struct Reference

Data Fields

- [NASQmiCbKnasSwiOTAMessageInd Info](#)
- `uint16_t` [Tlvresult](#)

8.333.1 Detailed Description

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 NASQmiCbKnasSwiOTAMessageInd for more details
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.333.2 Field Documentation

8.333.2.1 `NASQmiCbKnasSwiOTAMessageInd` `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info`

8.333.2.2 `uint16_t` `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult`

8.334 `unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t` Struct Reference

Data Fields

- [NASQmiCbKnasSystemSelPrefInd Info](#)
- `uint16_t` [Tlvresult](#)

8.334.1 Detailed Description

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

8.334.2 Field Documentation

8.334.2.1 [NASQmiCbkNasSystemSelPrefInd](#) unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

8.334.2.2 [uint16_t](#) unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult

8.335 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

Data Fields

- [uint8_t](#) [ValidityCW0](#)
- [uint8_t](#) [CQIValueCW0](#)
- [uint8_t](#) [ValidityCW1](#)
- [uint8_t](#) [CQIValueCW1](#)

8.335.1 Detailed Description

Parameters

<i>ValidityCW0[OUT]</i>	<ul style="list-style-type: none"> Values <ul style="list-style-type: none"> 0- Invalid. 1- Valid.
<i>CQIValueCW0[OUT]</i>	<ul style="list-style-type: none"> Values <ul style="list-style-type: none"> Range 0~15
<i>ValidityCW1[OUT]</i>	<ul style="list-style-type: none"> Values <ul style="list-style-type: none"> 0- Invalid. 1- Valid.

<i>CQIValueCW1[-OUT]</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – Range 0~15
--------------------------	--

8.335.2 Field Documentation

8.335.2.1 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0`

8.335.2.2 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1`

8.335.2.3 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0`

8.335.2.4 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1`

8.336 `unpack_nas_SLQSSysInfoCallback_t` Struct Reference

Data Fields

- `nas_SrvStatusInfo` * `pCDMASrvStatusInfo`
- `nas_SrvStatusInfo` * `pHDRSrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pGSMSrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pWCDMASrvStatusInfo`
- `nas_GSMSrvStatusInfo` * `pLTESrvStatusInfo`
- `nas_CDMASysInfo` * `pCDMASysInfo`
- `nas_HDRSysInfo` * `pHDRSysInfo`
- `nas_GSMSysInfo` * `pGSMSysInfo`
- `nas_WCDMASysInfo` * `pWCDMASysInfo`
- `nas_LTESysInfo` * `pLTESysInfo`
- `nas_AddCDMASysInfo` * `pAddCDMASysInfo`
- `uint16_t` * `pAddHDRSysInfo`
- `nas_AddSysInfo` * `pAddGSMSysInfo`
- `nas_AddSysInfo` * `pAddWCDMASysInfo`
- `uint16_t` * `pAddLTESysInfo`
- `nas_CallBarringSysInfo` * `pGSMCallBarringSysInfo`
- `nas_CallBarringSysInfo` * `pWCDMACallBarringSysInfo`
- `uint8_t` * `pLTEVoiceSupportSysInfo`
- `uint8_t` * `pGSMCipherDomainSysInfo`
- `uint8_t` * `pWCDMACipherDomainSysInfo`
- `uint8_t` * `pSysInfoNoChange`

8.336.1 Detailed Description

Parameters

<i>pCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> • See <code>SrvStatusInfo</code> for more information.
----------------------------	--

<i>pHDRSrvStatus-Info</i>	<ul style="list-style-type: none"> • See SrvStatusInfo for more information.
<i>pGSMSrvStatus-Info</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pWCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pLTESrvStatus-Info</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> • See CDMA SysInfo for more information.
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> • See HDRSysInfo for more information.
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> • See GSMSysInfo for more information.
<i>pWCDMASys-Info</i>	<ul style="list-style-type: none"> • See WCDMASysInfo for more information.
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> • See LTESysInfo for more information.
<i>pAddCDMASys-Info</i>	<ul style="list-style-type: none"> • See AddCDMASysInfo for more information.
<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.
<i>pAddGSMSys-Info</i>	<ul style="list-style-type: none"> • See AddSysInfo for more information.
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> • See AddSysInfo for more information.
<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.
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See CallBarringSysInfo for more information.
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See CallBarringSysInfo for more information.

<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
<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
<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
<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

8.336.2 Field Documentation

8.336.2.1 **nas_AddCDMASysInfo*** `unpack_nas_SLQSSysInfoCallback_t::pAddCDMASysInfo`

8.336.2.2 **nas_AddSysInfo*** `unpack_nas_SLQSSysInfoCallback_t::pAddGSM SysInfo`

8.336.2.3 **uint16_t*** `unpack_nas_SLQSSysInfoCallback_t::pAddHDRSysInfo`

8.336.2.4 **uint16_t*** `unpack_nas_SLQSSysInfoCallback_t::pAddLTESysInfo`

8.336.2.5 **nas_AddSysInfo*** `unpack_nas_SLQSSysInfoCallback_t::pAddWCDMASysInfo`

8.336.2.6 **nas_SrvStatusInfo*** `unpack_nas_SLQSSysInfoCallback_t::pCDMASrvStatusInfo`

8.336.2.7 **nas_CDMASysInfo*** `unpack_nas_SLQSSysInfoCallback_t::pCDMASysInfo`

8.336.2.8 **nas_CallBarringSysInfo*** `unpack_nas_SLQSSysInfoCallback_t::pGSMCallBarringSysInfo`

8.336.2.9 **uint8_t*** `unpack_nas_SLQSSysInfoCallback_t::pGSMCipherDomainSysInfo`

- 8.336.2.10 nas_GSMsSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_t::pGSMsSrvStatusInfo
- 8.336.2.11 nas_GSMsSysInfo* unpack_nas_SLQSSysInfoCallback_t::pGSMsSysInfo
- 8.336.2.12 nas_SrvStatusInfo* unpack_nas_SLQSSysInfoCallback_t::pHdRSrvStatusInfo
- 8.336.2.13 nas_HdRSysInfo* unpack_nas_SLQSSysInfoCallback_t::pHdRSysInfo
- 8.336.2.14 nas_GSMsSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_t::pLTESrvStatusInfo
- 8.336.2.15 nas_LTESysInfo* unpack_nas_SLQSSysInfoCallback_t::pLTESysInfo
- 8.336.2.16 uint8_t* unpack_nas_SLQSSysInfoCallback_t::pLTEVoiceSupportSysInfo
- 8.336.2.17 uint8_t* unpack_nas_SLQSSysInfoCallback_t::pSysInfoNoChange
- 8.336.2.18 nas_CallBarringSysInfo* unpack_nas_SLQSSysInfoCallback_t::pWCDMACallBarringSysInfo
- 8.336.2.19 uint8_t* unpack_nas_SLQSSysInfoCallback_t::pWCDMACipherDomainSysInfo
- 8.336.2.20 nas_GSMsSrvStatusInfo* unpack_nas_SLQSSysInfoCallback_t::pWCDMASrvStatusInfo
- 8.336.2.21 nas_WCDMASysInfo* unpack_nas_SLQSSysInfoCallback_t::pWCDMASysInfo

8.337 unpack_omaDmConfigTlv_t Struct Reference

Data Fields

- uint8_t [state](#)
- uint8_t [userInputReq](#)
- uint16_t [userInputTimeout](#)
- uint16_t [alertmsglength](#)
- uint8_t [alertmsg](#) [256]

8.337.1 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>user_input_req</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>user_input_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)

8.337.2 Field Documentation

8.337.2.1 `uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]`

8.337.2.2 `uint16_t unpack_omaDmConfigTlv_t::alertmsglength`

8.337.2.3 `uint8_t unpack_omaDmConfigTlv_t::state`

8.337.2.4 `uint8_t unpack_omaDmConfigTlv_t::userInputReq`

8.337.2.5 `uint16_t unpack_omaDmConfigTlv_t::userInputTimeout`

8.338 `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`

8.338.1 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>user_input_req</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>user_input_timeout</i>	<ul style="list-style-type: none"> • Timeout for user input in minutes. A value of 0 means no time-out
<i>fw_dload_size</i>	<ul style="list-style-type: none"> • The size (in bytes) of the firmware update package
<i>fw_dload_complete</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_complete_status</i>	<ul style="list-style-type: none"> • See table below.
<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

8.338.2 Field Documentation

- 8.338.2.1 `uint8_t unpack_omaDmFotaTlv_t::description[256]`
- 8.338.2.2 `uint16_t unpack_omaDmFotaTlv_t::descriptionlength`
- 8.338.2.3 `uint32_t unpack_omaDmFotaTlv_t::fwdloadsize`
- 8.338.2.4 `uint32_t unpack_omaDmFotaTlv_t::fwloadComplete`
- 8.338.2.5 `uint16_t unpack_omaDmFotaTlv_t::namelength`
- 8.338.2.6 `uint8_t unpack_omaDmFotaTlv_t::package_name[256]`
- 8.338.2.7 `uint8_t unpack_omaDmFotaTlv_t::sessionType`
- 8.338.2.8 `uint8_t unpack_omaDmFotaTlv_t::severity`
- 8.338.2.9 `uint8_t unpack_omaDmFotaTlv_t::state`
- 8.338.2.10 `uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus`
- 8.338.2.11 `uint8_t unpack_omaDmFotaTlv_t::userInputReq`
- 8.338.2.12 `uint16_t unpack_omaDmFotaTlv_t::userInputTimeout`
- 8.338.2.13 `uint8_t unpack_omaDmFotaTlv_t::version[256]`
- 8.338.2.14 `uint16_t unpack_omaDmFotaTlv_t::versionlength`

8.339 `unpack_omaDmNotificationsTlv_t` Struct Reference

Data Fields

- `uint8_t` [notification](#)
- `uint16_t` [sessionStatus](#)

8.339.1 Field Documentation

8.339.1.1 `uint8_t unpack_omaDmNotificationsTlv_t::notification`

8.339.1.2 `uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus`

8.340 unpack_qmi_t Struct Reference

Data Fields

- enum [msgtype type](#)
- `uint16_t` [msgid](#)
- `uint16_t` [xid](#)

8.340.1 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

8.340.2 Field Documentation

8.340.2.1 `uint16_t unpack_qmi_t::msgid`

8.340.2.2 `enum msgtype unpack_qmi_t::type`

8.340.2.3 `uint16_t unpack_qmi_t::xid`

8.341 unpack_qos_dataRate_t Struct Reference

Data Fields

- `uint32_t` [dataRateMax](#)
- `uint32_t` [guaranteedRate](#)

8.341.1 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)

8.341.2 Field Documentation

8.341.2.1 `uint32_t unpack_qos_dataRate_t::dataRateMax`

8.341.2.2 `uint32_t unpack_qos_dataRate_t::guaranteedRate`

8.342 unpack_qos_IPv4Addr_t Struct Reference

Data Fields

- uint32_t [addr](#)
- uint32_t [subnetMask](#)

8.342.1 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

8.342.2 Field Documentation

8.342.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

8.342.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

8.343 unpack_qos_IPv6Addr_t Struct Reference

Data Fields

- uint8_t [addr](#) [16]
- uint8_t [prefixLen](#)

8.343.1 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.

8.343.2 Field Documentation

8.343.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

8.343.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

8.344 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

- uint8_t [val](#)
- uint8_t [mask](#)

8.344.1 Detailed Description

This structure contains the IPv6 filter traffic class

Parameters

<i>val</i>	The traffic class value
<i>mask</i>	<p>The packet matches the traffic class filter if: (IPv6_filter_traffic_class_val and IPv6_filter_traffic_class_mask) == (Traffic class value in the IP packet & IPv6_filter_traffic_class_mask)</p> <p>Example:</p> <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

8.344.2 Field Documentation

8.344.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.344.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.345 unpack_qos_pktErrRate_t Struct Reference

Data Fields

- uint16_t [multiplier](#)
- uint16_t [exponent](#)

8.345.1 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)

8.345.2 Field Documentation

8.345.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.345.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.346 unpack_qos_Port_t Struct Reference

Data Fields

- uint16_t [port](#)
- uint16_t [range](#)

8.346.1 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

8.346.2 Field Documentation

8.346.2.1 uint16_t [unpack_qos_Port_t::port](#)

8.346.2.2 uint16_t [unpack_qos_Port_t::range](#)

8.347 [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

8.347.1 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 LIBPACK_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 LIBPACK_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.

8.347.2 Field Documentation

8.347.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`

8.347.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`

8.347.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`

8.347.2.4 `uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters`

8.347.2.5 `uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters`

8.347.2.6 `unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState`

8.347.2.7 `unpack_qos_swiQosFilter_t` `unpack_qos_QosFlowInfo_t::RxQFilter[25]`

8.347.2.8 `unpack_qos_swiQosFlow_t` `unpack_qos_QosFlowInfo_t::RxQFlowGranted`

8.347.2.9 `unpack_qos_swiQosFilter_t` `unpack_qos_QosFlowInfo_t::TxQFilter[25]`

8.347.2.10 `unpack_qos_swiQosFlow_t` `unpack_qos_QosFlowInfo_t::TxQFlowGranted`

8.348 `unpack_qos_QosFlowInfoState_t` Struct Reference

Data Fields

- `uint32_t id`
- `uint8_t isNewFlow`
- `uint8_t state`

8.348.1 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>	<p>This indicates that the flow that was added/modified/deleted:</p> <ul style="list-style-type: none"> • 0x01 – Flow activated • 0x02 – Flow modified • 0x03 – Flow deleted • 0x04 – Flow suspended • 0x05 – Flow enabled • 0x06 – Flow disabled

8.348.2 Field Documentation

8.348.2.1 `uint32_t` `unpack_qos_QosFlowInfoState_t::id`

8.348.2.2 `uint8_t` `unpack_qos_QosFlowInfoState_t::isNewFlow`

8.348.2.3 `uint8_t` `unpack_qos_QosFlowInfoState_t::state`

8.349 `unpack_qos_SLQSQosGetNetworkStatus_t` Struct Reference

Data Fields

- `uint8_t NWQoSStatus`

8.349.1 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
--------------------	--

8.349.2 Field Documentation

8.349.2.1 `uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus`8.350 `unpack_qos_SLQSQosSwiReadApnExtraParams_t` Struct Reference

Data Fields

- `uint32_t` [apnId](#)
- `uint8_t` [ambr_ul](#)
- `uint8_t` [ambr_dl](#)
- `uint8_t` [ambr_ul_ext](#)
- `uint8_t` [ambr_dl_ext](#)
- `uint8_t` [ambr_ul_ext2](#)
- `uint8_t` [ambr_dl_ext2](#)

8.350.1 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
<i>ambr_ul</i>	<ul style="list-style-type: none"> • APN AMBR uplink • APN AMBR uplink values from 1 kbps to 8640 kbps
<i>ambr_dl</i>	<ul style="list-style-type: none"> • APN AMBR downlink • APN AMBR downlink values from 1 kbps to 8640 kbps
<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

<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
<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
<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

8.350.2 Field Documentation

8.350.2.1 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl`

8.350.2.2 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl_ext`

8.350.2.3 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl_ext2`

8.350.2.4 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_ul`

8.350.2.5 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_ul_ext`

8.350.2.6 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_ul_ext2`

8.350.2.7 `uint32_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::apnId`

8.351 unpack_qos_SLQSQoSwiReadDataStats_t Struct Reference

Data Fields

- `uint32_t apnId`
- `uint32_t total_tx_pkt`
- `uint32_t total_tx_pkt_drp`
- `uint32_t total_rx_pkt`
- `uint64_t total_tx_bytes`
- `uint64_t total_tx_bytes_drp`
- `uint64_t total_rx_bytes`
- `uint32_t numQoSFlow`
- `unpack_QoSFlowStat_t qosFlow [10]`

8.351.1 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
<i>total_tx_pkt</i>	<ul style="list-style-type: none"> • sum of all packets sent
<i>total_tx_pkt_drp</i>	<ul style="list-style-type: none"> • sum of all(TX) packets dropped
<i>total_rx_pkt</i>	<ul style="list-style-type: none"> • sum of all packets received
<i>total_tx_bytes</i>	<ul style="list-style-type: none"> • sum of all bytes sent
<i>total_tx_bytes_drp</i>	<ul style="list-style-type: none"> • sum of all(TX) bytes dropped
<i>total_rx_bytes</i>	<ul style="list-style-type: none"> • number of received bytes for the QoS flow ID
<i>numQosFlow</i>	<ul style="list-style-type: none"> • pointer to number of QoS flow Stat
<i>qosFlow[LIBPACK_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 LIBPACK_MAX_QOS_FLOW_PER_APN_STATS for more information

8.351.2 Field Documentation

8.351.2.1 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId

8.351.2.2 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow

8.351.2.3 unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]

8.351.2.4 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes

8.351.2.5 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt

8.351.2.6 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes

8.351.2.7 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp

8.351.2.8 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt

8.351.2.9 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp

8.352 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

Data Fields

- uint8_t [NumFlows](#)
- [unpack_qos_QosFlowInfo_t](#) [QosFlowInfo](#) [8]

8.352.1 Detailed Description

Structure with QoS event details

Parameters

<i>NumFlows</i>	<ul style="list-style-type: none">• Number of QoS flows available
<i>QosFlowInfo</i>	<ul style="list-style-type: none">• The Qos flow details, please check unpack_qos_QosFlowInfo_t for more information• See LIBPACK_MAX_QOS_FLOWS for more information

8.352.2 Field Documentation

8.352.2.1 uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows

8.352.2.2 [unpack_qos_QosFlowInfo_t](#) unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]

8.353 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

Data Fields

- uint8_t [status](#)

8.353.1 Detailed Description

Structure with network's QoS status

Parameters

<i>status</i>	Network QoS support status <ul style="list-style-type: none">• 0x00 – Current network does not support QoS• 0x01 – Current network supports QoS
---------------	--

Note

- Technology Supported: CDMA

8.353.2 Field Documentation

8.353.2.1 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

8.354 `unpack_qos_SLQSSetQosPriEventCallback_ind_t` Struct Reference

Data Fields

- `uint16_t event`

8.354.1 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
--------------	---

8.354.2 Field Documentation

8.354.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

8.355 `unpack_qos_SLQSSetQosStatusCallback_ind_t` Struct Reference

Data Fields

- `uint32_t id`
- `uint8_t status`
- `uint8_t event`
- `uint8_t reason`

8.355.1 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
<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
<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
<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
Generated on Tue May 31 2016 14:24:34 by Doxygen 1.8.14 0x16 - QMI_QOS_NETWORK_BUSY	

8.355.2 Field Documentation

8.355.2.1 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event`

8.355.2.2 `uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id`

8.355.2.3 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason`

8.355.2.4 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status`

8.356 `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`

8.356.1 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

8.356.2 Field Documentation

8.356.2.1 `uint32_t unpack_qos_swiQosFilter_t::EspSpi`

8.356.2.2 `uint16_t unpack_qos_swiQosFilter_t::Id`

8.356.2.3 `uint8_t unpack_qos_swiQosFilter_t::index`

8.356.2.4 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr`

8.356.2.5 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr`

8.356.2.6 `unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos`

8.356.2.7 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr`

8.356.2.8 `uint32_t unpack_qos_swiQosFilter_t::IPv6Label`

8.356.2.9 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr`

8.356.2.10 `unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls`

8.356.2.11 `uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available`

8.356.2.12 `uint8_t unpack_qos_swiQosFilter_t::is_Id_Available`

8.356.2.13 uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available

8.356.2.14 uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available

8.356.2.15 uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available

8.356.2.16 uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available

8.356.2.17 uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available

8.356.2.18 uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available

8.356.2.19 uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available

8.356.2.20 uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available

8.356.2.21 uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available

8.356.2.22 uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available

8.356.2.23 uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available

8.356.2.24 uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available

8.356.2.25 uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available

8.356.2.26 uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available

8.356.2.27 uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available

8.356.2.28 uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto

8.356.2.29 uint16_t unpack_qos_swiQosFilter_t::Precedence

8.356.2.30 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort

8.356.2.31 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort

8.356.2.32 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort

8.356.2.33 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort

8.356.2.34 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort

8.356.2.35 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort

8.356.2.36 uint8_t unpack_qos_swiQosFilter_t::version

8.357 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

8.357.1 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: • 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>MaxAllowedPkt-Sz</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-HdIPri</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

8.357.2 Field Documentation

8.357.2.1 `unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate`

8.357.2.2 `uint8_t unpack_qos_swiQosFlow_t::index`

8.357.2.3 `uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.357.2.4 `uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.357.2.5 `uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available`

8.357.2.6 `uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.357.2.7 `uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.357.2.8 `uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.357.2.9 `uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.357.2.10 `uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.357.2.11 `uint8_t unpack_qos_swiQosFlow_t::is_TokenBucket_Available`

8.357.2.12 `uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.357.2.13 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.357.2.14 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

8.357.2.15 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available`

8.357.2.16 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available`

- 8.357.2.17 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available`
- 8.357.2.18 `uint32_t unpack_qos_swiQosFlow_t::Jitter`
- 8.357.2.19 `uint32_t unpack_qos_swiQosFlow_t::Latency`
- 8.357.2.20 `uint8_t unpack_qos_swiQosFlow_t::LteQci`
- 8.357.2.21 `uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz`
- 8.357.2.22 `uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz`
- 8.357.2.23 `unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate`
- 8.357.2.24 `uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2`
- 8.357.2.25 `unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket`
- 8.357.2.26 `uint8_t unpack_qos_swiQosFlow_t::TrafficClass`
- 8.357.2.27 `uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri`
- 8.357.2.28 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn`
- 8.357.2.29 `uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER`
- 8.357.2.30 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd`
- 8.357.2.31 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri`

8.358 `unpack_qos_tokenBucket_t` Struct Reference

Data Fields

- `uint32_t` [peakRate](#)
- `uint32_t` [tokenRate](#)
- `uint32_t` [bucketSz](#)

8.358.1 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

8.358.2 Field Documentation

- 8.358.2.1 `uint32_t unpack_qos_tokenBucket_t::bucketSz`

8.358.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

8.358.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

8.359 unpack_qos_Tos_t Struct Reference

Data Fields

- uint8_t [val](#)
- uint8_t [mask](#)

8.359.1 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.

8.359.2 Field Documentation

8.359.2.1 uint8_t unpack_qos_Tos_t::mask

8.359.2.2 uint8_t unpack_qos_Tos_t::val

8.360 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](#)

8.360.1 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

8.360.2 Field Documentation

8.360.2.1 `uint32_t` `unpack_QosFlowStat_t::bearerId`

8.360.2.2 `uint64_t` `unpack_QosFlowStat_t::tx_bytes`

8.360.2.3 `uint64_t` `unpack_QosFlowStat_t::tx_bytes_drp`

8.360.2.4 `uint32_t` `unpack_QosFlowStat_t::tx_pkt`

8.360.2.5 `uint32_t` `unpack_QosFlowStat_t::tx_pkt_drp`

8.361 `unpack_sms_SendSMS_t` Struct Reference

Data Fields

- `uint16_t` `messageID`
- `uint32_t` `messageFailureCode`

8.361.1 Detailed Description

Parameters

<i>messageID</i>	<ul style="list-style-type: none"> • WMS message ID
<i>messageFailure-Code</i>	<ul style="list-style-type: none"> • pointer to message failure code. If cause code is not provided, then value will be 0xF-FFFFFF

8.361.2 Field Documentation

8.361.2.1 `uint32_t` `unpack_sms_SendSMS_t::messageFailureCode`

8.361.2.2 uint16_t unpack_sms_SendSMS_t::messageID

8.362 unpack_sms_SetNewSMSCallback_ind_t Struct Reference

Data Fields

- struct [newMTMessageTlv](#) NewMMTlv
- struct [transferRouteMessageTlv](#) TRMessageTlv
- struct [messageModeTlv](#) MMTlv
- struct [sMSEtwsMessageTlv](#) ETWSTlv
- struct [eTWSPLMNInfoTlv](#) ETWSPLMNTlv
- struct [sMSCAddressTlv](#) SMSCtlv
- struct [sMSONIMSTlv](#) IMSTlv

8.362.1 Detailed Description

Parameters

<i>NewMMTlv</i>	<ul style="list-style-type: none"> • MT message
<i>TRMessageTlv</i>	<ul style="list-style-type: none"> • Transfer Route MT Message • See transferRouteMessageTlv for more information
<i>MMTlv</i>	<ul style="list-style-type: none"> • Message mode • See messageModeTlv for more information
<i>ETWSTlv</i>	<ul style="list-style-type: none"> • ETWS Message • See sMSEtwsMessageTlv for more information
<i>ETWSPLMNTlv</i>	<ul style="list-style-type: none"> • ETWS PLMN Information • See eTWSPLMNInfoTlv for more information
<i>SMSCtlv</i>	<ul style="list-style-type: none"> • SMSC Address • See sMSCAddressTlv for more information
<i>IMSTlv</i>	<ul style="list-style-type: none"> • SMS on IMS • See sMSONIMSTlv for more information

8.362.2 Field Documentation

8.362.2.1 struct `eTWSPLMNInfoTlv` `unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv`

8.362.2.2 struct `sMSEtwsMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv`

8.362.2.3 struct `sMSOnIMSTlv` `unpack_sms_SetNewSMSCallback_ind_t::IMSTlv`

8.362.2.4 struct `messageModeTlv` `unpack_sms_SetNewSMSCallback_ind_t::MMTlv`

8.362.2.5 struct `newMTMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv`

8.362.2.6 struct `sMSCAddressTlv` `unpack_sms_SetNewSMSCallback_ind_t::SMSCSTlv`

8.362.2.7 struct `transferRouteMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv`

8.363 `unpack_sms_SetNewSMSCallback_t` Struct Reference

8.364 `unpack_sms_SLQSDeleteSMS_t` Struct Reference

8.365 `unpack_sms_SLQSGetSMS_t` Struct Reference

Data Fields

- uint32_t [messageTag](#)
- uint32_t [messageFormat](#)
- uint32_t [messageSize](#)
- uint8_t [message](#) [255]

8.365.1 Detailed Description

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
-------------------	---

<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">• 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.

Parameters

<i>message</i>	<ul style="list-style-type: none">• The message contents array
----------------	--

8.365.2 Field Documentation

8.365.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[255]

8.365.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

8.365.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

8.365.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

8.366 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

- uint32_t [messageListSize](#)
- [qmiSmsMessageList](#) [messageList](#) [255]

8.366.1 Detailed Description

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.
<i>messageList</i>	<ul style="list-style-type: none">• Message List• See qmiSmsMessageList for more information

8.366.2 Field Documentation

8.366.2.1 `qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]`

8.366.2.2 `uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize`

8.367 `unpack_sms_SLQSModifySMSStatus_t` Struct Reference

8.368 `unpack_sms_SLQSWmsMemoryFullCallBack_ind_t` Struct Reference

Data Fields

- `uint8_t storageType`
- `uint8_t messageMode`

8.368.1 Detailed Description

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>messageMode</i>	<ul style="list-style-type: none"> • 0x00 - CDMA, LTE (if network type is CDMA) • 0x01 - GW, LTE (if network type is UMTS)

8.368.2 Field Documentation

8.368.2.1 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode`

8.368.2.2 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType`

8.369 `unpack_swiloc_SwiLocGetAutoStart_t` Struct Reference

Data Fields

- `uint8_t function`
- `int function_reported`
- `uint8_t fix_type`
- `int fix_type_reported`
- `uint8_t max_time`
- `int max_time_reported`
- `uint32_t max_dist`
- `int max_dist_reported`
- `uint32_t fix_rate`
- `int fix_rate_reported`

8.369.1 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>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
<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
<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
<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

<i>fix_rate_ - reported</i>	<ul style="list-style-type: none"> • 0 - not reported by modem • 1 - reported by modem
---------------------------------	--

8.369.2 Field Documentation

8.369.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

8.369.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

8.369.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

8.369.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

8.369.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

8.369.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

8.369.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

8.369.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

8.369.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

8.369.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

8.370 unpack_swima_SLQSOMADMAAlertCallback_ind_t Struct Reference

Data Fields

- uint32_t [eventType](#)
- [unpack_omaDmFotaTlv_t](#) [SessionInfoFota](#)
- [unpack_omaDmConfigTlv_t](#) [SessionInfoConfig](#)
- [unpack_omaDmNotificationsTlv_t](#) [SessionInfoNotification](#)

8.370.1 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
<i>SessionInfo-Fota[OUT]</i>	<ul style="list-style-type: none"> • See unpack_omaDmFotaTlv_t for more information
<i>SessionInfo-Config[OUT]</i>	<ul style="list-style-type: none"> • See unpack_omaDmConfigTlv_t for more information
<i>SessionInfo-Notification[OUT]</i>	<ul style="list-style-type: none"> • See unpack_omaDmNotificationsTlv_t for more information

8.370.2 Field Documentation

8.370.2.1 uint32_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::eventType

8.370.2.2 unpack_omaDmConfigTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoConfig

8.370.2.3 unpack_omaDmFotaTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoFota

8.370.2.4 unpack_omaDmNotificationsTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoNotification

8.371 unpack_swisma_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint8_t [Status](#)
- uint16_t [UpdateCompleteStatus](#)
- uint8_t [Severity](#)
- uint16_t [SourceLength](#)
- uint8_t [Source](#) [255]
- uint16_t [PkgNameLength](#)
- uint8_t [PkgName](#) [255]
- uint16_t [PkgDescLength](#)
- uint8_t [PkgDescription](#) [255]
- uint16_t [DateLength](#)
- uint8_t [Date](#) [255]
- uint16_t [TimeLength](#)
- uint8_t [Time](#) [255]
- uint8_t [SessionType](#)
- uint8_t [SessionState](#)
- uint16_t [RetryCount](#)

8.371.1 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
<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
<i>Severity</i>	<ul style="list-style-type: none"> • 1 byte parameter indicating severity <ul style="list-style-type: none"> – 0x01 - Mandatory – 0x02 - Optional
<i>SourceLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Vendor Name String in Bytes.
<i>Source</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Vendor Name in ASCII • See LIBPACK_MAX_SWIOMA_STR_LEN for more information
<i>PkgNameLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Name String in Bytes.
<i>PkgName</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Name in ASCII • See LIBPACK_MAX_SWIOMA_STR_LEN for more information
<i>PkgDescLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Description String in Bytes.
<i>PkgDescription</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Description in ASCII • See LIBPACK_MAX_SWIOMA_STR_LEN for more information

<i>DateLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Package Description String in Bytes.
<i>Date</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Package Description in ASCII • See LIBPACK_MAX_SWIOMA_STR_LEN for more information
<i>TimeLength</i>	<ul style="list-style-type: none"> • 2 byte parameter indicating Length of Time String in Bytes.
<i>Time</i>	<ul style="list-style-type: none"> • Variable length parameter indicating Time String in ASCII • See LIBPACK_MAX_SWIOMA_STR_LEN for more information
<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
<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
<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

8.371.2 Field Documentation

8.371.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

8.371.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

8.371.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

8.371.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

- 8.371.2.5 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::PkgName[255]
- 8.371.2.6 uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::PkgNameLength
- 8.371.2.7 uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::RetryCount
- 8.371.2.8 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::SessionState
- 8.371.2.9 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::SessionType
- 8.371.2.10 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Severity
- 8.371.2.11 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Source[255]
- 8.371.2.12 uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::SourceLength
- 8.371.2.13 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Status
- 8.371.2.14 uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Time[255]
- 8.371.2.15 uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::TimeLength
- 8.371.2.16 uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

8.372 unpack_swima_SLQSOMADMGetSettings_t Struct Reference

Data Fields

- uint32_t [OMADMEEnabled](#)
- uint8_t [FOTAdownload](#)
- uint8_t [FOTAUpdate](#)
- uint8_t [Autosdm](#)
- uint8_t [FwAutoCheck](#)

8.372.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

Parameters

<i>OMADM-Enabled[OUT]</i>	<ul style="list-style-type: none">• 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• function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.
---------------------------	--

<i>FOTAdownload[OUT]</i>	<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” function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
<i>FOTAUpdate[OUT]</i>	<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 function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
<i>Autosdm[OUT]</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 function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.
<i>FwAutoCheck[OUT]</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 function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.

8.372.2 Field Documentation

8.372.2.1 uint8_t unpack_swima_SLQSOMADMGetSettings_t::Autosdm

8.372.2.2 uint8_t unpack_swima_SLQSOMADMGetSettings_t::FOTAdownload

8.372.2.3 uint8_t unpack_swima_SLQSOMADMGetSettings_t::FOTAUpdate

8.372.2.4 uint8_t unpack_swima_SLQSOMADMGetSettings_t::FwAutoCheck

8.372.2.5 uint32_t unpack_swima_SLQSOMADMGetSettings_t::OMADMEabled

8.373 unpack_swima_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [FwAvailability](#)

8.373.1 Detailed Description

Structure that contains the responses for OMA start session command

Parameters

<i>pFwAvailability</i> [OUT]	<ul style="list-style-type: none"> • OMA-DM CHECK FW Available <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
---------------------------------	---

8.373.2 Field Documentation

8.373.2.1 uint32_t unpack_swima_SLQSOMADMStartSession_t::FwAvailability

8.374 unpack_uim_ChangePin_t Struct Reference

Data Fields

- uim_remainingRetries * [pRemainingRetries](#)
- uim_encryptedPIN1 * [pEncryptedPIN1](#)
- uint32_t * [pIndicationToken](#)
- uint16_t [Tlvresult](#)

8.374.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

Parameters

<i>pRemainingRetries</i> (optional)	<ul style="list-style-type: none"> • See uim_remainingRetries for more information.
<i>pEncryptedPIN1</i> (optional)	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information.
<i>pIndicationToken</i> (optional)	<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. • 0xFFFFFFFF, if unavailable

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.374.2 Field Documentation

8.374.2.1 `uim_encryptedPIN1*` `unpack_uim_ChangePin_t::pEncryptedPIN1`

8.374.2.2 `uint32_t*` `unpack_uim_ChangePin_t::pIndicationToken`

8.374.2.3 `uim_remainingRetries*` `unpack_uim_ChangePin_t::pRemainingRetries`

8.374.2.4 `uint16_t` `unpack_uim_ChangePin_t::Tlvresult`

8.375 `unpack_uim_GetCardStatus_t` Struct Reference

Data Fields

- `uim_cardStatus` * `pCardStatus`
- `uim_hotSwapStatus` * `pHotSwapStatus`
- `uint16_t` `Tlvresult`

8.375.1 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 <code>uim_cardStatus</code> for more information.
<i>pHotSwap- Status(optional)</i>	<ul style="list-style-type: none"> • See <code>uim_hotSwapStatus</code> for more information.

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.375.2 Field Documentation

8.375.2.1 `uim_cardStatus*` `unpack_uim_GetCardStatus_t::pCardStatus`

8.375.2.2 `uim_hotSwapStatus*` `unpack_uim_GetCardStatus_t::pHotSwapStatus`

8.375.2.3 `uint16_t` `unpack_uim_GetCardStatus_t::Tlvresult`

8.376 `unpack_uim_ReadTransparent_t` Struct Reference

Data Fields

- `uim_cardResult` * `pCardResult`
- `uim_readResult` * `pReadResult`
- `uint32_t` * `pIndicationToken`

- uint8_t * [pEncryptedData](#)
- uint16_t [Tlvresult](#)

8.376.1 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 cardResult for more information.
<i>pReadResult</i>	<ul style="list-style-type: none"> • See readResult 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.
<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.

Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

8.376.2 Field Documentation

8.376.2.1 uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult

8.376.2.2 uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData

8.376.2.3 uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken

8.376.2.4 uim_readResult* unpack_uim_ReadTransparent_t::pReadResult

8.376.2.5 uint16_t unpack_uim_ReadTransparent_t::Tlvresult

8.377 unpack_uim_SetPinProtection_t Struct Reference

Data Fields

- [uim_remainingRetries](#) * [pRemainingRetries](#)
- [uim_encryptedPIN1](#) * [pEncryptedPIN1](#)
- uint32_t * [pIndicationToken](#)
- uint16_t [Tlvresult](#)

8.377.1 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.
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information.
<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. • 0xFFFFFFFF, if unavailable

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.377.2 Field Documentation

8.377.2.1 [uim_encryptedPIN1](#)* [unpack_uim_SetPinProtection_t::pEncryptedPIN1](#)

8.377.2.2 [uint32_t](#)* [unpack_uim_SetPinProtection_t::pIndicationToken](#)

8.377.2.3 [uim_remainingRetries](#)* [unpack_uim_SetPinProtection_t::pRemainingRetries](#)

8.377.2.4 [uint16_t](#) [unpack_uim_SetPinProtection_t::Tlvresult](#)

8.378 [unpack_uim_SetUimSlotStatusChangeCallback_ind_t](#) Struct Reference

Data Fields

- [slots_t slotsstatusChange](#)
- [uint8_t bNumberOfPhySlots](#)

8.378.1 Detailed Description

Structure consist of card status params

Parameters

<i>slotsstatusChange</i>	<ul style="list-style-type: none"> • See slot_t for more information
<i>bNumberOfPhySlots</i>	<ul style="list-style-type: none"> • Number of Physical Slot(s)

8.378.2 Field Documentation

8.378.2.1 [uint8_t](#) [unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots](#)

8.378.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

8.379 unpack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t [eventMask](#)

8.379.1 Detailed Description

Parameters

<i>eventMask</i>	- bit 1 - card status
------------------	-----------------------

8.379.2 Field Documentation

8.379.2.1 uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask

8.380 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

Data Fields

- uint8_t * [pNumberOfPhySlot](#)
- slots_t * [pUimSlotsStatus](#)

8.380.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

Parameters

<i>pNumberOfPhySlot</i>	<ul style="list-style-type: none">• Number of sets of the Slot Status.
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none">• Slots Status See slots_t for more information..

8.380.2 Field Documentation

8.380.2.1 uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot

8.380.2.2 slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus

8.381 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

Data Fields

- uim_cardStatus * [pCardStatus](#)

8.381.1 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.
--------------------	--

8.381.2 Field Documentation

8.381.2.1 `uim_cardStatus*` `unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus`

8.382 unpack_uim_UnblockPin_t Struct Reference

Data Fields

- `uim_remainingRetries` * `pRemainingRetries`
- `uim_encryptedPIN1` * `pEncryptedPIN1`
- `uint32_t` * `pIndicationToken`
- `uint16_t` `Tlvresult`

8.382.1 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.
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> See uim_encryptedPIN1 for more information.
<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. 0xFFFFFFFF, if unavailable

8.382.2 Field Documentation

8.382.2.1 `uim_encryptedPIN1*` `unpack_uim_UnblockPin_t::pEncryptedPIN1`8.382.2.2 `uint32_t*` `unpack_uim_UnblockPin_t::pIndicationToken`8.382.2.3 `uim_remainingRetries*` `unpack_uim_UnblockPin_t::pRemainingRetries`8.382.2.4 `uint16_t` `unpack_uim_UnblockPin_t::Tlvresult`

8.383 unpack_uim_VerifyPin_t Struct Reference

Data Fields

- [uim_remainingRetries](#) * [pRemainingRetries](#)
- [uim_encryptedPIN1](#) * [pEncryptedPIN1](#)
- [uint32_t](#) * [pIndicationToken](#)
- [uint16_t](#) [Tlvresult](#)

8.383.1 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.
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See uim_encryptedPIN1 for more information.
<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. • 0xFFFFFFFF, if unavailable

Note

Using NULL for the pointers would make sure that the parameter is not returned.

8.383.2 Field Documentation

8.383.2.1 [uim_encryptedPIN1](#)* [unpack_uim_VerifyPin_t::pEncryptedPIN1](#)

8.383.2.2 [uint32_t](#)* [unpack_uim_VerifyPin_t::pIndicationToken](#)

8.383.2.3 [uim_remainingRetries](#)* [unpack_uim_VerifyPin_t::pRemainingRetries](#)

8.383.2.4 [uint16_t](#) [unpack_uim_VerifyPin_t::Tlvresult](#)

8.384 [unpack_wds_GetConnectionRate_t](#) Struct Reference

Data Fields

- [uint32_t](#) [currentChannelTXRate](#)
- [uint32_t](#) [currentChannelRXRate](#)
- [uint32_t](#) [maxChannelTXRate](#)
- [uint32_t](#) [maxChannelRXRate](#)

8.384.1 Detailed Description

Parameters

<i>currentChannel-TXRate</i>	Instantaneous channel Tx rate
<i>currentChannel-RXRate</i>	Instantaneous channel Rx rate
<i>maxChannelTX-Rate</i>	Maximum Tx rate
<i>maxChannelRX-Rate</i>	Maximum Rx rate

8.384.2 Field Documentation

8.384.2.1 `uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate`

8.384.2.2 `uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate`

8.384.2.3 `uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate`

8.384.2.4 `uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate`

8.385 unpack_wds_GetDefaultProfile_t Struct Reference

Data Fields

- `uint32_t pdptype`
- `uint32_t ipaddr`
- `uint32_t pridns`
- `uint32_t secdns`
- `uint16_t ipaddrv6`
- `uint16_t pridnsv6`
- `uint16_t secdnsv6`
- `uint32_t auth`
- `uint8_t namesize`
- `int8_t name` [255]
- `uint8_t apnsize`
- `int8_t apnname` [255]
- `uint8_t usersize`
- `int8_t username` [255]

8.385.1 Detailed Description

Parameters

<i>pdptype</i>	pdp type
<i>ipaddr</i>	ip address
<i>pridns</i>	primary dns
<i>secdns</i>	secondry dns
<i>ipaddrv6</i>	ip address v6
<i>pridnsv6</i>	primary dns v6
<i>secdnsv6</i>	secondry dns v6
<i>namesize</i>	profile name size
<i>name</i>	profile name
<i>apnsize</i>	apn size
<i>apnname</i>	apn name
<i>usersize</i>	username size
<i>username</i>	username

8.385.2 Field Documentation

8.385.2.1 `int8_t unpack_wds_GetDefaultProfile_t::apnname[255]`

8.385.2.2 `uint8_t unpack_wds_GetDefaultProfile_t::apnsize`

8.385.2.3 `uint32_t unpack_wds_GetDefaultProfile_t::auth`

8.385.2.4 `uint32_t unpack_wds_GetDefaultProfile_t::ipaddr`

8.385.2.5 `uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6`

8.385.2.6 `int8_t unpack_wds_GetDefaultProfile_t::name[255]`

8.385.2.7 `uint8_t unpack_wds_GetDefaultProfile_t::namesize`

8.385.2.8 `uint32_t unpack_wds_GetDefaultProfile_t::pdptype`

8.385.2.9 `uint32_t unpack_wds_GetDefaultProfile_t::pridns`

8.385.2.10 `uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6`

8.385.2.11 `uint32_t unpack_wds_GetDefaultProfile_t::secdns`

8.385.2.12 `uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6`

8.385.2.13 `int8_t unpack_wds_GetDefaultProfile_t::username[255]`

8.385.2.14 `uint8_t unpack_wds_GetDefaultProfile_t::usersize`

8.386 `unpack_wds_GetDefaultProfileNum_t` Struct Reference

Data Fields

- `uint8_t` [index](#)

8.386.1 Detailed Description

Parameters

<i>index</i>	profile index
--------------	---------------

8.386.2 Field Documentation

8.386.2.1 `uint8_t unpack_wds_GetDefaultProfileNum_t::index`

8.387 `unpack_wds_GetDormancyState_t` Struct Reference

Data Fields

- `uint32_t dormancyState`

8.387.1 Detailed Description

Parameters

<i>dormancyState</i>	dormancy status
----------------------	-----------------

8.387.2 Field Documentation

8.387.2.1 `uint32_t unpack_wds_GetDormancyState_t::dormancyState`

8.388 `unpack_wds_GetLastMobileIPError_t` Struct Reference

Data Fields

- `uint32_t error`

8.388.1 Detailed Description

Parameters

<i>error</i>	last mip status 0-success >0- error code
--------------	--

8.388.2 Field Documentation

8.388.2.1 `uint32_t unpack_wds_GetLastMobileIPError_t::error`

8.389 `unpack_wds_GetMobileIP_t` Struct Reference

Data Fields

- `uint32_t mipMode`

8.389.1 Detailed Description

Parameters

<i>mipMode</i>	mobile IP mode
----------------	----------------

8.389.2 Field Documentation

8.389.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.390 unpack_wds_GetMobileIPProfile_t Struct Reference

Data Fields

- uint8_t [enabled](#)
- uint32_t [address](#)
- uint32_t [primaryHA](#)
- uint32_t [secondaryHA](#)
- uint8_t [revTunneling](#)
- uint8_t [naiSize](#)
- int8_t [NAI](#) [255]
- uint32_t [HASPI](#)
- uint32_t [AAASPI](#)
- uint32_t [HASState](#)
- uint32_t [AAASState](#)

8.390.1 Detailed Description

Parameters

<i>enabled</i>	mobile ip profile state enabled/disabled
<i>address</i>	mobile ip profile home address
<i>primaryHA</i>	mobile ip profile home agent primary
<i>secondaryHA</i>	mobile ip profile secondary home agent address
<i>revTunneling</i>	mobile ip profile rev tunneling
<i>naiSize</i>	mobile ip profile NAI size
<i>NAI</i>	NAI string in ASCII text.
<i>HASPI</i>	HA security parameter index.
<i>AAASPI</i>	AAA server security parameter index.
<i>HASState</i>	Mobile IP Profile HA Key State
<i>AAASState</i>	Mobile IP Profile AAA Key State

8.390.2 Field Documentation

8.390.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

8.390.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAASState

8.390.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

8.390.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

8.390.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

8.390.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HASState

8.390.2.7 `int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]`

8.390.2.8 `uint8_t unpack_wds_GetMobileIPProfile_t::naiSize`

8.390.2.9 `uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA`

8.390.2.10 `uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling`

8.390.2.11 `uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA`

8.391 `unpack_wds_GetPacketStatus_t` Struct Reference

Data Fields

- `uint32_t tXPacketSuccesses`
- `uint32_t rXPacketSuccesses`
- `uint32_t tXPacketErrors`
- `uint32_t rXPacketErrors`
- `uint32_t tXPacketOverflows`
- `uint32_t rXPacketOverflows`
- `uint64_t tXOkBytesCount`
- `uint64_t rXOkBytesCount`
- `uint64_t tXOKBytesLastCall`
- `uint64_t rXOKBytesLastCall`
- `uint32_t tXDroppedCount`
- `uint32_t rXDroppedCount`

8.391.1 Detailed Description

Parameters

<i>tXPacket-Successes</i>	Tx Packets OK
<i>rXPacket-Successes</i>	Rx Packets OK
<i>tXPacketErrors</i>	Tx Packet Errors
<i>rXPacketErrors</i>	Rx Packet Errors
<i>tXPacket-Overflows</i>	Tx Overflows
<i>rXPacket-Overflows</i>	Rx Overflows
<i>tXOkBytesCount</i>	Tx Bytes OK
<i>rXOkBytesCount</i>	Rx Bytes OK
<i>tXOKBytesLast-Call</i>	Last call Tx Bytes OK
<i>rXOKBytesLast-Call</i>	Last call Rx Bytes OK
<i>tXDroppedCount</i>	Tx Packets Dropped
<i>rXDroppedCount</i>	Rx Packets Dropped

8.391.2 Field Documentation

8.391.2.1 `uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount`

- 8.391.2.2 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount
- 8.391.2.3 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesLastCall
- 8.391.2.4 uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors
- 8.391.2.5 uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows
- 8.391.2.6 uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses
- 8.391.2.7 uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount
- 8.391.2.8 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount
- 8.391.2.9 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesLastCall
- 8.391.2.10 uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors
- 8.391.2.11 uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows
- 8.391.2.12 uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses

8.392 unpack_wds_GetSessionDuration_t Struct Reference

Data Fields

- uint64_t [callDuration](#)

8.392.1 Detailed Description

Parameters

<i>callDuration</i>	call duration in milliseconds
---------------------	-------------------------------

8.392.2 Field Documentation

- 8.392.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

8.393 unpack_wds_GetSessionState_t Struct Reference

Data Fields

- uint32_t [connectionStatus](#)

8.393.1 Detailed Description

Parameters

<i>connection-Status</i>	state of the current packet data session
--------------------------	--

8.393.2 Field Documentation

8.393.2.1 `uint32_t unpack_wds_GetSessionState_t::connectionStatus`

8.394 `unpack_wds_RMSetTransferStatistics_t` Struct Reference

8.395 `unpack_wds_SetMobileIPProfile_t` Struct Reference

8.396 `unpack_wds_SLQSCreateProfile_t` Struct Reference

Data Fields

- [PackCreateProfileOut](#) * [pCreateProfileOut](#)
- `uint8_t` * [pProfileID](#)
- `uint16_t` [Tlvresult](#)

8.396.1 Detailed Description

Parameters

<i>profile</i>	type
<i>profile</i>	index
<i>extended</i>	error

8.396.2 Field Documentation

8.396.2.1 `PackCreateProfileOut*` `unpack_wds_SLQSCreateProfile_t::pCreateProfileOut`

8.396.2.2 `uint8_t*` `unpack_wds_SLQSCreateProfile_t::pProfileID`

8.396.2.3 `uint16_t` `unpack_wds_SLQSCreateProfile_t::Tlvresult`

8.397 `unpack_wds_SLQSDeleteProfile_t` Struct Reference

Data Fields

- `uint16_t` [extendedErrorCode](#)

8.397.1 Detailed Description

Parameters

<i>extendedError-Code</i>	extended error code
---------------------------	---------------------

8.397.2 Field Documentation

8.397.2.1 `uint16_t` `unpack_wds_SLQSDeleteProfile_t::extendedErrorCode`

8.398 unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference

Data Fields

- uint16_t [profileList](#) [5]
- uint8_t [defaultPDNEnabled](#)
- uint8_t [_3gppRelease](#)
- uint16_t [LTEAttachProfileList](#) [24]
- uint16_t [LTEAttachProfileListLen](#)

8.398.1 Detailed Description

Parameters

	<i>profileList</i>	Profile List
out	<i>defaultPDN-Enabled</i>	<ul style="list-style-type: none"> • 0 - disabled • 1 - enabled
out	<i>_3gppRelease</i>	3GPP release <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)
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.
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

8.398.2 Field Documentation

8.398.2.1 `uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease`

8.398.2.2 `uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled`

8.398.2.3 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]`

8.398.2.4 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen`

8.398.2.5 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]`

8.399 `unpack_wds_SLQSGetCurrDataSystemStat_t` Struct Reference

Data Fields

- `uint8_t prefNetwork`
- `uint8_t networkInfoLen`
- `currNetworkInfo currNetworkInfo` [255]

8.399.1 Detailed Description

Parameters

<i>prefNetwork</i>	preferred network
<i>networkInfoLen</i>	number of set of <code>currNetworkInfo</code> elements
<i>currNetworkInfo</i>	current network infomation.

8.399.2 Field Documentation

8.399.2.1 `currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]`

8.399.2.2 `uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen`

8.399.2.3 `uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork`

8.400 `unpack_wds_SLQSGetDataBearerTechnology_t` Struct Reference

Data Fields

- `uint8_t dataBearerMask`
- `qmiWSDDataBearerTechnology curDataBearerTechnology`
- `qmiWSDDataBearerTechnology lastCallDataBearerTechnology`

8.400.1 Detailed Description

Parameters

<i>dataBearerMask</i>	bit mask indicates bearer info is for current and/or last call
<i>curDataBearer-Technology</i>	current data bearer technology value
<i>lastCallData-Bearer-Technology</i>	last call data bearer technology value

8.400.2 Field Documentation

8.400.2.1 qmiWSDDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology

8.400.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

8.400.2.3 qmiWSDDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology

8.401 unpack_wds_SLQSGetDUNCallInfo_t Struct Reference

Data Fields

- [connectionStatus](#) connectionStatus
- uint16_t [callEndReason](#)
- uint64_t [txOKBytesCount](#)
- uint64_t [rxOKBytesCount](#)
- uint8_t [dormancyStatus](#)
- uint8_t [dataBearerTech](#)
- [dunchannelRate](#) channelRate
- uint64_t [lastCallTXOKBytesCnt](#)
- uint64_t [lastCallRXOKBytesCnt](#)
- uint64_t [mdmCallDurationActive](#)
- uint8_t [lastCallDataBearerTech](#)

8.401.1 Detailed Description

Parameters

connectionStatus	Connection Status
callEndReason	Last Modem Call End Reason
txOKBytesCount	Tx Bytes OK
rxOKBytesCount	Rx Bytes OK
dormancyStatus	Dormancy Status
dataBearerTech	data bearer technology
channelRate	data Channel Rate
lastCallTXOKBytesCnt	Last Call Tx Bytes OK
lastCallRXOKBytesCnt	Last Call Rx Bytes OK
mdmCallDurationActive	Call active duration
lastCallDataBearerTech	Last Call Data Bearer Technology

8.401.2 Field Documentation

8.401.2.1 uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason

8.401.2.2 dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate

8.401.2.3 connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus

- 8.401.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`
- 8.401.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`
- 8.401.2.6 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech`
- 8.401.2.7 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt`
- 8.401.2.8 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt`
- 8.401.2.9 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive`
- 8.401.2.10 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount`
- 8.401.2.11 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount`

8.402 `unpack_wds_SLQSGetProfileSettings_t` Struct Reference

Data Fields

- [UnPackGetProfileSettingOut](#) * `pProfileSettings`
- `uint8_t` [ProfileType](#)
- `uint16_t` [Tlvresult](#)

8.402.1 Field Documentation

- 8.402.1.1 `UnPackGetProfileSettingOut`* `unpack_wds_SLQSGetProfileSettings_t::pProfileSettings`
- 8.402.1.2 `uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType`
- 8.402.1.3 `uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult`

8.403 `unpack_wds_SLQSGetRuntimeSettings_t` Struct Reference

Data Fields

- `uint32_t` [IPv4](#)
- `uint8_t` [ProfileName](#) [128]
- `uint32_t` [PDPTType](#)
- `uint8_t` [APNName](#) [128]
- `uint32_t` [PrimaryDNSV4](#)
- `uint32_t` [SecondaryDNSV4](#)
- [LibPackUMTSQoS](#) `UMTSGrantedQoS`
- `struct` [wds_GPRSQoS](#) `GPRSGrantedQoS`
- `uint8_t` [Username](#) [128]
- `uint32_t` [Authentication](#)
- `struct` [wds_ProfileIdentifier](#) `ProfileID`
- `uint32_t` [GWAddressV4](#)
- `uint32_t` [SubnetMaskV4](#)
- `uint8_t` [PCSCFAddrPCO](#)
- `struct`
[wds_PCSCFIPv4ServerAddressList](#) `ServerAddrList`
- `struct` [wds_PCSCFFQDNAddressList](#) `PCSCFFQDNAddrList`

- uint16_t [PrimaryDNSV6](#) [8]
- uint16_t [SecondaryDNSV6](#) [8]
- uint32_t [Mtu](#)
- struct [wds_DomainNameList](#) [DomainList](#)
- uint8_t [IPFamilyPreference](#)
- uint8_t [IMCNflag](#)
- uint16_t [Technology](#)
- struct [wds_IPV6AddressInfo](#) [IPv6AddrInfo](#)
- struct [wds_IPV6GWAddressInfo](#) [IPv6GWAddrInfo](#)

8.403.1 Detailed Description

Parameters

<i>IPv4</i>	ipv4 address
<i>ProfileName</i>	profile name
<i>PDPTType</i>	PDP type
<i>APNName</i>	APN name
<i>PrimaryDNSV4</i>	
<i>SecondaryDNS-V4</i>	
<i>UMTSGranted-QoS</i>	UMTS Granted QoS
<i>GPRSGranted-QoS</i>	GPRS Granted QoS
<i>Username</i>	
<i>Authentication</i>	
<i>ProfileID</i>	
<i>GWAddressV4</i>	Gateway IPv4
<i>SubnetMaskV4</i>	Subnet mask IPV4
<i>PCSCFAddrPC-O</i>	
<i>PrimaryDNSV6</i>	Primary DNS IPV6
<i>SecondaryDNS-V6</i>	Secondary DNS IPV6
<i>UMTSGranted-QoS</i>	UMTS Granted QoS
<i>SecondaryDNS-V4</i>	
<i>Mtu</i>	Maximum Transfer Unit
<i>DomainList</i>	
<i>IPFamily-Preference</i>	

8.403.2 Field Documentation

8.403.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

8.403.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

8.403.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.403.2.4 struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

8.403.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

- 8.403.2.6 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag`
- 8.403.2.7 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference`
- 8.403.2.8 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4`
- 8.403.2.9 `struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6AddrInfo`
- 8.403.2.10 `struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6GWAddrInfo`
- 8.403.2.11 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu`
- 8.403.2.12 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO`
- 8.403.2.13 `struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList`
- 8.403.2.14 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPTType`
- 8.403.2.15 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4`
- 8.403.2.16 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]`
- 8.403.2.17 `struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID`
- 8.403.2.18 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]`
- 8.403.2.19 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4`
- 8.403.2.20 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]`
- 8.403.2.21 `struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList`
- 8.403.2.22 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4`
- 8.403.2.23 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology`
- 8.403.2.24 `LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS`
- 8.403.2.25 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]`

8.404 `unpack_wds_SLQSModifyProfile_t` Struct Reference

Data Fields

- `uint16_t * pExtErrorCode`

8.404.1 Detailed Description

Parameters

<i>extended</i>	error
-----------------	-------

8.404.2 Field Documentation

8.404.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

8.405 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.405.1 Detailed Description

Parameters

<i>Tlvresult</i>	unpack result
------------------	---------------

8.405.2 Field Documentation

8.405.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

8.406 unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference

Data Fields

- uint8_t [conn_status](#)
- uint8_t [reconfigReqd](#)
- uint16_t [sessionEndReason](#)
- uint16_t [verboseSessnEndReasonType](#)
- uint16_t [verboseSessnEndReason](#)
- uint8_t [ipFamily](#)
- uint16_t [techName](#)
- uint8_t [bearerID](#)

8.406.1 Detailed Description

Parameters

<i>conn_status</i>	connection status
<i>reconfigReqd</i>	Indicates whether the network interface on the host needs to be reconfigured.
<i>sessionEndReason</i>	Call End Reason
<i>verboseSessnEndReasonType</i>	Verbose call end reason type
<i>verboseSessnEndReason</i>	Reason the call ended (verbose)
<i>ipFamily</i>	IP family of the packet data connection.
<i>techName</i>	Technology name of the packet data connection.
<i>bearerID</i>	<ul style="list-style-type: none">• bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection.• Valid Values - 0 to 16• 0xFF - Invalid value.

8.406.2 Field Documentation

- 8.406.2.1 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID`
- 8.406.2.2 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status`
- 8.406.2.3 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily`
- 8.406.2.4 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd`
- 8.406.2.5 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason`
- 8.406.2.6 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName`
- 8.406.2.7 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason`
- 8.406.2.8 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType`

8.407 `unpack_wds_SLQSSetWdsEventCallback_ind_t` Struct Reference

Data Fields

- `uint8_t xferStatAvail`
- `uint64_t tx_bytes`
- `uint64_t rx_bytes`
- `uint64_t tx_pkts`
- `uint64_t rx_pkts`
- `uint8_t mipstatAvail`
- `uint32_t mipStatus`
- `uint8_t dBTechAvail`
- `uint32_t dBTechnology`
- `uint8_t dormancyStatAvail`
- `uint32_t dormancyStatus`
- `uint8_t currDBTechAvail`
- `uint32_t ratMask`
- `uint32_t soMask`
- `uint8_t dataSysStatAvail`
- `uint8_t prefNetwork`
- `uint8_t netInfoLen`
- `wds_currNetworkInfo currNWInfo` [255]

8.407.1 Detailed Description

Parameters

<i>xferStatAvail</i>	transfer statistic available
<i>tx_bytes</i>	transmit bytes
<i>rx_bytes</i>	received bytes
<i>tx_pkts</i>	transmit packets
<i>rx_pkts</i>	received packets

<i>mipstatAvail</i>	Mobile IP status available
<i>mipStatus</i>	Mobile IP status
<i>dBTechAvail</i>	Data Bearer technology available
<i>dBTechnology</i>	Data Bearer technology
<i>dormancyStatAvail</i>	Dormancy status available
<i>dormancyStatus</i>	Dormancy status
<i>currDBTechAvail</i>	Current Data Bearer technology available
<i>ratMask</i>	RAT mask to indicate type of technology
<i>soMask</i>	SO mask to indicate the service type
<i>dataSysStatAvail</i>	Data System Status available
<i>prefNetwork</i>	preferred network
<i>currNWInfo</i>	Current Network Info

8.407.2 Field Documentation

8.407.2.1 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail`

8.407.2.2 `wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]`

8.407.2.3 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail`

8.407.2.4 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail`

8.407.2.5 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology`

8.407.2.6 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail`

8.407.2.7 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus`

8.407.2.8 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail`

8.407.2.9 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus`

8.407.2.10 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen`

8.407.2.11 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork`

8.407.2.12 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask`

8.407.2.13 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes`

8.407.2.14 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts`

8.407.2.15 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask`

8.407.2.16 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes`

8.407.2.17 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts`

8.407.2.18 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail`

8.408 unpack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

- [wdsDhcpv4HwConfig](#) * [pHwConfig](#)
- [wdsDhcpv4OptionList](#) * [pRequestOptionList](#)

8.408.1 Detailed Description

Parameters

<i>pHwConfig</i>	pointer to HW Config structure
<i>pRequestOptionList</i>	pointer to Option List structure to be sent in DHCP request

8.408.2 Field Documentation

8.408.2.1 [wdsDhcpv4HwConfig](#)* [unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig](#)

8.408.2.2 [wdsDhcpv4OptionList](#)* [unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList](#)

8.409 [unpack_wds_SLQSStartDataSession_t](#) Struct Reference

Data Fields

- [uint32_t](#) * [psid](#)
- [uint32_t](#) * [pFailureReason](#)
- [uint32_t](#) * [pVerboseFailReasonType](#)
- [uint32_t](#) * [pVerboseFailureReason](#)

8.409.1 Detailed Description

Parameters

<i>psid</i>	<ul style="list-style-type: none"> • Assigned session ID when starting a data session
<i>pFailureReason</i>	<ul style="list-style-type: none"> • Reason data session failed to be established • See qaGobiApiTableCallEndReasons.h for Call End Reason
<i>pVerboseFailReasonType</i>	<ul style="list-style-type: none"> • Parameter describing type of verbose failure reason • See qaGobiApiTableCallEndReasons.h for Call End Reason Type
<i>pVerboseFailureReason</i>	<ul style="list-style-type: none"> • Verbose reason explaining why call failed. Depends on verbFailReasonType parameter • See qaGobiApiTableCallEndReasons.h for Call End Reason

8.409.2 Field Documentation

8.409.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

8.409.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

8.409.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

8.409.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

8.410 unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

- uint8_t [contextId](#)
- uint8_t [bearerId](#)
- int8_t [apnName](#) [100]
- uint32_t [ipv4Address](#)
- uint32_t [ipv4GWAddress](#)
- uint32_t [prDNSIPv4Address](#)
- uint32_t [seDNSIPv4Address](#)
- struct [ipv6AddressInfo](#) [ipv6Address](#)
- struct [ipv6AddressInfo](#) [ipv6GWAddress](#)
- uint16_t [prDNSIPv6Address](#) [8]
- uint16_t [seDNSIPv6Address](#) [8]
- uint32_t [prPCSCFIPv4Address](#)
- uint32_t [sePCSCFIPv4Address](#)
- uint16_t [prPCSCFIPv6Address](#) [8]
- uint16_t [sePCSCFIPv6Address](#) [8]

8.410.1 Detailed Description

Parameters

<i>contextId</i>	Context Identifier
<i>bearerId</i>	Bearer Identity
<i>apnName</i>	APN name associated with the context id
<i>ipv4Address</i>	IPv4 Address
<i>ipv4GWAddress</i>	IPv4 Gateway Address
<i>prDNSIPv4-Address</i>	Primary DNS IPv4 Address
<i>seDNSIPv4-Address</i>	Secondary DNS IPv4 Address
<i>ipv6Address</i>	IPv6 Address
<i>ipv6GWAddress</i>	IPv6 Gateway Address
<i>prDNSIPv6-Address</i>	Primary IPv6 DNS Address
<i>seDNSIPv6-Address</i>	Secondary IPv6 DNS Address
<i>prPCSCFIPv4-Address</i>	Primary PCSCF IPv4 Address
<i>sePCSCFIPv4-Address</i>	Secondary PCSCF IPv4 Address
<i>prPCSCFIPv6-Address</i>	Primary PCSCF IPv6 Address
<i>sePCSCFIPv6-Address</i>	Secondary PCSCF IPv6 Address

8.410.2 Field Documentation

8.410.2.1 `int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]`

8.410.2.2 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId`

8.410.2.3 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.410.2.4 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address`

8.410.2.5 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress`

8.410.2.6 `struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address`

8.410.2.7 `struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress`

8.410.2.8 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address`

8.410.2.9 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`

8.410.2.10 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`

8.410.2.11 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`

8.410.2.12 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`

8.410.2.13 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`

8.410.2.14 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`

8.410.2.15 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

8.411 UnPackGetProfileSettingOut Struct Reference

Data Fields

- [UnpackQmiProfileInfo curProfile](#)
- `uint16_t *` [pExtErrCode](#)

8.411.1 Field Documentation

8.411.1.1 `UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile`

8.411.1.2 `uint16_t*` `UnPackGetProfileSettingOut::pExtErrCode`

8.412 unpackWdsProfileParam Union Reference

Data Fields

- [LibpackProfile3GPP SlqsProfile3GPP](#)
- [LibpackProfile3GPP2 SlqsProfile3GPP2](#)

8.412.1 Field Documentation

8.412.1.1 `LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP`

8.412.1.2 `LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2`

8.413 wds_currNetworkInfo Struct Reference

Data Fields

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

8.413.1 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"> • Radio Access Technology (RAT) mask to indicate the type of technology. • Values: <ul style="list-style-type: none"> – 0 - Don't Care – 0x8000 - NULL Bearer • CDMA RAT mask values: <ul style="list-style-type: none"> – 0x01 - CDMA_1x – 0x02 - EVDO_REV0 – 0x04 - EVDO_REVA – 0x08 - EVDO_REVB – 0x10 - EHRPD – 0x20 - FMC • UMTS RAT mask values: <ul style="list-style-type: none"> – 0x01 - WCDMA – 0x02 - GPRS – 0x04 - HSDPA – 0x08 - HSUPA – 0x10 - EDGE – 0x20 - LTE – 0x40 - HSDPA+ – 0x80 - DC_HSDPA+ – 0x100 - 64_QAM – 0x200 - TDSCDMA

<i>SOMask</i>	<ul style="list-style-type: none"> • Service Option (SO) mask to indicate the service option or type of application. • Values: <ul style="list-style-type: none"> – 0 - Don't Care • CDMA 1x SO mask values: <ul style="list-style-type: none"> – 0x01 - CDMA_1X_IS95 – 0x02 - CDMA_1X_IS2000 – 0x04 - CDMA_1X_IS2000_REL_A • CDMA EV-DO Rev 0 SO mask values: <ul style="list-style-type: none"> – 0x01 - DPA • CDMA EV-DO Rev A SO mask values: <ul style="list-style-type: none"> – 0x01 - DPA – 0x02 - MFPA – 0x04 - EMPA – 0x08 - EMPA_EHRPD • CDMA EV-DO Rev B SO mask values: <ul style="list-style-type: none"> – 0x01 - DPA – 0x02 - MFPA – 0x04 - EMPA – 0x08 - EMPA_EHRPD – 0x10 - MMPA – 0x20 - MMPA_EHRPD
---------------	--

8.413.2 Field Documentation

8.413.2.1 `uint8_t wds_currNetworkInfo::NetworkType`

8.413.2.2 `uint32_t wds_currNetworkInfo::RATMask`

8.413.2.3 `uint32_t wds_currNetworkInfo::SOMask`

8.414 wds_Domain Struct Reference

Data Fields

- `uint16_t domainLen`
- `uint8_t domainName` [256]

8.414.1 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)

8.414.2 Field Documentation

8.414.2.1 uint16_t wds_Domain::domainLen

8.414.2.2 uint8_t wds_Domain::domainName[256]

8.415 wds_DomainNameList Struct Reference

Data Fields

- uint8_t [numInstances](#)
- struct [wds_Domain domain](#) [10]

8.415.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> Number of Domain name receieved
<i>domain</i>	<ul style="list-style-type: none"> Domain name information(Max 10 Domain names)

8.415.2 Field Documentation

8.415.2.1 struct wds_Domain wds_DomainNameList::domain[10]

8.415.2.2 uint8_t wds_DomainNameList::numInstances

8.416 wds_GPRSQoS Struct Reference

Data Fields

- uint32_t [precedenceClass](#)
- uint32_t [delayClass](#)
- uint32_t [reliabilityClass](#)
- uint32_t [peakThroughputClass](#)
- uint32_t [meanThroughputClass](#)

8.416.1 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

8.416.2 Field Documentation

8.416.2.1 uint32_t wds_GPRSQoS::delayClass

8.416.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

8.416.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

8.416.2.4 uint32_t wds_GPRSQoS::precedenceClass

8.416.2.5 uint32_t wds_GPRSQoS::reliabilityClass

8.417 wds_IPV6AddressInfo Struct Reference

Data Fields

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

8.417.1 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 <ul style="list-style-type: none"> – 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.
--------------------	---

8.417.2 Field Documentation

8.417.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

8.417.2.2 uint8_t wds_IPV6AddressInfo::IPv6PrefixLen

8.418 wds_IPV6GWAddressInfo Struct Reference

Data Fields

- uint8_t [gwV6PrefixLen](#)
- uint16_t [gwAddressV6](#) [8]

8.418.1 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.

8.418.2 Field Documentation

8.418.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

8.418.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

8.419 wds_PCSCFFQDNAddress Struct Reference

Data Fields

- uint16_t [fqdnLen](#)
- uint8_t [fqdnAddr](#) [256]

8.419.1 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)

8.419.2 Field Documentation

8.419.2.1 `uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]`8.419.2.2 `uint16_t wds_PCSCFFQDNAddress::fqdnLen`8.420 `wds_PCSCFFQDNAddressList` Struct Reference

Data Fields

- `uint8_t numInstances`
- struct `wds_PCSCFFQDNAddress pcsfFQDNAddress` [10]

8.420.1 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)

8.420.2 Field Documentation

8.420.2.1 `uint8_t wds_PCSCFFQDNAddressList::numInstances`8.420.2.2 `struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]`8.421 `wds_PCSCFIPv4ServerAddressList` Struct Reference

Data Fields

- `uint8_t numInstances`
- `uint32_t pcsfIPv4Addr` [64]

8.421.1 Detailed Description

This structure contains the PCSCFIPv4ServerAddressList Information

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • number of address following
<i>pcscfIPv4Addr</i>	<ul style="list-style-type: none"> • P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)

8.421.2 Field Documentation

8.421.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

8.421.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pcscfIPv4Addr[64]

8.422 wds_ProfileIdentifier Struct Reference

Data Fields

- uint8_t [profileType](#)
- uint8_t [profileIndex](#)

8.422.1 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

8.422.2 Field Documentation

8.422.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.422.2.2 uint8_t wds_ProfileIdentifier::profileType

8.423 wds_profileInfo Union Reference

Data Fields

- [LibPackprofile_3GPP](#) [SlqsProfile3GPP](#)
- [LibPackprofile_3GPP2](#) [SlqsProfile3GPP2](#)

8.423.1 Detailed Description

This union consist of profile_3GPP and profile_3GPP2 out of which one will be used to create profile.

8.423.2 Field Documentation

8.423.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

8.423.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

8.424 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](#)

8.424.1 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>	- Residual bit error ratio <ul style="list-style-type: none"> • 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>	- 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.

8.424.2 Field Documentation

8.424.2.1 `uint8_t wds_UMTSMInQoS::deliveryErrSDU`

8.424.2.2 `uint32_t wds_UMTSMInQoS::grntDownlinkBitrate`

- 8.424.2.3 `uint32_t wds_UMTSMInQoS::grntUplinkBitrate`
- 8.424.2.4 `uint32_t wds_UMTSMInQoS::maxDownlinkBitrate`
- 8.424.2.5 `uint32_t wds_UMTSMInQoS::maxSDUSize`
- 8.424.2.6 `uint32_t wds_UMTSMInQoS::maxUplinkBitrate`
- 8.424.2.7 `uint8_t wds_UMTSMInQoS::qosDeliveryOrder`
- 8.424.2.8 `uint8_t wds_UMTSMInQoS::resBerRatio`
- 8.424.2.9 `uint8_t wds_UMTSMInQoS::sduErrorRatio`
- 8.424.2.10 `uint8_t wds_UMTSMInQoS::trafficClass`
- 8.424.2.11 `uint32_t wds_UMTSMInQoS::trafficPriority`
- 8.424.2.12 `uint32_t wds_UMTSMInQoS::transferDelay`

8.425 wdsDhcpv4HwConfig Struct Reference

Data Fields

- `uint8_t hwType`
- `uint8_t chaddrLen`
- `uint8_t chaddr [16]`

8.425.1 Detailed Description

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

8.425.2 Field Documentation

- 8.425.2.1 `uint8_t wdsDhcpv4HwConfig::chaddr[16]`
- 8.425.2.2 `uint8_t wdsDhcpv4HwConfig::chaddrLen`
- 8.425.2.3 `uint8_t wdsDhcpv4HwConfig::hwType`

8.426 wdsDhcpv4Option Struct Reference

Data Fields

- [uint8_t optCode](#)
- [uint8_t optValLen](#)
- [uint8_t optVal](#) [255]

8.426.1 Detailed Description

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

8.426.2 Field Documentation

8.426.2.1 [uint8_t wdsDhcpv4Option::optCode](#)

8.426.2.2 [uint8_t wdsDhcpv4Option::optVal](#)[255]

8.426.2.3 [uint8_t wdsDhcpv4Option::optValLen](#)

8.427 wdsDhcpv4OptionList Struct Reference

Data Fields

- [uint8_t numOpt](#)
- [wdsDhcpv4Option * pOptList](#)

8.427.1 Detailed Description

Parameters

<i>numOpt</i>	number of options <ul style="list-style-type: none"> • 0 - 255
<i>pOptList</i>	pointer to list of DHCP Options

8.427.2 Field Documentation

8.427.2.1 [uint8_t wdsDhcpv4OptionList::numOpt](#)

8.427.2.2 [wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList](#)

8.428 wdsDhcpv4ProfileId Struct Reference

Data Fields

- uint8_t [profileType](#)
- uint8_t [profileId](#)

8.428.1 Detailed Description

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)

8.428.2 Field Documentation

8.428.2.1 uint8_t wdsDhcpv4ProfileId::profileId

8.428.2.2 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 common.h File Reference

Data Structures

- struct [pack_qmi_t](#)
- struct [unpack_qmi_t](#)

Macros

- #define [SDU_HDR_LEN](#) (3)
- #define [MINREQBKLEN](#) (2048)
- #define [MSGID_AND_LEN](#) (4)
- #define [MSGID_DONT_CARE](#) (0xffff)
- #define [UNUSEDPARAM](#)(x) (void)x
- #define [DEAULT_LOC_TIMEOUT_IN_SEC](#) 2
- #define [SDK_VALIDATE_INPUT_PACK_PARAM](#)(pCtx, pBuf, pLen)

Typedefs

- typedef void(* [logger](#))(uint8_t lvl, const char *buff)

Enumerations

- enum [eLOG_LEVEL](#) {
[eLOG_INFO](#),
[eLOG_DEBUG](#),
[eLOG_WARN](#),
[eLOG_FATAL](#) }
- enum [eTimeout](#) {
[eTIMEOUT_2_S](#) = 2000,
[eTIMEOUT_5_S](#) = 5000,
[eTIMEOUT_8_S](#) = 8000,
[eTIMEOUT_10_S](#) = 10000,
[eTIMEOUT_20_S](#) = 20000,
[eTIMEOUT_30_S](#) = 30000,
[eTIMEOUT_60_S](#) = 60000,
[eTIMEOUT_300_S](#) = 300000,
[eTIMEOUT_DEFAULT](#) = [eTIMEOUT_8_S](#) }
- enum [eQMI_SVC](#) {
[eCTL](#),
[eWDS](#),
[eDMS](#),
[eNAS](#) =3,
[eQOS](#),
[eSMS](#) =5,
[eUIM](#) =0x0B,
[eLOC](#) =0x10,
[eSWIOMA](#) =240,
[eSWILOC](#) =246 }
- enum [msgtype](#) {
[eREQ](#) =0,
[eRSP](#) =2,
[eIND](#) =4 }

Functions

- [uint16_t helper_get_xid](#) ([uint8_t](#) *qmi_resp)
- const char * [helper_get_resp_ctx](#) ([uint8_t](#) svc, [uint8_t](#) *pbuf, [uint32_t](#) len, [unpack_qmi_t](#) *pCtx)
- unsigned [unpack_result_code_only](#) ([uint8_t](#) *pMdmResp)
- int [helper_set_log_func](#) ([logger](#) func)
- void [libpack_log](#) ([uint8_t](#) lvl, const char *fmt,...)
- int [helper_set_log_lvl](#) ([uint8_t](#) lvl)
- void [fill_sdu_hdr](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf)
- void [fill_pack_ctx](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [uint8_t](#) svc, int timeout)
- char * [get_version](#) ()

Variables

- [logger](#) [glog](#)
- [uint8_t](#) [gloglvl](#)

9.2.1 Macro Definition Documentation

9.2.1.1 #define [DEFAULT_LOC_TIMEOUT_IN_SEC](#) 2

9.2.1.2 `#define MINREQBKLEN (2048)`

9.2.1.3 `#define MSGID_AND_LEN (4)`

9.2.1.4 `#define MSGID_DONT_CARE (0xffff)`

9.2.1.5 `#define SDK_VALIDATE_INPUT_PACK_PARAM(pCtx, pBuf, pLen)`

Value:

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
{ \
    libpack_log(eLOG_DEBUG, "[ pack] %s parameter NULL\n", __func__); \
    return eQCWWAN_ERR_INVALID_ARG; \
}
```

9.2.1.6 `#define SDU_HDR_LEN (3)`

9.2.1.7 `#define UNUSEDPARAM(x)(void)x`

9.2.2 Typedef Documentation

9.2.2.1 `typedef void(* logger)(uint8_t lvl, const char *buff)`

9.2.3 Enumeration Type Documentation

9.2.3.1 `enum eLOG_LEVEL`

log levels

Enumerator

eLOG_INFO
eLOG_DEBUG
eLOG_WARN
eLOG_FATAL

9.2.3.2 `enum eQMI_SVC`

qmi service

Enumerator

eCTL
eWDS
eDMS
eNAS
eQOS
eSMS
eUIM
eLOC
eSWIOMA
eSWILOC

9.2.3.3 enum eTimeout

eTimeout

Enumerator

eTIMEOUT_2_S
eTIMEOUT_5_S
eTIMEOUT_8_S
eTIMEOUT_10_S
eTIMEOUT_20_S
eTIMEOUT_30_S
eTIMEOUT_60_S
eTIMEOUT_300_S
eTIMEOUT_DEFAULT

9.2.3.4 enum msgtype

qmi message type

Enumerator

eREQ
eRSP
eIND

9.2.4 Function Documentation

9.2.4.1 void fill_pack_ctx (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t svc, int timeout)

9.2.4.2 void fill_sdu_hdr (pack_qmi_t * pCtx, uint8_t * pReqBuf)

9.2.4.3 char* get_version ()

Returns

version string

9.2.4.4 const char* helper_get_resp_ctx (uint8_t svc, uint8_t * pbuf, uint32_t len, unpack_qmi_t * pCtx)

extract msgid/xid/type from modem reply

Parameters

in	svc	qmi service
in	pbuf	qmi response/indication
in	len	response/indication length
out	pCtx	unpacked context

Returns

qmi message string

9.2.4.5 `uint16_t helper_get_xid (uint8_t * qmi_resp)`

9.2.4.6 `int helper_set_log_func (logger func)`

set log function

9.2.4.7 `int helper_set_log_lvl (uint8_t lvl)`

set log level

9.2.4.8 `void libpack_log (uint8_t lvl, const char * fmt, ...)`

9.2.4.9 `unsigned unpack_result_code_only (uint8_t * pMdmResp)`

common handler for unpacking response with TLV type 0x02 only

9.2.5 Variable Documentation

9.2.5.1 `logger glog`

9.2.5.2 `uint8_t gloglvl`

9.3 dms.h File Reference

Data Structures

- struct [unpack_dms_GetModelID_t](#)
- struct [unpack_dms_GetIMSI_t](#)
- struct [unpack_dms_GetFirmwareInfo_t](#)
- struct [unpack_dms_GetPower_t](#)
- struct [unpack_dms_GetSerialNumbers_t](#)
- struct [unpack_dms_GetHardwareRevision_t](#)
- struct [unpack_dms_SLQSGetBandCapability_t](#)
- struct [unpack_dms_GetDeviceCapabilities_t](#)
- struct [unpack_dms_GetFirmwareRevisions_t](#)
- struct [unpack_dms_GetFirmwareRevision_t](#)
- struct [unpack_dms_GetDeviceSerialNumbers_t](#)
- struct [unpack_dms_GetPRLVersion_t](#)
- struct [unpack_dms_GetNetworkTime_t](#)
- struct [unpack_dms_GetVoiceNumber_t](#)
- struct [unpack_dms_GetDeviceHardwareRev_t](#)
- struct [unpack_dms_GetFSN_t](#)
- struct [unpack_dms_GetDeviceCap_t](#)
- struct [pack_dms_SetPower_t](#)
- struct [unpack_dms_SetPower_t](#)
- struct [unpack_dms_GetBandCapability_t](#)
- struct [unpack_dms_GetUSBComp_t](#)
- struct [pack_dms_SetUSBComp_t](#)
- struct [unpack_dms_SetUSBComp_t](#)
- struct [pack_dms_SetCustFeature_t](#)
- struct [unpack_dms_SetCustFeature_t](#)
- struct [unpack_dms_GetCustFeature_t](#)
- struct [unpack_dms_SetFirmwarePreference_t](#)

- struct [unpack_dms_GetCrashAction_t](#)
- struct [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](#)

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 SLQSFWINFO_MODELID_SZ 20](#)
- [#define SLQSFWINFO_BOOTVERSION_SZ 85](#)
- [#define SLQSFWINFO_APPVERSION_SZ 85](#)
- [#define SLQSFWINFO_SKU_SZ 15](#)
- [#define SLQSFWINFO_PACKAGEID_SZ 85](#)
- [#define SLQSFWINFO_CARRIER_SZ 20](#)
- [#define SLQSFWINFO_PRIVERSION_SZ 10](#)
- [#define SLQSFWINFO_CUR_CARR_NAME 17](#)
- [#define SLQSFWINFO_CUR_CARR_REV 13](#)
- [#define MAX_BUILD_ID_LEN 255](#)
- [#define UNIQUE_ID_LEN 16](#)
- [#define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160](#)
- [#define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20](#)
- [#define DMS_PM_ONLINE 0x00 /* Online */](#)
- [#define DMS_PM_LOW 0x01 /* Low Power */](#)
- [#define DMS_PM_FACTORY 0x02 /* Factory Test Mode */](#)

- `#define DMS_PM_OFFLINE 0x03 /* Offline */`
- `#define DMS_PM_RESET 0x04 /* Reset */`
- `#define DMS_PM_SHUT_DOWN 0x05 /* Shut Down */`
- `#define DMS_PM_PERSISTENT_LOW 0x06 /* Persistent Low Power */`
- `#define DMS_SET_REPORT_ENABLE 1`
- `#define DMS_SET_REPORT_DISABLE 0`
- `#define DMS_SWI_SET_IND_ENABLE 1`
- `#define DMS_SWI_SET_IND_DISABLE 0`

Functions

- `int pack_dms_GetIMSI (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetIMSI (uint8_t *pResp, uint16_t respLen, unpack_dms_GetIMSI_t *pOutput)`
- `int pack_dms_GetModelID (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetModelID (uint8_t *pResp, uint16_t respLen, unpack_dms_GetModelID_t *pOutput)`
- `int pack_dms_GetFirmwareInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetFirmwareInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareInfo_t *pOutput)`
- `int pack_dms_GetPower (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetPower (uint8_t *pResp, uint16_t respLen, unpack_dms_GetPower_t *pOutput)`
- `int pack_dms_GetSerialNumbers (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetSerialNumbers (uint8_t *pResp, uint16_t respLen, unpack_dms_GetSerialNumbers_t *pOutput)`
- `int pack_dms_GetHardwareRevision (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetHardwareRevision (uint8_t *pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t *pOutput)`
- `int pack_dms_SLQSGetBandCapability (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_SLQSGetBandCapability (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetBandCapability_t *pOutput)`
- `int pack_dms_GetDeviceCapabilities (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetDeviceCapabilities (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceCapabilities_t *pOutput)`
- `int pack_dms_GetFirmwareRevisions (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetFirmwareRevisions (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareRevisions_t *pOutput)`
- `int pack_dms_GetFirmwareRevision (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetFirmwareRevision (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareRevision_t *pOutput)`
- `int pack_dms_GetDeviceSerialNumbers (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetDeviceSerialNumbers (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceSerialNumbers_t *pOutput)`
- `int pack_dms_GetPRLVersion (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetPRLVersion (uint8_t *pResp, uint16_t respLen, unpack_dms_GetPRLVersion_t *pOutput)`
- `int pack_dms_GetNetworkTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetNetworkTime (uint8_t *pResp, uint16_t respLen, unpack_dms_GetNetworkTime_t *pOutput)`
- `int pack_dms_GetVoiceNumber (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetVoiceNumber (uint8_t *pResp, uint16_t respLen, unpack_dms_GetVoiceNumber_t *pOutput)`
- `int pack_dms_GetDeviceHardwareRev (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetDeviceHardwareRev (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceHardwareRev_t *pOutput)`
- `int pack_dms_GetFSN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetFSN (uint8_t *pResp, uint16_t respLen, unpack_dms_GetFSN_t *pOutput)`

- int [pack_dms_GetDeviceCap](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetDeviceCap](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetDeviceCap_t](#) *pOutput)
- int [pack_dms_SetPower](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetPower_t](#) *reqArg)
- int [unpack_dms_SetPower](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetPower_t](#) *pOutput)
- int [pack_dms_GetBandCapability](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetBandCapability](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetBandCapability_t](#) *pOutput)
- int [pack_dms_GetUSBComp](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetUSBComp](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetUSBComp_t](#) *pOutput)
- int [pack_dms_SetUSBComp](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetUSBComp_t](#) *reqArg)
- int [unpack_dms_SetUSBComp](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetUSBComp_t](#) *pOutput)
- int [pack_dms_SetCustFeature](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetCustFeature_t](#) *reqArg)
- int [unpack_dms_SetCustFeature](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetCustFeature_t](#) *pOutput)
- int [pack_dms_GetCustFeature](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetCustFeature](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetCustFeature_t](#) *pOutput)
- int [pack_dms_SetFirmwarePreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_SetFirmwarePreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetFirmwarePreference_t](#) *pOutput)
- int [pack_dms_GetCrashAction](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetCrashAction](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetCrashAction_t](#) *pOutput)
- int [pack_dms_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)

9.3.1 Macro Definition Documentation

9.3.1.1 `#define DMS_IMGDETAILS_LEN 16`

9.3.1.2 `#define DMS_MAX_CUST_ID_LEN 64`

9.3.1.3 `#define DMS_MAX_CUST_VALUE_LEN 8`

9.3.1.4 `#define DMS_PM_FACTORY 0x02 /* Factory Test Mode */`

9.3.1.5 `#define DMS_PM_LOW 0x01 /* Low Power */`

9.3.1.6 `#define DMS_PM_OFFLINE 0x03 /* Offline */`

9.3.1.7 `#define DMS_PM_ONLINE 0x00 /* Online */`

9.3.1.8 `#define DMS_PM_PERSISTENT_LOW 0x06 /* Persistent Low Power */`

9.3.1.9 `#define DMS_PM_RESET 0x04 /* Reset */`

9.3.1.10 `#define DMS_PM_SHUT_DOWN 0x05 /* Shut Down */`

9.3.1.11 `#define DMS_SET_REPORT_DISABLE 0`

9.3.1.12 `#define DMS_SET_REPORT_ENABLE 1`

9.3.1.13 `#define DMS_SWI_SET_IND_DISABLE 0`

9.3.1.14 `#define DMS_SWI_SET_IND_ENABLE 1`

9.3.1.15 `#define DMS_UINT8_MAX_STRING_SZ 255`

9.3.1.16 `#define MAX_BUILD_ID_LEN 255`

9.3.1.17 `#define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160`

9.3.1.18 `#define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20`

9.3.1.19 `#define SLQSFWINFO_APPVERSION_SZ 85`

9.3.1.20 `#define SLQSFWINFO_BOOTVERSION_SZ 85`

9.3.1.21 `#define SLQSFWINFO_CARRIER_SZ 20`

9.3.1.22 `#define SLQSFWINFO_CUR_CARR_NAME 17`

9.3.1.23 `#define SLQSFWINFO_CUR_CARR_REV 13`

9.3.1.24 `#define SLQSFWINFO_MODELID_SZ 20`

9.3.1.25 `#define SLQSFWINFO_PACKAGEID_SZ 85`

9.3.1.26 `#define SLQSFWINFO_PRIVERSION_SZ 10`

9.3.1.27 `#define SLQSFWINFO_SKU_SZ 15`

9.3.1.28 `#define UNIQUE_ID_LEN 16`

9.3.2 Function Documentation

9.3.2.1 `int pack_dms_GetActivationState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Get Activation State pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.2 `int pack_dms_GetBandCapability (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Band Capability pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.3 int pack_dms_GetCrashAction (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Crash Action pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.4 int pack_dms_GetCustFeature (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)

Get Custom Feature pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.5 int pack_dms_GetCustFeaturesV2 (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_GetCustFeaturesV2_t * reqArg)

9.3.2.6 int pack_dms_GetDeviceCap (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)

Get Device Capabilities pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.7 int pack_dms_GetDeviceCapabilities (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)

get device capability pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.8 `int pack_dms_GetDeviceHardwareRev (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Hardware Revision pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.9 `int pack_dms_GetDeviceMfr (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Manufacture pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.10 `int pack_dms_GetDeviceSerialNumbers (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get Device Serial Number pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.11 `int pack_dms_GetFirmwareInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get firmware info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.12 `int pack_dms_GetFirmwareRevision (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get Firmware Revision pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.13 `int pack_dms_GetFirmwareRevisions (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get Firmware Revisions pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.14 int pack_dms_GetFSN (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get FSN pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.15 int pack_dms_GetHardwareRevision (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get hardware revision pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.16 int pack_dms_GetIMSI (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get IMSI pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.17 `int pack_dms_GetModelID (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get model id pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.18 `int pack_dms_GetNetworkTime (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Network Time pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.19 `int pack_dms_GetPower (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get power pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.20 int pack_dms_GetPRLVersion (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get PRL Versions pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.21 int pack_dms_GetSerialNumbers (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get serial numbers pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.22 int pack_dms_GetUSBComp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get USB Comp pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.23 int pack_dms_GetVoiceNumber (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)

Get Voice Number pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.24 int pack_dms_SetCustFeature (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetCustFeature_t * reqArg)

Set Custom Feature pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.25 int pack_dms_SetCustFeaturesV2 (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetCustFeaturesV2_t * reqArg)

Set Cust Features pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.26 `int pack_dms_SetEventReport (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetEventReport_t * reqArg)`

Set Event Report pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.27 `int pack_dms_SetFirmwarePreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Set Firmware Preference pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.28 `int pack_dms_SetPower (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetPower_t * reqArg)`

Set Power pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.29 int pack_dms_SetUSBComp (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetUSBComp_t * reqArg)

Set USB Comp pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.30 int pack_dms_SLQSDmsSwiGetResetInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

To get reset info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.31 int pack_dms_SLQSDmsSwiIndicationRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSDmsSwiIndicationRegister_t * reqArg)

Set the registration state for different indication pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.32 `int pack_dms_SLQSGetBandCapability (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get band capability pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.33 `int pack_dms_SLQSSwiClearDyingGaspStatistics (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Clear Dying GASP Statistics pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.34 `int pack_dms_SLQSSwiGetDyingGaspCfg (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Get Dying GASP Config pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.35 int pack_dms_SLQSSwiGetDyingGaspStatistics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get Dying GASP Statistics pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.36 int pack_dms_SLQSSwiGetFirmwareCurr (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get currently active image pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.37 int pack_dms_SLQSSwiSetDyingGaspCfg (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SLQSSwiSetDyingGaspCfg_t * *reqArg*)

Set Dying GASP Config pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.38 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.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.39 int unpack_dms_GetActivationState (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetActivationState_t * *pOutput*)

Get Activation State unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.40 int unpack_dms_GetBandCapability (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetBandCapability_t * *pOutput*)

Get Band Capabilities unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.41 int unpack_dms_GetCrashAction (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetCrashAction_t * *pOutput*)

Get Crash Action unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.42 int unpack_dms_GetCustFeature (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetCustFeature_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

9.3.2.43 int unpack_dms_GetCustFeaturesV2 (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetCustFeaturesV2_t * *pOutput*)

9.3.2.44 int unpack_dms_GetDeviceCap (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceCap_t * *pOutput*)

Get Device Capabilities unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.45 int unpack_dms_GetDeviceCapabilities (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceCapabilities_t * *pOutput*)

get device capability unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.46 int unpack_dms_GetDeviceHardwareRev (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceHardwareRev_t * *pOutput*)

Get Hardware Revision unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.47 int unpack_dms_GetDeviceMfr (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceMfr_t * *pOutput*)

Get Manufacture unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.48 int unpack_dms_GetDeviceSerialNumbers (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_GetDeviceSerialNumbers_t * *pOutput*)

get Device Serial Number unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.49 int unpack_dms_GetFirmwareInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmwareInfo_t *
pOutput)

get firmware info unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.50 int unpack_dms_GetFirmwareRevision (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmware-
Revision_t * *pOutput*)

get Firmware Revision unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.51 `int unpack_dms_GetFirmwareRevisions (uint8_t * pResp, uint16_t respLen, unpack_dms_GetFirmwareRevisions_t * pOutput)`

get Firmware Revisions unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.52 `int unpack_dms_GetFSN (uint8_t * pResp, uint16_t respLen, unpack_dms_GetFSN_t * pOutput)`

Get FSN unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.53 `int unpack_dms_GetHardwareRevision (uint8_t * pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t * pOutput)`

get hardware revision unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.54 int unpack_dms_GetIMSI (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetIMSI_t * *pOutput*)

get model id unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.55 int unpack_dms_GetModelID (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetModelID_t * *pOutput*)

get model id unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.56 int unpack_dms_GetNetworkTime (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetNetworkTime_t * *pOutput*)

Get Network Time unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.57 int unpack_dms_GetPower (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPower_t * *pOutput*)

get power unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.58 int unpack_dms_GetPRLVersion (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPRLVersion_t * *pOutput*)

Get PRL Versions unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.59 int unpack_dms_GetSerialNumbers (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetSerialNumbers_t * *pOutput*)

get serial numbers unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.60 int unpack_dms_GetUSBComp (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetUSBComp_t * *pOutput*)

Get USB Comp unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.61 int unpack_dms_GetVoiceNumber (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetVoiceNumber_t * *pOutput*)

Get Voice Number unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.62 int unpack_dms_SetCustFeature (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetCustFeature_t * *pOutput*)

Set 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

9.3.2.63 `int unpack_dms_SetCustFeaturesV2 (uint8_t * pResp, uint16_t respLen, unpack_dms_SetCustFeaturesV2_t * pOutput)`

Set Cust features unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.64 `int unpack_dms_SetEventReport (uint8_t * pResp, uint16_t respLen, unpack_dms_SetEventReport_t * pOutput)`

Set Event Report unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.65 `int unpack_dms_SetEventReport_ind (uint8_t * pResp, uint16_t respLen, unpack_dms_SetEventReport_ind_t * pOutput)`

Event Report Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.66 int unpack_dms_SetFirmwarePreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetFirmwarePreference_t * *pOutput*)

Set Firmware Preference unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.67 int unpack_dms_SetPower (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetPower_t * *pOutput*)

Set Power unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.68 int unpack_dms_SetUSBComp (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetUSBComp_t * *pOutput*)

Set USB Comp unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.69 `int unpack_dms_SLQSDmsSwiGetResetInfo (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_t * pOutput)`

To get reset info unpack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.70 `int unpack_dms_SLQSDmsSwiGetResetInfo_Ind (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t * pOutput)`

DMS reset info Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.71 `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

9.3.2.72 int unpack_dms_SLQSGetBandCapability (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSGetBandCapability_t * *pOutput*)

get band capability unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.73 int unpack_dms_SLQSSwiClearDyingGaspStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiClearDyingGaspStatistics_t * *pOutput*)

Clear Dying GASP Statistics unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.74 int unpack_dms_SLQSSwiGetDyingGaspCfg (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspCfg_t * *pOutput*)

Get Dying GASP Config unpack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.75 int unpack_dms_SLQSSwiGetDyingGaspStatistics (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_SLQSSwiGetDyingGaspStatistics_t * *pOutput*)

Get Dying GASP Statistics unpack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.76 int unpack_dms_SLQSSwiGetFirmwareCurr (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_SLQSSwiGetFirmwareCurr_t * *pOutput*)

get currently active image unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.77 int unpack_dms_SLQSSwiSetDyingGaspCfg (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_SLQSSwiSetDyingGaspCfg_t * *pOutput*)

Set Dying GASP Config unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.78 int unpack_dms_UIMGetICCID (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMGetICCID_t * *pOutput*)

Unpacks the UIMGetICCID response message to a user-provided response structure.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.4 fms.h File Reference

Data Structures

- struct [CarrierImage_t](#)
- struct [pack_fms_GetImagesPreference_t](#)
- struct [FMSImageElement](#)
- struct [FMSPrefImageList](#)
- struct [unpack_fms_GetImagesPreference_t](#)
- struct [pack_fms_GetStoredImages_t](#)
- struct [FMSImageIdElement](#)
- struct [FMSImageIdEntries](#)
- struct [FMSImageList](#)
- struct [unpack_fms_GetStoredImages_t](#)
- struct [pack_fms_SetImagesPreference_t](#)
- struct [unpack_fms_SetImagesPreference_t](#)

Macros

- #define [FMS_GOBI_MBN_IMG_ID_STR_LEN](#) 16
- #define [FMS_GOBI_MBN_BUILD_ID_STR_LEN](#) 100
- #define [FMS_GOBI_LISTENTRIES_MAX](#) 2

- `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`
- `#define FMS_MAX_IMAGE_ID_ELEMENT 50`
- `#define FMS_IMAGE_ID_MAX_ENTRIES 2`
- `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`
- `#define FMS_IMAGE_ID_IMG_ID_LEN 16`
- `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`
- `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

Functions

- `int pack_fms_GetImagesPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_fms_GetImagesPreference_t *reqArg)`
- `int unpack_fms_GetImagesPreference (uint8_t *pResp, uint16_t respLen, unpack_fms_GetImagesPreference_t *pOutput)`
- `int pack_fms_GetStoredImages (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_fms_GetStoredImages_t *reqArg)`
- `int unpack_fms_GetStoredImages (uint8_t *pResp, uint16_t respLen, unpack_fms_GetStoredImages_t *pOutput)`
- `int pack_fms_SetImagesPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_fms_SetImagesPreference_t *reqArg)`
- `int unpack_fms_SetImagesPreference (uint8_t *pResp, uint16_t respLen, unpack_fms_SetImagesPreference_t *pOutput)`
- `uint32_t GetValidFwPriCombinations (FMSImageList *pStoredImageList, uint32_t *pValidCombinationSize, CarrierImage_t *pValidCombinations)`

9.4.1 Macro Definition Documentation

9.4.1.1 `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`

9.4.1.2 `#define FMS_GOBI_LISTENTRIES_MAX 2`

9.4.1.3 `#define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100`

9.4.1.4 `#define FMS_GOBI_MBN_IMG_ID_STR_LEN 16`

9.4.1.5 `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`

9.4.1.6 `#define FMS_IMAGE_ID_IMG_ID_LEN 16`

9.4.1.7 `#define FMS_IMAGE_ID_MAX_ENTRIES 2`

9.4.1.8 `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

9.4.1.9 `#define FMS_MAX_IMAGE_ID_ELEMENT 50`

9.4.1.10 `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`

9.4.2 Function Documentation

9.4.2.1 `uint32_t GetValidFwPriCombinations (FMSImageList * pStoredImageList, uint32_t * pValidCombinationSize, CarrierImage_t * pValidCombinations)`

This API distills valid Firmware/PRI combinations from GetStoredImages result

Parameters

in	<i>pStoredImage-List</i>	<ul style="list-style-type: none"> image list returned from GetStoredImages See FMSImageList
in, out	<i>pValid-CombinationSize</i>	<ul style="list-style-type: none"> number of combination passed in and returned
out	<i>pValid-Combinations</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 combinatons returned

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.4.2.2 `int pack_fms_GetImagesPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_fms_GetImagesPreference_t * reqArg)`

Get Images Preference pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

9.4.2.3 `int pack_fms_GetStoredImages (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_fms_GetStoredImages_t * reqArg)`

Get Images Preference pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

9.4.2.4 `int pack_fms_SetImagesPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_fms_SetImagesPreference_t * reqArg)`

Set Images Preference pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

9.4.2.5 `int unpack_fms_GetImagesPreference (uint8_t * pResp, uint16_t respLen, unpack_fms_GetImagesPreference_t * pOutput)`

Get Images Preference unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

9.4.2.6 `int unpack_fms_GetStoredImages (uint8_t * pResp, uint16_t respLen, unpack_fms_GetStoredImages_t * pOutput)`

Get Images Preference unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

9.4.2.7 `int unpack_fms_SetImagesPreference (uint8_t * pResp, uint16_t respLen, unpack_fms_SetImagesPreference_t * pOutput)`

Set Images Preference unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

9.5 loc.h File Reference

Data Structures

- struct [loc_LocApplicationInfo](#)
- struct [loc_SV](#)
- struct [loc_SVInfo](#)
- struct [loc_GnssData](#)
- struct [loc_CellDb](#)
- struct [loc_ClkInfo](#)
- struct [loc_BdsSV](#)
- struct [loc_BdsSVInfo](#)
- struct [pack_loc_EventRegister_t](#)
- struct [unpack_loc_EventRegister_t](#)

- struct [pack_loc_SetExtPowerState_t](#)
- struct [unpack_loc_SetExtPowerState_t](#)
- struct [pack_loc_Start_t](#)
- struct [unpack_loc_Start_t](#)
- struct [pack_loc_Stop_t](#)
- struct [unpack_loc_Stop_t](#)
- struct [pack_loc_SetOperationMode_t](#)
- struct [unpack_loc_SetOperationMode_t](#)
- struct [pack_loc_Delete_Assist_Data_t](#)
- struct [unpack_loc_Delete_Assist_Data_t](#)
- struct [loc_precisionDilution](#)
- struct [loc_sensorDataUsage](#)
- struct [loc_svUsedforFix](#)
- struct [loc_gpsTime](#)
- struct [unpack_loc_PositionRpt_Ind_t](#)
- struct [unpack_loc_EngineState_Ind_t](#)

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 [LOCEVENTMASKSETSPITSTREAMINGREPORT](#) 0x00001000
- #define [LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ](#) 0x00002000
- #define [LOCEVENTMASKNIGEOFENCENOTIFICATION](#) 0x00004000
- #define [LOCEVENTMASKGEOFENCEGENALERT](#) 0x00008000
- #define [LOCEVENTMASKGEOFENCEBREACHNOTIFICATION](#) 0x00010000
- #define [LOCEVENTMASKPEDOMETERCONTROL](#) 0x00020000
- #define [LOCEVENTMASKMOTIONDATACONTROL](#) 0x00040000
- #define [LOCEVENTMASKBATCHFULLNOTIFICATION](#) 0x00080000
- #define [LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT](#) 0x00100000
- #define [LOCEVENTMASKINJECTWIFIAPDATAREQ](#) 0x00200000
- #define [LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION](#) 0x00400000
- #define [LOCEVENTMASKVEHICLEDATAREADYSTATUS](#) 0x00800000
- #define [LOCEVENTMASKGNSSMEASUREMENTREPORT](#) 0x01000000
- #define [LOCEVENTMASKINVALIDVALUE](#) 0xFFFFFFFF

Enumerations

- enum {
[eQMI_LOC_SESS_STATUS_SUCCESS](#) =0,
[eQMI_LOC_SESS_STATUS_IN_PROGRESS](#) =1,
[eQMI_LOC_SESS_STATUS_FAILURE](#) =2,
[eQMI_LOC_SESS_STATUS_TIMEOUT](#) =3 }

Functions

- int [pack_loc_EventRegister](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_loc_EventRegister_t](#) *reqArg)
- int [unpack_loc_EventRegister](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_EventRegister_t](#) *pOutput)
- int [pack_loc_SetExtPowerState](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_loc_SetExtPowerState_t](#) *reqArg)
- int [unpack_loc_SetExtPowerState](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_SetExtPowerState_t](#) *pOutput)
- int [pack_loc_Start](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_loc_Start_t](#) *reqArg)
- int [unpack_loc_Start](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_Start_t](#) *pOutput)
- int [pack_loc_Stop](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_loc_Stop_t](#) *reqArg)
- int [unpack_loc_Stop](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_Stop_t](#) *pOutput)
- int [pack_loc_SetOperationMode](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_loc_SetOperationMode_t](#) *reqArg)
- int [unpack_loc_SetOperationMode](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_SetOperationMode_t](#) *pOutput)
- int [pack_loc_DeleteAssistData](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_loc_Delete_Assist_Data_t](#) *reqArg)
- int [unpack_loc_DeleteAssistData](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_Delete_Assist_Data_t](#) *pOutput)
- int [unpack_loc_PositionRpt_Ind](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_PositionRpt_Ind_t](#) *pOutput)
- int [unpack_loc_EngineState_Ind](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_loc_EngineState_Ind_t](#) *pOutput)

9.5.1 Macro Definition Documentation

9.5.1.1 `#define LOC_UINT8_MAX_STRING_SZ 255`

9.5.1.2 `#define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000`

The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.

9.5.1.3 `#define LOCEVENTMASKENGINESTATE 0x00000080`

The control point must enable this mask to receive engine state report event indications.

9.5.1.4 `#define LOCEVENTMASKFIXSESSIONSTATE 0x00000100`

The control point must enable this mask to receive fix session status report event indications.

9.5.1.5 `#define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000`

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

9.5.1.6 `#define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000`

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.

9.5.1.7 #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000

The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, for example, if GPS is turned off or if the network is unavailable.

9.5.1.8 #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000

The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG(Not yet supported).

9.5.1.9 #define LOCEVENTMASKGNSSSVINFO 0x00000002

The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate.

9.5.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040

The control point must enable this mask to receive position injection request event indications.

9.5.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020

The control point must enable this mask to receive predicted orbits request event indications.

9.5.1.12 #define LOCEVENTMASKINJECTTIMERREQ 0x00000010

The control point must enable this mask to receive time injection request event indications.

9.5.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000

The control point must enable this mask to receive Wi-Fi Access Point (AP) data inject request event indications.

9.5.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF

Invalid Event Mask

9.5.1.15 #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000

The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.

9.5.1.16 #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000

The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.

9.5.1.17 #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000

The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.

9.5.1.18 #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000

The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.

9.5.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008

The control point must enable this mask to receive NI Notify/Verify request event indications.

9.5.1.20 #define LOCEVENTMASKNMEA 0x00000004

The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

9.5.1.21 #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000

The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

9.5.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001

The control point must enable this mask to receive position report event indications.

9.5.1.23 #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.).

9.5.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000

The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications.

9.5.1.25 #define LOCEVENTMASKTIMESYNCREQ 0x00000800

The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

9.5.1.26 #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).

9.5.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200

The control point must enable this mask to receive Wi-Fi position request event indications.

9.5.2 Enumeration Type Documentation

9.5.2.1 anonymous enum

Enumerator

eQMI_LOC_SESS_STATUS_SUCCESS
eQMI_LOC_SESS_STATUS_IN_PROGRESS
eQMI_LOC_SESS_STATUS_FAILURE
eQMI_LOC_SESS_STATUS_TIMEOUT

9.5.3 Function Documentation

9.5.3.1 `int pack_loc_DeleteAssistData (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Delete_Assist_Data_t * reqArg)`

Delete Assistant Data pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.2 `int pack_loc_EventRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_EventRegister_t * reqArg)`

Event Register pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.3 `int pack_loc_SetExtPowerState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetExtPowerState_t * reqArg)`

Set Ext Power State pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.4 int pack_loc_SetOperationMode (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetOperationMode_t * reqArg)

Set Operation Mode pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.5 int pack_loc_Start (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Start_t * reqArg)

LOC Start pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.6 int pack_loc_Stop (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Stop_t * reqArg)

Loc Stop pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.7 int unpack_loc_DeleteAssistData (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Delete_Assist_Data_t * *pOutput*)

Delete Assistant Data unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.8 int unpack_loc_EngineState_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EngineState_Ind_t * *pOutput*)

Loc Engine State Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.9 int unpack_loc_EventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EventRegister_t * *pOutput*)

Event Register unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.10 int unpack_loc_PositionRpt_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_PositionRpt_Ind_t * *pOutput*)

Loc Position Report Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.11 int unpack_loc_SetExtPowerState (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerState_t * *pOutput*)

Set Ext Power State unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.12 int unpack_loc_SetOperationMode (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetOperationMode_t * *pOutput*)

Set Operation Mode unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.13 int unpack_loc_Start (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Start_t * *pOutput*)

Loc Start unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.14 int unpack_loc_Stop (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Stop_t * *pOutput*)

Loc Stop unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6 nas.h File Reference

Data Structures

- struct [unpack_nas_GetSignalStrengths_t](#)

- struct [unpack_nas_SLQSGetSysSelectionPref_t](#)
- struct [nas_netSelectionPref](#)
- struct [nas_acqOrderPref](#)
- struct [nas_CSGID](#)
- struct [pack_nas_SLQSSetSysSelectionPref_t](#)
- struct [pack_nas_SLQSNasIndicationRegisterExt_t](#)
- struct [RFBandInfoElements](#)
- struct [unpack_nas_GetRFInfo_t](#)
- struct [cdmaSSInfo](#)
- struct [hdrSSInfo](#)
- struct [lteSSInfo](#)
- struct [tdscdmaSigInfoExt](#)
- struct [unpack_nas_SLQSNasGetSigInfo_t](#)
- struct [unpack_nas_SLQSNasSigInfoCallback_t](#)
- struct [unpack_nas_GetHomeNetwork_t](#)
- struct [nas_SrvStatusInfo](#)
- struct [nas_GSMSrvStatusInfo](#)
- struct [nas_sysInfoCommon](#)
- struct [nas_CDMASysInfo](#)
- struct [nas_HDRSysInfo](#)
- struct [nas_GSMSysInfo](#)
- struct [nas_WCDMASysInfo](#)
- struct [nas_LTESysInfo](#)
- struct [nas_AddCDMASysInfo](#)
- struct [nas_AddSysInfo](#)
- struct [nas_CallBarringSysInfo](#)
- struct [unpack_nas_SLQSGetSysInfo_t](#)
- struct [unpack_nas_SLQSSysInfoCallback_t](#)
- struct [unpack_nas_GetServingNetwork_t](#)
- struct [unpack_nas_GetServingNetworkCapabilities_t](#)
- struct [nas_QmiNas3GppNetworkInfo](#)
- struct [nas_QmiNas3GppNetworkRAT](#)
- struct [nas_QmisNasPcsDigit](#)
- struct [unpack_nas_PerformNetworkScan_t](#)
- struct [unpack_nas_SLQSSwiGetLteCQI_t](#)
- struct [nas_CommInfo](#)
- struct [nas_LTEInfo](#)
- struct [unpack_nas_SLQSNasSwiModemStatus_t](#)
- struct [nas_servSystem](#)
- struct [nas_dataSrvCapabilities](#)
- struct [nas_currentPLMN](#)
- struct [nas_roamIndList](#)
- struct [nas_qaQmi3Gpp2TimeZone](#)
- struct [nas_detailSvcInfo](#)
- struct [nas_CDMASysInfoExt](#)
- struct [nas_callBarStatus](#)
- struct [unpack_nas_SLQSGetServingSystem_t](#)
- struct [nas_rxSignalStrengthListElement](#)
- struct [nas_ecioListElement](#)
- struct [nas_errorRateListElement](#)
- struct [nas_rsrqInformation](#)
- struct [nas_lteSnrinformation](#)
- struct [nas_lteRsrpinformation](#)
- struct [unpack_nas_SLQSGetSignalStrength_t](#)
- struct [nas_SLQSSignalStrengthsIndReq](#)

- struct [pack_nas_SLQSSetSignalStrengthsCallback_t](#)
- struct [nas_SLQSSignalStrengthsInformation](#)
- struct [nas_RejectReasonTlv](#)
- struct [nas_SignalStrengthTlv](#)
- struct [nas_RFInfoTlv](#)
- struct [nas_SLQSSignalStrengthsTlv](#)
- struct [unpack_nas_SetEventReportInd_t](#)
- struct [unpack_nas_GetCDMANetworkParameters_t](#)
- struct [pack_nas_SetACCOLC_t](#)
- struct [nas_CDMARSSIThresh](#)
- struct [nas_CDMAECIOThresh](#)
- struct [nas_HDRRSSIThresh](#)
- struct [nas_HDRECIOThresh](#)
- struct [nas_HDRSINRThreshold](#)
- struct [nas_HDRIOThresh](#)
- struct [nas_GSMRSSIThresh](#)
- struct [nas_WCDMARSSIThresh](#)
- struct [nas_WCDMAECIOThresh](#)
- struct [nas_LTERSSIThresh](#)
- struct [nas_LTESNRThreshold](#)
- struct [nas_LTERSRQThresh](#)
- struct [nas_LTERSRPThresh](#)
- struct [nas_LTESigRptConfig](#)
- struct [nas_TDSCDMARSCPTthresh](#)
- struct [nas_TDSCDMARSSIThresh](#)
- struct [nas_TDSCDMAECIOThresh](#)
- struct [nas_TDSCDMASINRThresh](#)
- struct [pack_nas_SLQSNasConfigSigInfo2_t](#)
- struct [unpack_nas_SetDataCapabilitiesCallback_ind_t](#)
- struct [unpack_nas_GetNetworkPreference_t](#)
- struct [pack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetRoamingIndicatorCallback_ind_t](#)
- struct [NAServingSystemInfo](#)
- struct [unpack_nas_SetServingSystemCallback_ind_t](#)
- struct [NASPhyCaAggScellIndType](#)
- struct [NASPhyCaAggScellIDBw](#)
- struct [NASPhyCaAggScellInfo](#)
- struct [NASPhyCaAggPcellInfo](#)
- struct [NASPhyCaAggScellIndex](#)
- struct [NasGetLTECphyCAInfo](#)
- struct [unpack_nas_SlqsGetLTECphyCAInfo_t](#)
- struct [NASEmergencyModeTlv](#)
- struct [NASModePreferenceTlv](#)
- struct [NASBandPreferenceTlv](#)
- struct [NASPRLPreferenceTlv](#)
- struct [NASRoamPreferenceTlv](#)
- struct [NASLTEBandPreferenceTlv](#)
- struct [NASNetSelPreferenceTlv](#)
- struct [NASServDomainPrefTlv](#)
- struct [NASGWAcqOrderPrefTlv](#)
- struct [NASQmiCbkNasSystemSelPrefInd](#)
- struct [unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t](#)
- struct [NASOTAMessageTlv](#)
- struct [NASLteNasReleaseInfoTlv](#)

- struct [NASTimeInfoTlv](#)
- struct [NASQmiCbkNasSwtOTAMessageInd](#)
- struct [unpack_nas_SLQSNasSwtOTAMessageCallback_ind_t](#)
- struct [nas_MNRInfo](#)
- struct [pack_nas_SLQSInitiateNetworkRegistration_t](#)
- struct [pack_nas_SLQSNasSwtOTAMessageCallback_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](#)

Macros

- [#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048](#)
- [#define NAS_MAX_NUM_NETWORKS 30](#)
- [#define NAS_MAX_DESCRIPTION_LENGTH 255](#)
- [#define NAS_PLMN_LENGTH 3](#)
- [#define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255](#)

Enumerations

- enum [LIBPACK_NAS_LTE_CPHY_SCELL_STATE](#) {
[eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED](#) =0x00,
[eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED](#) =0x01,
[eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED](#) =0x02 }
- enum [LIBPACK_NAS_LTE_CPHY_CA_BW_NRB](#) {
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6](#) =0x00,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15](#) =0x01,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25](#) =0x02,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50](#) =0x03,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75](#) =0x04,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100](#) =0x05 }

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)
- int [pack_nas_SLQSSetBandPreference](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint64_t bandPref)
- int [unpack_nas_SLQSSetBandPreference](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSNasIndicationRegisterExt](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSNasIndicationRegisterExt_t](#) *pReqParam)
- int [unpack_nas_SLQSNasIndicationRegisterExt](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_GetRFInfo](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetRFInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetRFInfo_t](#) *pOutput)
- int [pack_nas_SLQSNasGetSigInfo](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSNasGetSigInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasGetSigInfo_t](#) *pOutput)
- int [unpack_nas_SLQSNasSigInfoCallback](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSigInfoCallback_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](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSysInfoCallback_t](#) *pOutput)
- int [pack_nas_GetServingNetwork](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetServingNetwork](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetServingNetwork_t](#) *pOutput)
- int [pack_nas_GetServingNetworkCapabilities](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetServingNetworkCapabilities](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetServingNetworkCapabilities_t](#) *pOutput)
- int [pack_nas_PerformNetworkScan](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_PerformNetworkScan](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_PerformNetworkScan_t](#) *pOutput)
- int [pack_nas_SLQSSwiGetLteCQI](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSSwiGetLteCQI](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSwiGetLteCQI_t](#) *pOutput)
- int [pack_nas_SLQSNasSwiModemStatus](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSNasSwiModemStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSwiModemStatus_t](#) *pOutput)
- int [pack_nas_SLQSSetServingSystem](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSSetServingSystem](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSetServingSystem_t](#) *pOutput)
- int [pack_nas_SLQSSetSignalStrength](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint16_t reqMask)
- int [unpack_nas_SLQSSetSignalStrength](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSetSignalStrength_t](#) *pOutput)
- int [pack_nas_SLQSSetSignalStrengthsCallback](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSSetSignalStrengthsCallback_t](#) *pReqParam)

- int [unpack_nas_SLQSSetSignalStrengthsCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SetRFInfoCallback](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int [unpack_nas_SetRFInfoCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SetLURejectCallback](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int [unpack_nas_SetLURejectCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_nas_SetEventReportInd](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetEventReportInd_t](#) *pOutput)
- int [pack_nas_GetCDMANetworkParameters](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetCDMANetworkParameters](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetCDMA-NetworkParameters_t](#) *pOutput)
- int [pack_nas_GetANAAAAAuthenticationStatus](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetANAAAAAuthenticationStatus](#) (uint8_t *pResp, uint16_t respLen, uint32_t *pAuthStatus)
- int [pack_nas_GetACCOLC](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetACCOLC](#) (uint8_t *pResp, uint16_t respLen, uint8_t *pAccolc)
- int [pack_nas_SetACCOLC](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SetACCOLC_t](#) reqParam)
- int [unpack_nas_SetACCOLC](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSNasConfigSigInfo2](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_S-LQSNasConfigSigInfo2_t](#) *pReqParam)
- int [unpack_nas_SLQSNasConfigSigInfo2](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_nas_SetDataCapabilitiesCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetData-CapabilitiesCallback_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_GetNetwork-Preference_t](#) *pOutput)
- int [pack_nas_SetNetworkPreference](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_Set-NetworkPreference_t](#) *reqArg)
- int [unpack_nas_SetNetworkPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetNetwork-Preference_t](#) *pOutput)
- int [unpack_nas_SetRoamingIndicatorCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_Set-RoamingIndicatorCallback_ind_t](#) *pOutput)
- int [unpack_nas_SetServingSystemCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_Set-ServingSystemCallback_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_SlqsGetLTECphy-CAInfo_t](#) *pOutput)
- int [unpack_nas_SLQSSetSysSelectionPrefCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_S-LQSSetSysSelectionPrefCallBack_ind_t](#) *pOutput)
- int [unpack_nas_SLQSNasSwiOTAMessageCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_S-LQSNasSwiOTAMessageCallback_ind_t](#) *pOutput)
- int [pack_nas_SLQSInitiateNetworkRegistration](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack-nas_SLQSInitiateNetworkRegistration_t](#) *pReqParam)
- int [unpack_nas_SLQSInitiateNetworkRegistration](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSNasSwiOTAMessageCallback](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSNasSwiOTAMessageCallback_t](#) *pReqParam)
- int [unpack_nas_SLQSNasSwiOTAMessageCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSGetPLMNName](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQ-SGetPLMNName_t](#) *reqArg)
- int [unpack_nas_SLQSGetPLMNName](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetPLMN-Name_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_SLQSNas-GetCellLocationInfo_t](#) *pOutput)

9.6.1 Macro Definition Documentation

9.6.1.1 `#define NAS_MAX_DESCRIPTION_LENGTH 255`

9.6.1.2 `#define NAS_MAX_NUM_NETWORKS 30`

9.6.1.3 `#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048`

9.6.1.4 `#define NAS_PLMN_LENGTH 3`

9.6.1.5 `#define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255`

9.6.2 Enumeration Type Documentation

9.6.2.1 `enum LIBPACK_NAS_LTE_CPHY_CA_BW_NRB`

Enumerator

eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100

9.6.2.2 `enum LIBPACK_NAS_LTE_CPHY_SCELL_STATE`

Enumerator

eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED

9.6.3 Function Documentation

9.6.3.1 `int pack_nas_GetACCOLC (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.2 `int pack_nas_GetANAAAAAuthenticationStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.3 int pack_nas_GetCDMANetworkParameters (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.4 int pack_nas_GetHomeNetwork (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

get home network pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.5 int pack_nas_GetNetworkPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

9.6.3.6 `int pack_nas_GetRFInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get rf info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.7 `int pack_nas_GetServingNetwork (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.8 `int pack_nas_GetServingNetworkCapabilities (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.9 int pack_nas_GetSignalStrengths (pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen*)

get signal strengths pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.10 int pack_nas_PerformNetworkScan (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.11 int pack_nas_SetACCOLC (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetACCOLC_t *reqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.12 int pack_nas_SetLURejectCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>bEnable</i>	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.13 int pack_nas_SetNetworkPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SetNetworkPreference_t * *reqArg*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pack</i>	default prototype

9.6.3.14 int pack_nas_SetRFInfoCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>bEnable</i>	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.15 int pack_nas_SlqsGetLTEphyCAInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

9.6.3.16 int pack_nas_SLQSGetPLMNName (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SLQSGetPLMNName_t * *reqArg*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request prameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.17 int pack_nas_SLQSGetServingSystem (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.18 int pack_nas_SLQSGetSignalStrength (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint16_t *reqMask*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqMask</i>	request mask for fetching extra signal info

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.19 int pack_nas_SLQSGetSysInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.20 int pack_nas_SLQSGetSysSelectionPref (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.21 int pack_nas_SLQSIInitiateNetworkRegistration (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSIInitiateNetworkRegistration_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.22 int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasConfigSigInfo2_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.23 int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.24 int pack_nas_SLQSNasGetSigInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get sig info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.25 int pack_nas_SLQSNasIndicationRegisterExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SLQSNasIndicationRegisterExt_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.26 int pack_nas_SLQSNasSmiModemStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.27 int pack_nas_SLQSNasSmiOTAMessageCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasSmiOTAMessageCallback_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.28 int pack_nas_SLQSSetBandPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint64_t *bandPref*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>band</i>	preference

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.29 int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SLQSSetSignalStrengthsCallback_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.30 int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SLQSSetSysSelectionPref_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.31 int pack_nas_SLQSSwiGetLteCQI (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.32 int unpack_nas_GetACCOLC (uint8_t * *pResp*, uint16_t *respLen*, uint8_t * *pAccolc*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pAccolc</i>	accolc

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.33 int unpack_nas_GetANAAAuthenticationStatus (uint8_t * *pResp*, uint16_t *respLen*, uint32_t * *pAuthStatus*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>authStatus</i>	auth status

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.34 int unpack_nas_GetCDMANetworkParameters (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_GetCDMANetworkParameters_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	qmi output parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.35 `int unpack_nas_GetHomeNetwork (uint8_t * pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t * pOutput)`

get home network unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.36 `int unpack_nas_GetNetworkPreference (uint8_t * pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t * pOutput)`

9.6.3.37 `int unpack_nas_GetRFInfo (uint8_t * pResp, uint16_t respLen, unpack_nas_GetRFInfo_t * pOutput)`

get rf info unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.38 `int unpack_nas_GetServingNetwork (uint8_t * pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.39 int unpack_nas_GetServingNetworkCapabilities (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_GetServingNetworkCapabilities_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.40 int unpack_nas_GetSignalStrengths (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetSignalStrengths_t *
pOutput)

get signal strengths unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.41 int unpack_nas_PerformNetworkScan (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_PerformNetworkScan_t
* *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.42 int unpack_nas_SetACCOLC (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.43 int unpack_nas_SetDataCapabilitiesCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetDataCapabilitiesCallback_ind_t * *pOutput*)

Data Capabilities indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.44 int unpack_nas_SetEventReportInd (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetEventReportInd_t * *pOutput*)

9.6.3.45 int unpack_nas_SetLURejectCallback (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.46 int unpack_nas_SetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNetworkPreference_t * *pOutput*)

9.6.3.47 int unpack_nas_SetRFInfoCallback (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.48 int unpack_nas_SetRoamingIndicatorCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetRoamingIndicatorCallback_ind_t * *pOutput*)

Roaming indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.49 int unpack_nas_SetServingSystemCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetServingSystemCallback_ind_t * *pOutput*)

9.6.3.50 int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SlqsGetLTECphyCAInfo_t * *pOutput*)

9.6.3.51 int unpack_nas_SLQSGetPLMNName (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetPLMNName_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.52 int unpack_nas_SLQSGetservingSystem (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetservingSystem_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.53 int unpack_nas_SLQSGetSignalStrength (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSignalStrength_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.54 int unpack_nas_SLQSGetSysInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysInfo_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.55 int unpack_nas_SLQSGetSysSelectionPref (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSGetSysSelectionPref_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.56 int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.57 int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.58 int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSNasGetCellLocationInfo_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.59 int unpack_nas_SLQSNasGetSigInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetSigInfo_t
* *pOutput*)

get sig info unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.60 int unpack_nas_SLQSNasIndicationRegisterExt (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.61 int unpack_nas_SLQSNasSigInfoCallback (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSNasSigInfoCallback_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.62 int unpack_nas_SLQSNasSwiModemStatus (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSNasSwiModemStatus_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.63 int unpack_nas_SLQSNasSwiOTAMessageCallback (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.64 int unpack_nas_SLQSNasSwiOTAMessageCallback_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t * *pOutput*)

OTA message indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.65 int unpack_nas_SLQSNasSysInfoCallback (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSSysInfoCallback_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.66 int unpack_nas_SLQSSetBandPreference (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

9.6.3.67 `int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.68 `int unpack_nas_SLQSSetSysSelectionPref (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.69 `int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind (uint8_t * pResp, uint16_t respLen,
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t * pOutput)`

System Selection Preference indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.70 `int unpack_nas_SLQSSwiGetLteCQI (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteCQI_t *
pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.7 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

9.7.1 Detailed Description

Network Access Service API Band Classes table.

9.7.2 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

9.7.2.1 LTE Bands

- 28 - LTE Band Class 28
- 39 - LTE Band Class 39
- 40 - LTE Band Class 40
- 41 - LTE Band Class 41
- 120 - FDD UL:1920-1980; DL:2110-2170; E-UTRA Operating Band 1
- 121 - FDD UL:1850-1910; DL:1930-1990; E-UTRA Operating Band 2
- 122 - FDD UL:1710-1785; DL:1805-1880; E-UTRA Operating Band 3
- 123 - FDD UL:1710-1755; DL:2110-2155; E-UTRA Operating Band 4
- 124 - FDD UL: 824- 849; DL: 869- 894; E-UTRA Operating Band 5
- 125 - FDD UL: 830- 840; DL: 875- 885; E-UTRA Operating Band 6
- 126 - FDD UL:2500-2570; DL:2620-2690; E-UTRA Operating Band 7
- 127 - FDD UL: 880- 915; DL: 925- 960; E-UTRA Operating Band 8
- 128 - FDD UL:1749.9-1784.9; DL:1844.9-1879.9; E-UTRA Operating Band 9
- 129 - FDD UL:1710-1770; DL:2110-2170; E-UTRA Operating Band 10
- 130 - FDD UL:1427.9-1452.9; DL:1475.9-1500.9; E-UTRA Operating Band 11
- 131 - FDD UL:698-716; DL:728-746; E-UTRA Operating Band 12
- 132 - FDD UL: 777- 787; DL: 746-756; E-UTRA Operating Band 13
- 133 - FDD UL: 788- 798; DL: 758-768; E-UTRA Operating Band 14
- 134 - FDD UL: 704-716; DL: 734-746; E-UTRA Operating Band 17
- 135 - TDD LTE UL: 1900-1920; DL: 1900-1920; E-UTRA Operating Band 33
- 136 - TDD LTE UL: 2010-2025; DL: 2010-2025; E-UTRA Operating Band 34
- 137 - TDD LTE UL: 1850-1910; DL: 1850-1910; E-UTRA Operating Band 35
- 138 - TDD LTE UL: 1930-1990; DL: 1930-1990; E-UTRA Operating Band 36
- 139 - TDD LTE UL: 1910-1930; DL: 1910-1930; E-UTRA Operating Band 37
- 140 - TDD LTE UL: 2570-2620; DL: 2570-2620; E-UTRA Operating Band 38
- 141 - TDD LTE UL: 1880-1920; DL: 1880-1920; E-UTRA Operating Band 39
- 142 - TDD LTE UL: 2300-2400; DL: 2300-2400; E-UTRA Operating Band 40
- 143 - FDD LTE UL: 815-830; DL: 860-875; E-UTRA Operating Band 18
- 144 - FDD LTE UL: 830-845; DL: 875-890; E-UTRA Operating Band 19
- 145 - FDD LTE UL: 832-862; DL: 791-821; E-UTRA Operating Band 20
- 146 - FDD LTE UL: 1447.9-1462.9; DL: 1495.9-1510.9; E-UTRA Operating Band 21
- 147 - FDD LTE UL: 1626.5-1660.5; DL: 1525-1559; E-UTRA Operating Band 24
- 148 - FDD LTE UL: 1850-1919.5; DL: 1930-1995; E-UTRA Operating Band 25
- 149 - TDD LTE UL: 2496-2690; DL: 2496-2690; E-UTRA Operating Band 41

- 150 - TDD LTE UL: 3400-3600; DL: 3400-3600; E-UTRA Operating Band 42
- 151 - TDD LTE UL: 3600-3800; DL: 3600-3800; E-UTRA Operating Band 43
- 200 - TD-SCDMA Band A
- 201 - TD-SCDMA Band B
- 202 - TD-SCDMA Band C
- 203 - TD-SCDMA Band D
- 204 - TD-SCDMA Band E
- 205 - TD-SCDMA Band F

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.8 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

9.8.1 Detailed Description

Call Control Return Reasons table.

9.8.2 Call Control Result Reasons (Value - Name - Description)

- 0x01 - QMI_VOICE_REASON_FWD_UNCONDITIONAL - Unconditional call forwarding
- 0x02 - QMI_VOICE_REASON_FWD_MOBILEBUSY - Forward when the mobile is busy
- 0x03 - QMI_VOICE_REASON_FWD_NOREPLY - Forward when there is no reply
- 0x04 - QMI_VOICE_REASON_FWD_UNREACHABLE - Forward when the call is unreachable
- 0x05 - QMI_VOICE_REASON_FWD_ALLFORWARDING - All forwarding
- 0x06 - QMI_VOICE_REASON_FWD_ALLCONDITIONAL - All conditional forwarding
- 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing
- 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal
- 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOM - Outgoing external to home
- 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming
- 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred
- 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI_VOICE_REASON_CALLWAITING - Call waiting
- 0x10 - VOICE_CC_SUPS_RESULT_REASON_CLIP - CLIP
- 0x11 - VOICE_CC_SUPS_RESULT_REASON_CLIR - CLIR
- 0x12 - VOICE_CC_SUPS_RESULT_REASON_COLP - COLP

- 0x13 - VOICE_CC_SUPS_RESULT_REASON_COLR - COLR
- 0x14 - VOICE_CC_SUPS_RESULT_REASON_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

9.9 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

9.9.1 Detailed Description

Wireless Data Service Call End Reasons.

9.9.2 Call end reason codes (Code - Reason)

9.9.2.1 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

9.9.2.2 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

9.9.2.3 WCDMA/GSM call end reasons

- 1000 - Call origination request failed; WCDMA/GSM Only
- 1001 - Client rejected the incoming call; WCDMA/GSM Only
- 1002 - Device has no UMTS service; WCDMA/GSM Only
- 1003 - Network ended the call, look in cc call; WCDMA/GSM Only
- 1004 - LLC(Logical Link Control) or SNDSCP(Sub Network Dependent Convergence Protocol) failure
- 1005 - Insufficient resources, 3GPP equivalent ESM(EPS Session Management) cause code value 26, Insufficient resources
- 1006 - Service option temporarily out of order, 3GPP equivalent ESM(EPS Session Management) cause code value 34, Service option temporarily out of order
- 1007 - PTI already used, 3GPP equivalent ESM(EPS Session Management) cause code value 35, PTI(-Procedure Transaction Identity) already in use
- 1008 - Regular PDP context deactivation, 3GPP equivalent ESM(EPS Session Management) cause code value 36, Regular deactivation
- 1009 - Network failure, 3GPP equivalent ESM(EPS Session Management) cause code value 38, Network failure
- 1010 - Reactivation requested, 3GPP equivalent ESM(EPS Session Management) cause code value 39, Reactivation requested
- 1011 - Protocol error, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 111, Protocol error, unspecified
- 1012 - Operator determined barring, 3GPP equivalent ESM(EPS Session Management) cause code value 8, Operator Determined Barring
- 1013 - Unknown or missing Access Point Name (APN), 3GPP equivalent ESM(EPS Session Management) cause code value 27, Missing or unknown APN
- 1014 - Unknown PDP address or PDP type, 3GPP equivalent ESM(EPS Session Management) cause code value 28, Unknown PDN type
- 1015 - Activation rejected by GGSN, 3GPP equivalent ESM(EPS Session Management) cause code value 30, Requested rejected by Serving GW or PDN GW

- 1016 - Activation rejected, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 31, Request rejected, unspecified
- 1017 - Service option not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 32, Service option not supported
- 1018 - Requested service option not subscribed, 3GPP equivalent ESM(EPS Session Management) cause code value 33, Requested service option not subscribed
- 1019 - EPS Quality of Service (QoS) not accepted, 3GPP equivalent ESM(EPS Session Management) cause code value 37, EPS QoS not accepted
- 1020 - Semantic error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 41, Semantic error in the TFT operation
- 1021 - Syntactical error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 42, Syntactical error in the TFT operation
- 1022 - Unknown PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 43, Invalid EPS bearer identity
- 1023 - Semantic errors in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 44, Semantic errors in packet filter(s)
- 1024 - Syntactical error in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 45, Syntactical errors in packet filter(s)
- 1025 - PDP context without TFT already activated, 3GPP equivalent ESM(EPS Session Management) cause code value 46, Unused
- 1026 - Invalid transaction identifier value, 3GPP equivalent ESM(EPS Session Management) cause code value 81, Invalid PTI value
- 1027 - Semantically incorrect message, 3GPP equivalent ESM(EPS Session Management) cause code value 95, Semantically incorrect message
- 1028 - Invalid mandatory information, 3GPP equivalent ESM(EPS Session Management) cause code value 96, Invalid mandatory information
- 1029 - Message type non-existent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 97, Message type non-existent or not implemented
- 1030 - Message not compatible with state, 3GPP equivalent ESM(EPS Session Management) cause code value 98, Message type not compatible with the protocol state
- 1031 - Information element nonexistent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 99, Information element non-existent or not implemented
- 1032 - Conditional information element error, 3GPP equivalent ESM(EPS Session Management) cause code value 100, Conditional IE error
- 1033 - Message not compatible with protocol state, 3GPP equivalent ESM(EPS Session Management) cause code value 101, Message not compatible with the protocol state
- 1034 - APN restriction value incompatible with active PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 112, APN restriction value incompatible with
 - active EPS bearer context
- 1035 - No GPRS context present
- 1036 - Requested feature not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 40, Feature not supported
- 1037 - Illegal MS, 3GPP equivalent EMM(EPS Mobility Management) cause code value 3, Illegal UE (MS)

- 1038 - Illegal ME, 3GPP equivalent EMM(EPS Mobility Management) cause code value 6, Illegal ME. This error code is sent to the MS if the ME used is not acceptable
 - to the network, e.g. blacklisted
- 1039 - GPRS and non GPRS services not allowed
- 1040 - GPRS services not allowed
- 1041 - MS identity not derived by the network, 3GPP equivalent EMM(EPS Mobility Management) cause code value 9, UE (MS) Identify cannot be derived by the network
- 1042 - Implicitly detached, 3GPP equivalent EMM(EPS Mobility Management) cause code value 10, Implicitly Detached
- 1043 - PLMN not allowed, 3GPP equivalent EMM(EPS Mobility Management) cause code value 11, PLMN not allowed
- 1044 - LA not allowed, this cause is sent to the MS if it requests location updating in a location area where the HPLMN determines that the MS, by subscription, is not allowed to operate.
- 1045 - GPRS services not allowed in PLMN
- 1046 - PDP duplicate
- 1047 - UE radio access technology change
- 1048 - app preempted
- 1049 - Congestion, This cause is sent if the service request or LOCATION UPDATING REQUEST message cannot be actioned because of congestion (e.g. congestion of the MSC or SGSN or GGSN or PDN Gateway; no channel; facility busy/congested etc.).
- 1050 - No PDP context activated
- 1051 - Access class DSAC rejection

9.9.2.4 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

9.9.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

9.9.2.6 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

9.9.2.7 Internal call end reasons (Type=2)

- 200 - INTERNAL MIN, internal error table offset value, no meaningful message to the error.
- 201 - INTERNAL ERROR, this error code is returned when data call is brought down due to some unspecified internal error
- 202 - CALL ENDED
- 203 - INTERNAL UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 204 - UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 205 - CLOSE IN PROGRESS
- 206 - NETWORK INITIATED TERMINATION
- 207 - APP PREEMPTED
- 208 - ERR PDN IPV4 CALL DISALLOWED, this error code is returned when V4 PDN is in throttled state due to network providing only V6 address during the previous VSNCP bring up (subs_limited_to_v6). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 209 - ERR PDN IPV4 CALL THROTTLED, this error code is returned when V4 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 210 - ERR PDN IPV6 CALL DISALLOWED, this error code is returned when V6 PDN is in throttled state due to network providing only V4 address during the previous VSNCP bring up (subs_limited_to_v4). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 211 - ERR PDN IPV6 CALL THROTTLED, this error code is returned when V6 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 212 - MODEM RESTART
- 213 - PDP PPP NOT SUPPORTED
- 214 - UNPREFERRED RAT, this error code is returned when data call is brought down since the RAT on which the data call is attempted/connected is no longer the preferred RAT
- 215 - PHYS LINK CLOSE IN PROGRESS, this error code is returned when data call bring up is rejected because physical link is in the process of cleanup
- 216 - APN PENDING HANDOVER, this error code is returned when interface bring up is attempted for an APN that is yet to be handed over to target RAT
- 217 - PROFILE BEARER INCOMPATIBLE
- 218 - MMGSDI CARD EVT, this error code is returned when data call is brought down because card got refreshed/removed
- 219 - LPM OR PWR DOWN, this error code is returned when data call is brought down because device is going into lower power mode or powering down
- 220 - APN DISABLED, this error code is returned when APN is disabled in card

- 221 - MPIT EXPIRED, this error code is returned when data call is brought down because maximum PPP inactivity timer expired
- 222 - IPV6 ADDR TRANSFER FAILED
- 223 - TRAT SWAP FAILED
- 224 - EHRPD TO HRPD FALLBACK, this error code is returned when data call is brought down because device falls back from eHRPD to HRPD (not because of OOS on eHRPD but due to operator/spec driven eHRPD to HRPD fallback requirements)
- 225 - MANDATORY APN DISABLED, this error code is returned when any mandatory APN is disabled, and MinApnList Disallow call config item is set to TRUE in device
- 226 - MIP CONFIG FAILURE, this error code is returned when UE is in MIP Only config (QCMIP=2) but MIP config fails on call bring up due to incorrect provisioning

9.9.2.8 Call Manager defined call end reasons (Type=3)

- 500 - CDMA LOCK, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to device in CDMA locked state
- 501 - INTERCEPT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an intercept order from the base station
- 502 - REORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a reorder from base station
- 503 - REL SO REJ, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with reason: SO Reject
- 504 - INCOM CALL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an incoming call from base station
- 505 - ALERT STOP, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RL/FL fade (or) receiving call release from base stations
- 506 - ACTIVATION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to channel acquisition failures. This indicates that device has failed acquiring all the channels in the PRL
- 507 - MAX ACCESS PROBE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes transmitted
- 508 - CCS NOT SUPPORTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since concurrent service is not supported by base station
- 509 - NO RESPONSE FROM BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since there is no response received from base station
- 510 - REJECTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to base station rejecting the call
- 511 - INCOMPATIBLE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since concurrent services requested were not compatible
- 512 - ALREADY IN TC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since traffic channel is already up for voice calls
- 513 - USER CAL ORIG DURING GPS
- 514 - USER CAL ORIG DURING SMS, this error code is returned when data call is brought down because traffic channel request got rejected since SMS is ongoing

- 515 - NO CDMA SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have CDMA service
- 516 - MC ABORT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since MC aborted the origination/conversation
- 517 - PSIST NG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to persistence test failure
- 518 - UIM NOT PRESENT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RUIM not present
- 519 - RETRY ORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a retry order from base station
- 520 - ACCESS BLOCK, this error code is returned when data call is brought down because traffic channel rejected/released due to Access blocked by base station
- 521 - ACCESS BLOCK ALL, this error code is returned when data call is brought down because traffic channel rejected due to Access blocked by the base station for all mobile devices
- 522 - IS707B MAX ACC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes for IS-707B call
- 523 - THERMAL EMERGENCY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) to put device in thermal emergency
- 524 - CALL ORIG THROTTLED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since call origination is throttled by DCTM module
- 525 - USER CALL ORIGINATE DURING VOICE CALL, this error code is returned when data call is brought down because traffic channel got released by CM(Call Manager) in favor of voice call or SMS when concurrent voice and data are not supported
- 1000 - CONFERENCE FAILED
- 1001 - INCOMING REJECTED
- 1002 - NO GATEWAY SERVICE
- 1003 - NO GPRS CONTEXT
- 1004 - ILLEGAL MS, This cause is sent to the MS when the network refuses service to the MS either because an identity of the MS is not acceptable to the network or because the MS does not pass the authentication check, i.e. the SRES received from the MS is different from that generated by the network
- 1005 - ILLEGAL ME, This cause is sent to the MS if the ME used is not acceptable to the network, e.g. blacklisted
- 1006 - GPRS SERVICES AND NON GPRS SERVICES NOT ALLOWED
- 1007 - GPRS SERVICES NOT ALLOWED
- 1008 - MS IDENTITY CANNOT BE DERIVED BY THE NETWORK
- 1009 - IMPLICITLY DETACHED, this error code is sent to the MS either if the network has implicitly detached the MS, e.g. some while after the Mobile reachable timer has expired, or if the GMM context data related to the subscription does not exist in the SGSN e.g. because of a SGSN restart.
- 1010 - PLMN NOT ALLOWED, this error code is sent to the MS if it requests location updating in a PLMN where the MS, by subscription or due to operator determined barring is not allowed to operate
- 1011 - LOCAL AREA NOT ALLOWED
- 1012 - GPRS SERVICES NOT ALLOWED IN THIS PLMN
- 1013 - PDP DUPLICATE

- 1014 - USER EQUIPMENT RADIO ACCESS TECHNOLOGY CHANGE
- 1015 - CONGESTION
- 1016 - NO PDP CONEXT ACTIVATED
- 1017 - ACCESS CLASS DSAC REJECTION
- 1018 - PDP ACTIVATE MAX RETRY FAILED
- 1019 - RAB FAILURE
- 1020 - EPS SERVICE NOT ALLOWED
- 1021 - TRACKING AREA NOT ALLOWED
- 1022 - ROAMING NOT ALLOWED IN THIS TRACKING AREA
- 1023 - NO SUITABLE CELLS IN TRACKING AREA
- 1024 - NOT AUTHORIZED FOR THIS CLOSED SUBSCRIBER GROUP
- 1025 - ESM UNKNOWN EPS BEARER CONTEXT
- 1026 - DRB RELEASED AT RRC
- 1027 - NAS SIG CONN RELEASED
- 1028 - EPS MOBILITY MANAGEMENT DETACHED
- 1029 - EPS MOBILITY MANAGEMENT ATTACH FAILED
- 1030 - EPS MOBILITY MANAGEMENT ATTACH STARTED
- 1031 - LTE NAS SERVICE REQ FAILED
- 1032 - ESM(EPS Session Management) ACTIVE DEDICATED BEARER REACTIVATED BY NW
- 1033 - ESM(EPS Session Management) LOWER LAYER FAILURE
- 1034 - ESM(EPS Session Management) SYNC UP WITH NW
- 1035 - ESM(EPS Session Management) NW ACTIVATED DED BEARER WITH ID OF DEF BEARER
- 1036 - ESM(EPS Session Management) BAD OTA MESSAGE
- 1037 - ESM DS REJECTED THE CALL
- 1038 - ESM(EPS Session Management) CONTEXT TRANSFERRED DUE TO IRAT
- 1039 - DS EXPLICIT DEACT
- 1040 - ESM(EPS Session Management) LOCAL CAUSE NONE
- 1041 - LTE NAS SERVICE REQ FAILED NO THROTTLE
- 1042 - ACL FAILURE, This error code should rarely triggered and reported to the application
- 1043 - LTE NAS SERVICE REQ FAILED DS DISALLOW
- 1044 - EMM(EPS Mobility Management) T3417 EXPIRED
- 1045 - EMM(EPS Mobility Management) T3417 EXT EXPIRED
- 1046 - LRR(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRR(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRR(LTE Radio Resource Control) UL DATA CNF FAILURE CONN REL

- 1049 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE RLF
- 1050 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CTRL NOT CONN
- 1051 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE
- 1052 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE ABORTED
- 1053 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE ACCESS BARRED
- 1054 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRRC(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRRC(LTE Radio Resource Control) CONN REL RLF
- 1063 - LRRRC(LTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRRC(LTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRRC(LTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRRC(LTE Radio Resource Control) CONN REL SIB READ ERROR
- 1067 - DETACH WITH REATTACH LTE NW DETACH
- 1068 - DETACH WITH OUT REATTACH LTE NW DETACH
- 1069 - ESM(EPS Session Management) PROC TIME OUT
- 1070 - MESSAGE EXCEED MAX L2 LIMIT
- 1500 - CD GEN OR BUSY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of general or network busy
- 1501 - CD BILL OR AUTH, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of billing failure or authentication failure
- 1502 - CHG HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is a change to HDR system due to redirection or PRL not preferred
- 1503 - EXIT HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device exited HDR due to redirection or PRL not preferred
- 1504 - HDR NO SESSION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have a HDR session
- 1505 - HDR ORIG DURING GPS FIX, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since it is ending an HDR call origination in favor of a GPS fix
- 1506 - HDR CS TIMEOUT, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since connection setup on HDR system timed out

- 1507 - HDR RELEASED BY CM, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when it wants to release a HDR call so a 1X call can continue
- 1508 - COLLOC ACQ FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when device failed to acquire co-located HDR for origination
- 1509 - OTASP COMMIT IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since an OTASP commit is in progress
- 1510 - NO HYBR HDR SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device has no Hybrid HDR service
- 1511 - HDR NO LOCK GRANTED, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module could not get the RF lock
- 1512 - HOLD OTHER IN PROG, this error code is returned when data call is brought down by CM(Call Manager) because DBM or SMS is in progress
- 1513 - HDR FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module released the call due to fade
- 1514 - HDR ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to HDR system Access Failure
- 2000 - CLIENT END, this error code is returned when client ends the data call
- 2001 - NO SRV, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device has no service
- 2002 - FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device lost the system due to fade
- 2003 - REL NORMAL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with no reason
- 2004 - ACC IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access attempt already in progress
- 2005 - ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access Failure
- 2006 - REDIR OR HANDOFF, this error code is returned when data call is brought down because device is in the process of redirecting/handing off to a different target system
- 2500 - OFFLINE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device went offline
- 2501 - EMERGENCY MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device is operating in Emergency mode
- 2502 - PHONE IN USE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device is in use (e.g voice call)
- 2503 - INVALID MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the device's operational mode is different from the mode requested in the traffic channel bring up
- 2504 - INVALID SIM STATE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the SIM was marked by network as invalid for circuit and/or packet service domain
- 2505 - NO COLLOC HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is no collocated HDR
- 2506 - CALL CONTROL REJECTED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since Call control module rejected the request

9.9.2.9 3GPP specification defined call end reasons (Type=6)

- 8 - OPERATOR DETERMINED BARRING, this reason code is posted by the MME(Mobility Management Entity) to indicate operator has barred the UE
- 25 - LLC SNDSCP FAILURE, PDP context deactivation initiated by the MS or by the Network
- 26 - INSUFFICIENT RESOURCES, this reason is posted to indicate that the network cannot provide the requested service due to insufficient resources
- 27 - MISSING OR UNKNOWN APN, the APN was required and not specified or APN could not be resolved. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 28 - UNKNOWN PDN TYPE, the reason is posted by the network to indicate that the PDN type was not recognized
- 29 - AUTH FAILED, the reason is posted when authentication fails. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 30 - GGSN REJECT, the reason is posted when the request was rejected by Serving GW or PDN GW. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 31 - ACTIVATION REJECT, the reason is posted when the request is rejected by the network due to unspecified reasons
- 32 - OPTION NOT SUPPORTED, the reason is posted when UE requested a service not supported by the PLMN
- 33 - OPTION UNSUBSCRIBED, This cause is sent when the MS requests a service option for which it has no subscription
- 34 - OPTION TEMP OOO, service option temporarily out of order, this reason is posted when the network is temporarily out of resources to service the request
- 35 - PTI ALREADY USED, the reason is posted to indicate that PTI (Procedure Transaction Identifier) used in the request is already active via another UE requested procedure
- 36 - REGULAR DEACTIVATION, this reason is posted by the network to initiate a regular release of bearer resources
- 37 - EPS QOS NOT ACCEPTED, this reason is posted by the network to indicate that the QoS requested by the UE could not be accepted
- 38 - NETWORK FAILURE, this reason is posted when an error occurs in the network
- 39 - UMTS REACTIVATION REQ, this reason is posted by the network to request for bearer reactivation. This code may be posted during network congestion
- 40 - FEATURE NOT SUPPORTED, Unsuccessful MBMS context activation requested by the network
- 41 - TFT SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying TFT operation included in the request
- 42 - TFT SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying TFT operation included in the request
- 43 - UNKNOWN PDP CONTEXT, the reason is posted when the bearer identity (or linked bearer identity) in the request is invalid (or inactive)
- 44 - FILTER SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying packet filter(s) associated with a TFT
- 45 - FILTER SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying packet filter(s) associated with a TFT

- 46 - PDP WITHOUT ACTIVE TFT, the reason is posted by the network when UW requested more than one PDP connection without TFT
- 50 - IPV4 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 50, PDN type IPv4 only allowed.
- 51 - IPV6 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 51, PDN type IPv6 only allowed
- 52 - SINGLE ADDRESS BEARER ONLY, 3GPP equivalent ESM(EPS Session Management) cause code value 52, Single address bearers only allowed. The reason is posted when the network supports single address bearers only, meaning dual IP bearers are not supported
- 53 - ESM INFORMATION NOT RECEIVED, 3GPP equivalent ESM(EPS Session Management) cause code value 53, ESM information not received. The reason is posted by the network to indicate that the PDN connection request was rejected because ESM information was not received
- 54 - PND CONNECTION DOES NOT EXIST, 3GPP equivalent ESM(EPS Session Management) cause code value 54, PDN connection does not exist The reason is posted by the network during handover from a non-3G-PP network to indicate that the MME does not have any information regarding the requested PDN connection
- 55 - MULTIPLE CONNECTION TO SAME PDN NOT ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 55, Multiple PDN connections for a given APN not allowed. The reason is posted by the network to indicate that the UE is already connected to the requested APN via another PDN/PDN connection
- 81 - INVALID TRANSACTION ID, the reason is posted by the network to indicate that the PTI used in the request is unassigned or reserved
- 95 - MESSAGE INCORRECT SEMANTIC, the reason is posted by the network to indicate receipt of an invalid message
- 96 - INVALID MANDATORY INFO, the reason is posted by the network to indicate receipt of a message with semantic error in a mandatory information element
- 97 - MESSAGE TYPE UNSUPPORTED, the reason is posted by the network to indicate receipt of a message that is either undefined or defined but not implemented by the equipment sending this ESM cause
- 98 - MSG TYPE NONCOMPATIBLE STATE, the reason is posted by the network to indicate receipt of a message type that cannot be handled in the current network protocol state
- 99 - UNKNOWN INFO ELEMENT, the reason is posted by the network to indicate receipt of a message that includes an information element that is either not defined or defined but not implemented by the equipment sending the ESM cause
- 100 - CONDITIONAL IE ERROR, the reason is posted by the network to indicate receipt of a message that includes a syntactically incorrect information element. This message is ignored by the network.
- 101 - MSG AND PROTOCOL STATE UNCOMPATIBLE, the reason is posted by the network to indicate receipt of a message that cannot be handled in the current network protocol state
- 111 - PROTOCOL ERROR, the reason is posted by the network to indicate a protocol error when no other error applies
- 112 - APN TYPE CONFLICT
- 113 - INVALID PROXY-CALL SESSION CONTROL FUNCTION ADDRESS

9.9.2.10 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

9.9.2.11 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

9.9.2.12 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

9.10 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

9.10.1 Detailed Description

Carrier Codes table.

9.10.2 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

9.11 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

Macros

- `#define __GOBI_API_CODING_SCHEME_H__`

9.11.1 Detailed Description

Data Coding Scheme.

9.11.2 Call Control Result Reasons (Value - Name - Description)

9.11.2.1 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

9.11.3 Coding Group Bits 7..4(0001)

9.11.3.1 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.

9.11.4 Coding Group Bits 7..4(0010)

9.11.4.1 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

9.11.5 Coding Group Bits 7..4(0011)

9.11.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

9.11.6 Coding Group Bits 7..4(01xx)

9.11.6.1 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

9.11.7 Coding Group Bits 7..4(1001)

9.11.7.1 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

9.11.8 Coding Group Bits 7..4(1010..1101)

9.11.8.1 Reserved coding groups

9.11.9 Coding Group Bits 7..4(1110)

9.11.9.1 Defined by the WAP Forum

9.11.10 Coding Group Bits 7..4 (1111)

9.11.10.1 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

9.11.11 Macro Definition Documentation

9.11.11.1 `#define __GOBI_API_CODING_SCHEME_H__`

9.12 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

9.12.1 Detailed Description

Position Determination Service API GPS Capability Codes.

9.12.2 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

9.13 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

9.13.1 Detailed Description

Device Management Service API Power Modes table.

9.13.2 Power Modes (Value - Description)

- 0 - Online (default)
- 1 - Low power (airplane) mode
- 2 - Factory test mode
- 3 - Offline
- 4 - Reset
- 5 - Power off
- 6 - Persistent low power (airplane) mode
- 7 - Mode - only low power

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.14 qaGobiApiTableRadiolInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

9.14.1 Detailed Description

Network Access Service API Radio Interfaces table.

9.14.2 Radio interface

9.14.2.1 Technology (Value - Radio Interface Technology)

- 0 - No service
- 1 - CDMA 1xRTT
- 2 - CDMA 1xEV-DO
- 3 - AMPS (Unsupported)
- 4 - GSM
- 5 - UMTS
- 6 - WLAN
- 7 - GPS
- 8 - LTE

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.15 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

9.15.1 Detailed Description

Region Codes table.

9.15.2 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

9.16 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

9.16.1 Detailed Description

Voice Service Options.

9.16.2 Service Option codes (Code - Reason)

9.16.2.1 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

9.17 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

9.17.1 Detailed Description

Voice Supplementary Service Information Classes.

9.17.2 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

9.18 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

9.18.1 Detailed Description

Swi Audio related tables.

9.18.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

9.18.3 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

9.19 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

9.19.1 Detailed Description

Update Complete Status table.

9.19.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)

- 200 - Successful - The request has succeeded
- 250-299 - Successful(vendor specified) - successful operation with vendor specified ResultCode
- 400 - Management Client Error - Management Client error - based on User or Device behavior
- 401 - User Cancelled - User chose not to accept the operation when prompted
- 402 - Corrupted Firmware Update Package - Corrupted firmware update package did not store correctly. Detected for example, by mismatch CRCs between actual and expected
- 403 - Firmware UpdatePackage(Device Mismatch) - Wrong firmware update package delivered to device based on current device characteristics
- 404 - Failed Firmware Update Package Validation - Failure to positively validate digital signature of firmware update package
- 405 - Firmware Update Package Not acceptable - firmware update package is not acceptable
- 406 - Alternate Download Authentication Failure - authentication required but authentication failure was encountered when downloading firmware update package
- 407 - Alternate Download Request Timeout - client has encountered a timeout when downloading firmware update package
- 408 - Not Implemented - the device does not support the requested operation
- 409 - Undefined Error - indicates failure not defined by any other error code
- 410 - Firmware Update Failed - firmware update operation failed in device
- 411 - Malformed or Bad URL - the URL provided for alternate download is bad
- 412 - Alternate Download Server Unavailable - the alternate download server is unavailable or does not respond
- 450 - Client Error (OMADM General) - Vendor defined client error
- 451 - Client Error (OMADM SyncML) - Vendor defined client error
- 452 - Client Error (OMADM Auth) - Vendor defined client error
- 453 - Client Error (OMADM Protocol) - Vendor defined client error
- 454 - Client Error (OMADM Tree) - Vendor defined client error
- 455 - Client Error (OMADM DStore) - Vendor defined client error
- 456 - Client Error (OMADM Trigger) - Vendor defined client error
- 457 - Client Error (OMADM Fumo) - Vendor defined client error
- 458 - Client Error (OMADM Comms) - Vendor defined client error
- 459 - Client Error (OMADM Parse) - Vendor defined client error
- 460 - Client Error (OMADM TNDS) - Vendor defined client error
- 461 - Client Error (OMADM SCM) - Vendor defined client error
- 462 - Client Error (OMADM Impl) - Vendor defined client error

- 463-499 - Client Error (Vendor Specified) - client error encountered for operation with vendor specified result code
- 500 - Alternate Download Server Error - Alternate download server error encountered
- 501 - Download fails due to device out of memory - The download fails due to insufficient memory in the device to save the firmware update package
- 502 - Firmware update fails due to device out of memory - The update fails because there isn't sufficient memory to update the device
- 503 - Download fails due to network issues - The download fails due to network/transport level errors
- 550-599 - Alternate Download Server Error (vendor specified)- Alternate download server error encountered for operation with vendor specified result code

Copyright: © 2013 Sierra Wireless, Inc. all rights reserved

9.20 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

9.20.1 Detailed Description

Voice Service Call and supplementary services end reasons.

9.20.2 Voice Call and supplementary services end reason codes (Code - Reason)

9.20.2.1 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

9.20.2.2 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

9.20.2.3 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

9.20.2.4 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

9.20.2.5 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

9.20.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

9.20.2.7 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

9.20.2.8 reject causes

- 237 - Service option is not supported

9.20.2.9 IP end reasons

- 300 - Received SIP 400 bad request;waiting for INVITE response
- 301 - Received SIP 400 bad request;waiting for INVITE response
- 302 - Received SIP 404 not found; call failed; called party does not exist
- 303 - Received SIP 415 unsupported media type; call failed; called party does not support media
- 304 - Received SIP 480 temporarily unavailable; call failed; called party is not in the LTE area
- 305 - No network response; call failed
- 306 - No network response; unable to put call on hold
- 307 - Moved to eHRPD; call failed or dropped; not in the LTE area
- 308 - Upgrade/downgrade rejected (200 OK with the current call SDP)
- 309 - Received 403 call forbidden; waiting for INVITE response
- 310 - Generic timeout; did not receive a response from the server or other end
- 311 - Reported on the MO side for generic internal software errors; user can try again if the call still exists
- 312 - Reported on the MT side if the upgrade timer has been cancelled or cannot complete the request for some reason after notifying the user of a re-invite request
- 313 - Call origination is rejected due to a Service-Specific Access Control (SSAC) barring
- 314 - Phone was put in thermal emergency
- 315 - 1XCSFB call ended because of a soft failure
- 316 - 1XCSFB call ended because of a hard failure

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

9.21 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_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_INVALID_SERVICE_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 }

```

9.21.1 Enumeration Type Documentation

9.21.1.1 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_ENUM_END End of SLQS SDK specific error codes
eQCWWAN_ERR_QMI_OFFSET 1000 - This is not an error code but the offset from which mapped QMI error codes start from
eQCWWAN_ERR_QMI_MALFORMED_MSG 1001 - Malformed or Corrupted QMI msg
eQCWWAN_ERR_QMI_NO_MEMORY 1002 - Device could not allocate memory for QMI Resp
eQCWWAN_ERR_QMI_INTERNAL 1003 - Unexpected error occurred during processing
eQCWWAN_ERR_QMI_ABORTED 1004 - Processing aborted
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED 1005 - QMI client IDs have been exhausted
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION 1006 - Unable to abort QMI transaction
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID 1007 - Invalid QMI client ID
eQCWWAN_ERR_QMI_NO_THRESHOLDS 1008 - No thresholds were provided
eQCWWAN_ERR_QMI_INVALID_HANDLE 1009 - Invalid Handle provided in the QMI request
eQCWWAN_ERR_QMI_INVALID_PROFILE 1010 - Profile specified is invalid
eQCWWAN_ERR_QMI_INVALID_PINID 1011 - Invalid PIN ID specified
eQCWWAN_ERR_QMI_INCORRECT_PIN 1012 - Incorrect PIN ID specified
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND 1013 - No network found
eQCWWAN_ERR_QMI_CALL_FAILED 1014 - Call failed
eQCWWAN_ERR_QMI_OUT_OF_CALL 1015 - Device is not in a call
eQCWWAN_ERR_QMI_NOT_PROVISIONED 1016 - Requested information element not provisioned on device
eQCWWAN_ERR_QMI_MISSING_ARG 1017 - Mandatory QMI TLV not provided
eQCWWAN_ERR_QMI_ARG_TOO_LONG 1019 - Arg passed in QMI TLV larger than available storage in device
eQCWWAN_ERR_QMI_INVALID_TX_ID 1022 - Invalid TX ID specified
eQCWWAN_ERR_QMI_DEVICE_IN_USE 1023 - Device currently in a call
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED 1024 - The selected operation is not supported by the network
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED 1025 - The selected operation is not supported by the device
eQCWWAN_ERR_QMI_NO_EFFECT 1026 - Requested operation would have no effect
eQCWWAN_ERR_QMI_NO_FREE_PROFILE 1027 - No space for a profile is available
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE 1028 - Invalid PDP type specified
eQCWWAN_ERR_QMI_INVALID_TECH_PREF 1029 - Invalid technology preference specified
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE 1030 - Invalid profile type specified
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE 1031 - Invalid service type specified

eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION 1032 - Invalid register action specified

eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION 1033 - Invalid PS attach/detach action specified

eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED 1034 - Authentication of supplied information element failed

eQCWWAN_ERR_QMI_PIN_BLOCKED 1035 - PIN is blocked; an unblock operation needs to be issued

eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED 1036 - PIN is permanently blocked; the UIM is unusable

eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED 1037 - UIM initialization has not completed

eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE 1038 - Max QOS requests are used

eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER 1039 - The Flow filter is incorrect

eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE 1040 - Network unaware of the QOS requested

eQCWWAN_ERR_QMI_INVALID_ID 1041 - Invalid QOS ID

eQCWWAN_ERR_QMI_INVALID_QOS_ID 1041 - Invalid QOS ID

eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED 1042 - The request number is not supported

eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND 1043 - Unable to find the interface

eQCWWAN_ERR_QMI_FLOW_SUSPENDED 1044 - Flow suspended

eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT 1045 - Data format is invalid

eQCWWAN_ERR_QMI_GENERAL 1046 - General error

eQCWWAN_ERR_QMI_UNKNOWN 1047 - Unknown error

eQCWWAN_ERR_QMI_INVALID_ARG 1048 - A specified argument is invalid

eQCWWAN_ERR_QMI_INVALID_INDEX 1049 - A specified index is invalid

eQCWWAN_ERR_QMI_NO_ENTRY 1050 - No information element exists at specified memory designation

eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL 1051 - The memory storage specified in the request is full

eQCWWAN_ERR_QMI_DEVICE_NOT_READY 1052 - Device not in a ready state

eQCWWAN_ERR_QMI_NETWORK_NOT_READY 1053 - Network not in a ready state

eQCWWAN_ERR_QMI_CAUSE_CODE 1054 - Error provided in SMS cause code

eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT 1055 - The message could not be sent

eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE 1056 - The message could not be delivered

eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID 1057 - The message ID specified for the message is invalid

eQCWWAN_ERR_QMI_ENCODING 1058 - The message is not encoded properly

eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK 1059 - Maximum number of authentication failures has been reached

eQCWWAN_ERR_QMI_INVALID_TRANSITION 1060 - Operating mode transition from the current mode is invalid

eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE 1061 - The interface is not multicast

eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE 1062 - Maximum requests in use

eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE 1063 - Invalid multicast handle

eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF 1064 - Invalid IP family preference

eQCWWAN_ERR_QMI_SESSION_INACTIVE 1065 - No tracking session has been started

eQCWWAN_ERR_QMI_SESSION_INVALID 1066 - Current session does not allow this operation

eQCWWAN_ERR_QMI_SESSION_OWNERSHIP 1067 - Current tracking session not started by this QMI control point

eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES 1068 - Device GPS service resources insufficient for request

eQCWWAN_ERR_QMI_DISABLED 1069 - Device GPS service disabled

eQCWWAN_ERR_QMI_INVALID_OPERATION 1070 - Invalid operation specified

eQCWWAN_ERR_QMI_INVALID_QMI_CMD 1071 - Invalid/unknown QMI command specified

eQCWWAN_ERR_QMI_TPDU_TYPE 1072 - Message contains TPDU type that cannot be read as raw message

eQCWWAN_ERR_QMI_SMSC_ADDR 1073 - The SMSC address specified is invalid

eQCWWAN_ERR_QMI_INFO_UNAVAILABLE 1074 - Information element is unavailable at this point

eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG 1075 - Segment size too large

eQCWWAN_ERR_QMI_SEGMENT_ORDER 1076 - Segment order is incorrect

eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED 1077 - Bundling not supported

eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE 1078 - The operation failed partially

eQCWWAN_ERR_QMI_POLICY_MISMATCH 1079 - Policy mismatch

eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND 1080 - SIM file not found

eQCWWAN_ERR_QMI_EXTENDED_INTERNAL 1081 - Extended internal error

eQCWWAN_ERR_QMI_ACCESS_DENIED 1082 - Access to a required entity is not available

eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED 1083 - Selected operating mode is invalid with current hardware setting

eQCWWAN_ERR_QMI_ACK_NOT_SENT 1084 - ACK not sent

eQCWWAN_ERR_QMI_INJECT_TIMEOUT 1084 - Inject a timeout for the request

eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE 1090 - Incompatible state

eQCWWAN_ERR_QMI_FDN_RESTRICT 1091 - FDN Restrict

eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE 1092 - SUPS failure cause

eQCWWAN_ERR_QMI_NO_RADIO 1093 - No Radio

eQCWWAN_ERR_QMI_NOT_SUPPORTED 1094 - Not Supported

eQCWWAN_ERR_QMI_NO_SUBSCRIPTION 1095 - No Subscription

eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED 1096 - Card call control failed

eQCWWAN_ERR_QMI_NETWORK_ABORTED 1097 - Network Aborted

eQCWWAN_ERR_QMI_MSG_BLOCKED 1098 - Open Error

eQCWWAN_ERR_QMI_MAX Error - End of QMI specific defines

eQCWWAN_ERR_SWICM_START Vendor defines - **Connection Manager error codes**

eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED 0xE001 - The API is yet to be implemented

eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED 0xE002 - The service is not supported

eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED 0xE003 - The client is not supported

eQCWWAN_ERR_SWICM_TIMEOUT 0xE004 - API Timeout

eQCWWAN_ERR_SWICM_SOCKET_IN_USE 0xE005 - The communication socket is in use

eQCWWAN_ERR_SWICM_AM_VERS_ERROR 0xE006 - SLQS API and SDK version mismatch

eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS 0xE007 - Failed to kill SDK process

eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS 0xE008 - Call in progress

eQCWWAN_ERR_SWICM_V4DWN_V6DWN 0xE009 - IPV4 and IPV6 is down

eQCWWAN_ERR_SWICM_V4DWN_V6UP 0xE00A - IPV4 is down and IPV6 is up

eQCWWAN_ERR_SWICM_V4UP_V6DWN 0xE00B - IPV4 is up and IPV6 is down

eQCWWAN_ERR_SWICM_V4UP_V6UP 0xE00C - IPV4 and IPV6 is up

eQCWWAN_ERR_SWICM_INVALID_SESSION_ID 0xE00D - Invalid V4 Session ID

eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID 0xE00E - Invalid V4 Session ID

eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID 0xE00F - Invalid V6 Session ID

eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS 0xE010 - No available Session Manager slots for additional data sessions

eQCWWAN_ERR_SWICM_END 0xE011 - End of connection manager specific codes

eQCWWAN_ERR_SWISMS_START Vendor defines - SMS Error codes

eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG 0xE101 - SMS message length is long

eQCWWAN_ERR_SWISMS_MSG_CORRUPTED 0xE102 - The SMS message is corrupted (encoding wrong)

eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED 0xE103 - The SMS number is corrupted (incorrect number)

eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND 0xE104 - The SMS bearer data is not available

eQCWWAN_ERR_SWISM_END

eQCWWAN_ERR_SWIIM_START Vendor defines - Image Management error codes

eQCWWAN_ERR_SWIIM_INVALID_PATH 0xE801 - Invalid directory path

eQCWWAN_ERR_SWIIM_OPENING_DIR 0xE802 - Unable to open the directory

eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND 0xE803 - No Firmware image present in the path

eQCWWAN_ERR_SWIIM_OPENING_FILE 0xE804 - Unable to open the file

eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE 0xE805 - Firmware image is corrupted

eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED 0xE806 - No Firmware image download needed

eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL 0xE807 - Firmware update failed

eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH 0xE808 - Update success but pri/fw preference mismatch

eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS 0xE809 - Update successful

eQCWWAN_ERR_SWIIM_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_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

9.21.1.2 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

9.22 qos.h File Reference

Data Structures

- struct [unpack_qos_SLQSQosGetNetworkStatus_t](#)
- struct [pack_qos_SLQSQosSwiReadApnExtraParams_t](#)
- struct [unpack_qos_SLQSQosSwiReadApnExtraParams_t](#)

- struct [pack_qos_SLQSQosSwiReadDataStats_t](#)
- struct [unpack_QosFlowStat_t](#)
- struct [unpack_qos_SLQSQosSwiReadDataStats_t](#)
- struct [unpack_qos_SLQSSetQosNWStatusCallback_ind_t](#)
- struct [unpack_qos_SLQSSetQosStatusCallback_ind_t](#)
- struct [unpack_qos_SLQSSetQosPriEventCallback_ind_t](#)
- struct [pack_qos_SLQSSetQosEventCallback_t](#)
- struct [unpack_qos_QosFlowInfoState_t](#)
- struct [unpack_qos_dataRate_t](#)
- struct [unpack_qos_tokenBucket_t](#)
- struct [unpack_qos_pktErrRate_t](#)
- struct [unpack_qos_swiQosFlow_t](#)
- struct [unpack_qos_IPv4Addr_t](#)
- struct [unpack_qos_Tos_t](#)
- struct [unpack_qos_IPv6Addr_t](#)
- struct [unpack_qos_IPv6TrafCls_t](#)
- struct [unpack_qos_Port_t](#)
- struct [unpack_qos_swiQosFilter_t](#)
- struct [unpack_qos_QosFlowInfo_t](#)
- struct [unpack_qos_SLQSSetQosEventCallback_ind_t](#)

Macros

- `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`
- `#define LIBPACK_MAX_QOS_FILTERS 25`
- `#define LIBPACK_MAX_QOS_FLOWS 8`

Functions

- int [pack_qos_SLQSQosGetNetworkStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_qos_SLQSQosGetNetworkStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSQosGetNetworkStatus_t](#) *pOutput)
- int [pack_qos_SLQSQosSwiReadApnExtraParams](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_qos_SLQSQosSwiReadApnExtraParams_t](#) reqParam)
- int [unpack_qos_SLQSQosSwiReadApnExtraParams](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSQosSwiReadApnExtraParams_t](#) *pOutput)
- int [pack_qos_SLQSQosSwiReadDataStats](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_qos_SLQSQosSwiReadDataStats_t](#) reqParam)
- int [unpack_qos_SLQSQosSwiReadDataStats](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSQosSwiReadDataStats_t](#) *pOutput)
- int [unpack_qos_SLQSSetQosNWStatusCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosNWStatusCallback_ind_t](#) *pOutput)
- int [unpack_qos_SLQSSetQosStatusCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosStatusCallback_ind_t](#) *pOutput)
- int [unpack_qos_SLQSSetQosPriEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosPriEventCallback_ind_t](#) *pOutput)
- int [pack_qos_SLQSSetQosEventCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_qos_SLQSSetQosEventCallback_t](#) reqParam)
- int [unpack_qos_SLQSSetQosEventCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_qos_SLQSSetQosEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosEventCallback_ind_t](#) *pOutput)

9.22.1 Macro Definition Documentation

9.22.1.1 `#define LIBPACK_MAX_QOS_FILTERS 25`

9.22.1.2 `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`

9.22.1.3 `#define LIBPACK_MAX_QOS_FLOWS 8`

9.22.2 Function Documentation

9.22.2.1 `int pack_qos_SLQSQosGetNetworkStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Function to pack command to retrieve QoS status of the network. This maps to SLQSQosGetNetworkStatus

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> 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: 2 seconds
- Technology Supported: CDMA
- PDN Specific: No

9.22.2.2 `int pack_qos_SLQSQosSwiReadApnExtraParams (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSQosSwiReadApnExtraParams_t reqParam)`

Function to pack QMI command to query extra APN parameters This maps to SLQSQosSwiReadApnExtraParams

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> 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_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

9.22.2.3 `int pack_qos_SLQSQosSwiReadDataStats (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSQosSwiReadDataStats_t reqParam)`

Function to pack QMI command to query APN data statistics This maps to SLQSQosSwiReadDataStats

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> 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_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

9.22.2.4 `int pack_qos_SLQSSetQosEventCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSSetQosEventCallback_t reqParam)`

Function to pack QMI command to enable QoS event indications This maps to SLQSSetQosEventCallback

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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_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

9.22.2.5 `int unpack_qos_SLQSQosGetNetworkStatus (uint8_t * pResp, uint16_t respLen,
unpack_qos_SLQSQosGetNetworkStatus_t * pOutput)`

Function to unpack the response to get NW QoS status command This maps to SLQSQosGetNetworkStatus

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.22.2.6 int unpack_qos_SLQSQosSwiReadApnExtraParams (uint8_t * *pResp*, uint16_t *respLen*,
unpack_qos_SLQSQosSwiReadApnExtraParams_t * *pOutput*)

Function to unpack the response to get NW QoS status command This maps to SLQSQosSwiReadApnExtraParams

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.22.2.7 int unpack_qos_SLQSQosSwiReadDataStats (uint8_t * *pResp*, uint16_t *respLen*,
unpack_qos_SLQSQosSwiReadDataStats_t * *pOutput*)

Function to unpack APN data statistics response This maps to SLQSQosSwiReadDataStats

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.22.2.8 int unpack_qos_SLQSSetQosEventCallback (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack enable QoS event indications command's response This maps to SLQSSetQosEventCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • Response from modem
<i>respLen</i> [IN]	<ul style="list-style-type: none"> • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.22.2.9 int unpack_qos_SLQSSetQosEventCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosEventCallback_ind_t * *pOutput*)

Function to unpack QoS event indications This maps to SLQSSetQosEventCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.22.2.10 int unpack_qos_SLQSSetQosNWStatusCallback_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_qos_SLQSSetQosNWStatusCallback_ind_t * *pOutput*)

Function to unpack QoS NW status indication. This maps to SLQSSetQosNWStatusCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> 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_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

9.22.2.11 `int unpack_qos_SLQSSetQosPriEventCallback_ind (uint8_t * pResp, uint16_t respLen,
unpack_qos_SLQSSetQosPriEventCallback_ind_t * pOutput)`

Function to unpack QoS primary flow events. This maps to SLQSSetQosPriEventCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.22.2.12 `int unpack_qos_SLQSSetQosStatusCallback_ind (uint8_t * pResp, uint16_t respLen, unpack_qos_SLQSSetQosStatusCallback_ind_t * pOutput)`

Function to unpack QoS status indications. This maps to SLQSSetQosStatusCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.23 sms.h File Reference**Data Structures**

- struct [pack_sms_SLQSGetSMS_t](#)
- struct [unpack_sms_SLQSGetSMS_t](#)
- struct [pack_sms_SLQSGetSMSList_t](#)
- struct [qmiSmsMessageList](#)
- struct [unpack_sms_SLQSGetSMSList_t](#)
- struct [pack_sms_SLQSMModifySMSStatus_t](#)
- struct [unpack_sms_SLQSMModifySMSStatus_t](#)
- struct [pack_sms_SLQSDDeleteSMS_t](#)
- struct [unpack_sms_SLQSDDeleteSMS_t](#)
- struct [pack_sms_SendSMS_t](#)
- struct [unpack_sms_SendSMS_t](#)
- struct [pack_sms_SetNewSMSCallback_t](#)
- struct [unpack_sms_SetNewSMSCallback_t](#)
- struct [sMSMTMessage](#)
- struct [newMTMessageTlv](#)
- struct [sMSTransferRouteMTMessage](#)
- struct [transferRouteMessageTlv](#)
- struct [sMSMessageMode](#)
- struct [messageModeTlv](#)
- struct [sMSEtwsMessage](#)
- struct [sMSEtwsMessageTlv](#)
- struct [sMSEtwsPlmn](#)
- struct [eTWSPLMNInfoTlv](#)
- struct [sMSCAddress](#)
- struct [sMSCAddressTlv](#)
- struct [sMSOnIMS](#)
- struct [sMSOnIMSTlv](#)
- struct [unpack_sms_SetNewSMSCallback_ind_t](#)
- struct [unpack_sms_SLQSWmsMemoryFullCallBack_ind_t](#)

Macros

- [#define MAX_SMS_MESSAGE_SIZE 255](#)
- [#define MAX_SMS_LIST_SIZE 255](#)
- [#define MAX_MS_TRANSFER_ROUTE_MSG 256](#)
- [#define MAX_MSE_TWS_MSG 1254](#)
- [#define MAX_MSC_ADDRESS_SIZE 256](#)
- [#define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255](#)

Typedefs

- typedef struct [sMSMTMessage](#) [sMSMTMessageInfo](#)
- typedef struct [sMSTransferRouteMTMessage](#) [sMSTransferRouteMTMessageInfo](#)
- typedef struct [sMSMessageMode](#) [sMSMessageModelInfo](#)
- typedef struct [sMSEtwsMessage](#) [sMSEtwsMessageInfo](#)
- typedef struct [sMSEtwsPlmn](#) [sMSEtwsPlmnInfo](#)
- typedef struct [sMSCAddress](#) [sMSCAddressInfo](#)
- typedef struct [sMSONIMS](#) [sMSONIMSInfo](#)

Enumerations

- enum [eqmiCbKSetStatus](#) {
[LIBPACK_QMI_CBK_PARAM_RESET](#) = 0,
[LIBPACK_QMI_CBK_PARAM_SET](#) = 1,
[LIBPACK_QMI_CBK_PARAM_NOCHANGE](#) }

Functions

- int [pack_sms_SLQSGetSMS](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_sms_SLQSGetSMS_t](#) *reqParam)
- int [unpack_sms_SLQSGetSMS](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_sms_SLQSGetSMS_t](#) *pOutput)
- int [pack_sms_SLQSGetSMSList](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_sms_SLQSGetSMSList_t](#) *reqParam)
- int [unpack_sms_SLQSGetSMSList](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_sms_SLQSGetSMSList_t](#) *pOutput)
- int [pack_sms_SLQSModifySMSStatus](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_sms_SLQSModifySMSStatus_t](#) *reqParam)
- int [unpack_sms_SLQSModifySMSStatus](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_sms_SLQSModifySMSStatus_t](#) *pOutput)
- int [pack_sms_SLQSDeleteSMS](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_sms_SLQSDeleteSMS_t](#) *reqParam)
- int [unpack_sms_SLQSDeleteSMS](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_sms_SLQSDeleteSMS_t](#) *pOutput)
- int [pack_sms_SendSMS](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_sms_SendSMS_t](#) *reqParam)
- int [unpack_sms_SendSMS](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_sms_SendSMS_t](#) *pOutput)
- int [pack_sms_SetNewSMSCallback](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [pack_sms_SetNewSMSCallback_t](#) reqParam)
- int [unpack_sms_SetNewSMSCallback](#) ([uint8_t](#) *pResp, [uint16_t](#) respLen, [unpack_sms_SetNewSMSCallback_t](#) *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)

9.23.1 Macro Definition Documentation

9.23.1.1 `#define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255`

9.23.1.2 `#define MAX_MS_TRANSFER_ROUTE_MSG 256`

9.23.1.3 `#define MAX_MSC_ADDRESS_SIZE 256`

9.23.1.4 `#define MAX_MSE_TWS_MSG 1254`

9.23.1.5 `#define MAX_SMS_LIST_SIZE 255`

9.23.1.6 `#define MAX_SMS_MESSAGE_SIZE 255`

9.23.2 Typedef Documentation

9.23.2.1 typedef struct `sMSCAddress` `sMSCAddressInfo`

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

9.23.2.2 typedef struct `sMSEtwsMessage` `sMSEtwsMessageInfo`

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

9.23.2.3 typedef struct `sMSEtwsPImn` `sMSEtwsPImnInfo`

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

9.23.2.4 typedef struct `sSMSMessageMode` `sSMSMessageModelInfo`

Parameters

<i>messageMode</i>	Message Mode
--------------------	--------------

9.23.2.5 typedef struct **sMSMTMessage** **sMSMTMessageInfo**

Parameters

<i>storageType</i>	memory storage 0x00-UIM 0x01-NV
<i>messageIndex</i>	MT Message index

9.23.2.6 typedef struct **sMSOnIMS** **sMSOnIMSInfo**

Parameters

<i>smsOnIMS</i>	SMS on IMS
-----------------	------------

9.23.2.7 typedef struct **sMSTransferRouteMTMessage** **sMSTransferRouteMTMessageInfo**

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

9.23.3 Enumeration Type Documentation

9.23.3.1 enum **eqmiCbkSetStatus**

Enumerator

LIBPACK_QMI_CBK_PARAM_RESET
LIBPACK_QMI_CBK_PARAM_SET
LIBPACK_QMI_CBK_PARAM_NOCHANGE

9.23.4 Function Documentation

9.23.4.1 int pack_sms_SendSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SendSMS_t * *reqParam*)

send sms list pack

Parameters

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

9.23.4.2 int pack_sms_SetNewSMSCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SetNewSMSCallback_t *reqParam*)

set new sms callback pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.3 int pack_sms_SLQSDeleteSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SLQSDeleteSMS_t * *reqParam*)

delete sms pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.4 int pack_sms_SLQSGetSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SLQSGetSMS_t * *reqParam*)

get sms pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.5 int pack_sms_SLQSGetSMSList (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SLQSGetSMSList_t * *reqParam*)

get sms list pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.6 int pack_sms_SLQSModifySMSStatus (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SLQSModifySMSStatus_t * *reqParam*)

modify sms status pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.7 int unpack_sms_SendSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SendSMS_t * *pOutput*)

send sms unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.8 int unpack_sms_SetNewSMSCallback (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SetNewSMSCallback_t * *Output*)

set new sms callback unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.9 int unpack_sms_SetNewSMSCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SetNewSMSCallback_ind_t * *pOutput*)

set new sms callback indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.10 int unpack_sms_SLQSDDeleteSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSDDeleteSMS_t * *pOutput*)

delete sms unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.11 int unpack_sms_SLQSGetSMS (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMS_t * *pOutput*)

get sms unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.12 int unpack_sms_SLQSGetSMSList (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMSList_t * *pOutput*)

get sms list unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.13 int unpack_sms_SLQSMModifySMSStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSMModifySMSStatus_t * *pOutput*)

modify sms status unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.14 int unpack_sms_SLQSWmsMemoryFullCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t * *pOutput*)

sms full callback indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.24 SwiDataTypes.h File Reference

SWI data types.

Macros

- #define [SWI_API](#)
- #define [QMI_NO_LTE_FW_SUPPORT](#) 0
- #define [QMI_TLV_PLACEHOLDER](#) 0x8F
- #define [UNUSEDPARAM\(x\)](#) (void)x

Typedefs

- typedef unsigned long [ULONG](#)
- typedef unsigned long long [ULONGLONG](#)
- typedef signed char [INT8](#)
- typedef unsigned char [BYTE](#)
- typedef char [CHAR](#)
- typedef unsigned short [WORD](#)
- typedef unsigned short [USHORT](#)
- typedef const char * [LPCSTR](#)
- typedef int [BOOL](#)
- typedef signed short [SHORT](#)
- typedef signed int [INT32](#)
- typedef float [FLOAT](#)

9.24.1 Detailed Description

SWI data types.

9.24.2 Macro Definition Documentation

9.24.2.1 `#define QMI_NO_LTE_FW_SUPPORT 0`

9.24.2.2 `#define QMI_TLV_PLACEHOLDER 0x8F`

9.24.2.3 `#define SWI_API`

9.24.2.4 `#define UNUSEDPARAM(x) (void)x`

Macro used to avoid "unused variable" compiler warnings generated due to the inclusion of the "-Wextra" flag in our make files.

9.24.3 Typedef Documentation

9.24.3.1 `typedef int BOOL`

9.24.3.2 `typedef unsigned char BYTE`

9.24.3.3 `typedef char CHAR`

9.24.3.4 `typedef float FLOAT`

9.24.3.5 `typedef signed int INT32`

9.24.3.6 `typedef signed char INT8`

9.24.3.7 `typedef const char* LPCSTR`

9.24.3.8 `typedef signed short SHORT`

9.24.3.9 `typedef unsigned long ULONG`

9.24.3.10 `typedef unsigned long long ULONGLONG`

9.24.3.11 `typedef unsigned short USHORT`

9.24.3.12 `typedef unsigned short WORD`

9.25 swiloc.h File Reference

Data Structures

- struct [unpack_swiloc_SwiLocGetAutoStart_t](#)
- struct [pack_swiloc_SwiLocSetAutoStart_t](#)

Functions

- int [pack_swiloc_SwiLocGetAutoStart](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen)

- int [unpack_swiloc_SwiLocGetAutoStart](#) (uint8_t *pResp, uint16_t respLen, [unpack_swiloc_SwiLocGetAutoStart_t](#) *pOutput)
- int [pack_swiloc_SwiLocSetAutoStart](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swiloc_SwiLocSetAutoStart_t](#) *reqArg)
- int [unpack_swiloc_SwiLocSetAutoStart](#) (uint8_t *pResp, uint16_t respLen)

9.25.1 Function Documentation

9.25.1.1 int [pack_swiloc_SwiLocGetAutoStart](#) ([pack_qmi_t](#) * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Get Auto Start pack

Parameters

in	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.25.1.2 int [pack_swiloc_SwiLocSetAutoStart](#) ([pack_qmi_t](#) * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, [pack_swiloc_SwiLocSetAutoStart_t](#) * *reqArg*)

Set Auto Start pack

Parameters

in	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.25.1.3 int [unpack_swiloc_SwiLocGetAutoStart](#) (uint8_t * *pResp*, uint16_t *respLen*, [unpack_swiloc_SwiLocGetAutoStart_t](#) * *pOutput*)

Get Auto Start unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.25.1.4 int unpack_swiloc_SwiLocSetAutoStart (uint8_t * *pResp*, uint16_t *respLen*)

Set Auto Start unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.26 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_SLQSOMADMSelectSession_t](#)
- struct [pack_swioma_SLQSOMADMGetSessionInfo_t](#)
- struct [unpack_swioma_SLQSOMADMGetSessionInfo_t](#)
- struct [unpack_omaDmFotaTlv_t](#)
- struct [unpack_omaDmConfigTlv_t](#)
- struct [unpack_omaDmNotificationsTlv_t](#)
- struct [unpack_swioma_SLQSOMADMAAlertCallback_ind_t](#)

Macros

- #define [LIBPACK_MAX_SWIOMA_STR_LEN](#) 255

Functions

- int [pack_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)
- int [pack_swima_SLQSOMADMGetSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_swima_SLQSOMADMGetSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_swima_SLQSOMADMGetSettings_t](#) *pOutput)
- int [pack_swima_SLQSOMADMSetSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMSetSettings_t](#) reqParam)
- int [unpack_swima_SLQSOMADMSetSettings](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_swima_SLQSOMADMSendSelection](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMSendSelection_t](#) reqParam)
- int [unpack_swima_SLQSOMADMSendSelection](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_swima_SLQSOMADMGetSessionInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMGetSessionInfo_t](#) reqParam)
- int [unpack_swima_SLQSOMADMGetSessionInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_swima_SLQSOMADMGetSessionInfo_t](#) *pOutput)
- int [pack_swima_SLQSOMADMAAlertCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_swima_SLQSOMADMAAlertCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_swima_SLQSOMADMAAlertCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_swima_SLQSOMADMAAlertCallback_ind_t](#) *pOutput)

9.26.1 Macro Definition Documentation

9.26.1.1 `#define LIBPACK_MAX_SWIOMA_STR_LEN 255`

9.26.2 Function Documentation

9.26.2.1 `int pack_swima_SLQSOMADMAAlertCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Function to pack QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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

9.26.2.2 `int pack_swioma_SLQSOMADMCancelSession (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMCancelSession_t reqParam)`

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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

9.26.2.3 `int pack_swioma_SLQSOMADMGetSessionInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMGetSessionInfo_t reqParam)`

Function to pack QMI command to return information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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

9.26.2.4 int pack_swima_SLQSOMADMGetSettings (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Function to pack command to retrieve the OMA-DM settings from the device. This maps to SLQSOMADMGetSettings2

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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

9.26.2.5 `int pack_swioma_SLQSOMADMSendSelection (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMSendSelection_t reqParam)`

Function to pack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> 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

9.26.2.6 `int pack_swioma_SLQSOMADMSetSettings (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMSetSettings_t reqParam)`

Function to pack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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

9.26.2.7 `int pack_swima_SLQSOMADMStartSession (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_SLQSOMADMStartSession_t reqParam)`

Function to pack Start OMA-DM session command This maps to SLQSOMADMStartSession2

Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> • 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_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

9.26.2.8 int unpack_swioma_SLQSOMADMAAlertCallback (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack response of QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • Response from modem
<i>respLen</i> [IN]	<ul style="list-style-type: none"> • Length of pResp from modem

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

9.26.2.9 int unpack_swioma_SLQSOMADMAAlertCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMAAlertCallback_ind_t * *pOutput*)

Function to unpack SWIOMADM alert indications This maps to SetSLQSOMADMAAlertCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_swioama_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

9.26.2.10 int unpack_swioama_SLQSOMADMCancelSession (uint8_t * *pResp*, uint16_t *respLen*)

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • Response from modem
<i>respLen</i> [IN]	<ul style="list-style-type: none"> • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.26.2.11 int unpack_swioama_SLQSOMADMGetSessionInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_swioama_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

9.26.2.12 `int unpack_swioma_SLQSOMADMGetSettings (uint8_t * pResp, uint16_t respLen, unpack_swioma_SLQSOMADMGetSettings_t * pOutput)`

Function to unpack OMA-DM get settings response from modem This maps to SLQSOMADMGetSettings2

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> 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_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

9.26.2.13 `int unpack_swioma_SLQSOMADMSendSelection (uint8_t * pResp, uint16_t respLen)`

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> Response from modem
-------------------	---

<i>respLen</i> [IN]	<ul style="list-style-type: none"> • Length of pResp from modem
---------------------	--

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.26.2.14 int unpack_swima_SLQSOMADMSetSettings (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • Response from modem
<i>respLen</i> [IN]	<ul style="list-style-type: none"> • Length of pResp from modem

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.26.2.15 int unpack_swima_SLQSOMADMStartSession (uint8_t * *pResp*, uint16_t *respLen*, unpack_swima_SLQSOMADMStartSession_t * *pOutput*)

Function to unpack Start OMA-DM session response from modem This maps to SLQSOMADMStartSession2

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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

9.27 SWIWWANCMAPI.h File Reference

9.28 uim.h File Reference

Data Structures

- struct [uim_appStatus](#)
- struct [uim_slotInfo](#)
- struct [uim_cardStatus](#)
- struct [uim_hotSwapStatus](#)
- struct [unpack_uim_GetCardStatus_t](#)
- struct [uim_encryptedPIN1](#)
- struct [uim_remainingRetries](#)
- struct [uim_sessionInformation](#)
- struct [uim_verifyUIMPIN](#)
- struct [uim_unblockUIMPIN](#)
- struct [uim_cardResult](#)
- struct [uim_setPINProtection](#)
- struct [uim_changeUIMPIN](#)
- struct [uim_fileInfo](#)
- struct [uim_UIMSessionInformation](#)
- struct [uim_readTransparentInfo](#)
- struct [uim_readResult](#)
- struct [pack_uim_VerifyPin_t](#)
- struct [unpack_uim_VerifyPin_t](#)
- struct [pack_uim_UnblockPin_t](#)
- struct [unpack_uim_UnblockPin_t](#)
- struct [pack_uim_SetPinProtection_t](#)
- struct [unpack_uim_SetPinProtection_t](#)
- struct [pack_uim_ChangePin_t](#)
- struct [unpack_uim_ChangePin_t](#)
- struct [pack_uim_ReadTransparent_t](#)
- struct [unpack_uim_ReadTransparent_t](#)
- struct [pack_uim_SLQSUIEventRegister_t](#)
- struct [unpack_uim_SLQSUIEventRegister_t](#)
- struct [appStats](#)
- struct [slotInf](#)
- struct [unpack_uim_SLQSUISetStatusChangeCallBack_ind_t](#)
- struct [slot_t](#)
- struct [slots_t](#)
- struct [unpack_uim_SLQSUIGetSlotsStatus_t](#)
- struct [pack_uim_SLQSUISSwitchSlot_t](#)
- struct [unpack_uim_SetUimSlotStatusChangeCallback_ind_t](#)

Macros

- [#define UIM_UINT8_MAX_STRING_SZ 255](#)
- [#define UIM_MAX_DESCRIPTION_LENGTH 255](#)
- [#define UIM_MAX_NO_OF_SLOTS 5](#)
- [#define UIM_MAX_NO_OF_APPLICATIONS 10](#)
- [#define MAX_NO_OF_SLOTS 5](#)
- [#define MAX_NO_OF_APPLICATIONS 10](#)
- [#define MAX_DESCRIPTION_LENGTH 255](#)
- [#define MAX_SLOTS_STATUS 255](#)
- [#define MAX_ICCID_LENGTH 255](#)

Functions

- int [pack_uim_GetCardStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_uim_GetCardStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_GetCardStatus_t](#) *pOutput)
- int [pack_uim_VerifyPin](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_VerifyPin_t](#) *reqArg)
- int [unpack_uim_VerifyPin](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_VerifyPin_t](#) *pOutput)
- int [pack_uim_UnblockPin](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_UnblockPin_t](#) *reqArg)
- int [unpack_uim_UnblockPin](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_UnblockPin_t](#) *pOutput)
- int [pack_uim_SetPinProtection](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SetPinProtection_t](#) *reqArg)
- int [unpack_uim_SetPinProtection](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SetPinProtection_t](#) *pOutput)
- int [pack_uim_ChangePin](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_ChangePin_t](#) *reqArg)
- int [unpack_uim_ChangePin](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_ChangePin_t](#) *pOutput)
- int [pack_uim_ReadTransparent](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_ReadTransparent_t](#) *reqArg)
- int [unpack_uim_ReadTransparent](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_ReadTransparent_t](#) *pOutput)
- int [pack_uim_SLQSUIMEventRegister](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIMEventRegister_t](#) *reqArg)
- int [unpack_uim_SLQSUIMEventRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIMEventRegister_t](#) *pOutput)
- int [unpack_uim_SLQSUIMSetStatusChangeCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t](#) *pOutput)
- int [pack_uim_SLQSUIMGetSlotsStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_uim_SLQSUIMGetSlotsStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIMGetSlotsStatus_t](#) *pOutput)
- int [pack_uim_SLQSUIMSwitchSlot](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIMSwitchSlot_t](#) *reqArg)
- int [unpack_uim_SLQSUIMSwitchSlot](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_uim_SetUimSlotStatusChangeCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SetUimSlotStatusChangeCallBack_ind_t](#) *pOutput)

9.28.1 Macro Definition Documentation

9.28.1.1 `#define MAX_DESCRIPTION_LENGTH 255`

9.28.1.2 `#define MAX_ICCID_LENGTH 255`

9.28.1.3 `#define MAX_NO_OF_APPLICATIONS 10`

9.28.1.4 `#define MAX_NO_OF_SLOTS 5`

9.28.1.5 `#define MAX_SLOTS_STATUS 255`

9.28.1.6 `#define UIM_MAX_DESCRIPTION_LENGTH 255`

9.28.1.7 `#define UIM_MAX_NO_OF_APPLICATIONS 10`

9.28.1.8 `#define UIM_MAX_NO_OF_SLOTS 5`

9.28.1.9 `#define UIM_UINT8_MAX_STRING_SZ 255`

9.28.2 Function Documentation

9.28.2.1 `int pack_uim_ChangePin (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_ChangePin_t * reqArg)`

Change Pin pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.2 `int pack_uim_GetCardStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Card Status pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.3 `int pack_uim_ReadTransparent (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_ReadTransparent_t * reqArg)`

SLQS ReadTransparent pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.4 `int pack_uim_SetPinProtection (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SetPinProtection_t * reqArg)`

Set Pin Protection pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.5 `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	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.6 `int pack_uim_SLQSUIGetSlotsStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.7 int pack_uim_SLQSUIMSwitchSlot (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_uim_SLQSUIMSwitchSlot_t * *reqArg*)

switch slot pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.8 int pack_uim_UnblockPin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_UnblockPin_t * *reqArg*)

Unblock Pin pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.9 int pack_uim_VerifyPin (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_VerifyPin_t * *reqArg*)

Verify Pin Status pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.10 int unpack_uim_ChangePin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ChangePin_t * *pOutput*)

Change Pin unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.11 int unpack_uim_GetCardStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_GetCardStatus_t * *pOutput*)

Get Card Status unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.12 int unpack_uim_ReadTransparent (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ReadTransparent_t * *pOutput*)

SLQS ReadTransparent unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.13 int unpack_uim_SetPinProtection (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SetPinProtection_t * *pOutput*)

Set Pin Protection unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.14 int unpack_uim_SetUimSlotStatusChangeCallback_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_uim_SetUimSlotStatusChangeCallback_ind_t * *pOutput*)

UIM Slot Status Change indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

use pack_uim_SLQSUIEventRegister to subscribe

9.28.2.15 int unpack_uim_SLQSUIEventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIEvent-
Register_t * *pOutput*)

UIM Status Change callback enable unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.16 int unpack_uim_SLQSUIGetSlotsStatus (uint8_t * *pResp*, uint16_t *respLen*,
unpack_uim_SLQSUIGetSlotsStatus_t * *pOutput*)

get slot status unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.17 int unpack_uim_SLQSUIMSetStatusChangeCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t * *pOutput*)

UIM Status Change indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

use pack_uim_SLQSUIMEventRegister to subscribe

9.28.2.18 int unpack_uim_SLQSUIMSwitchSlot (uint8_t * *pResp*, uint16_t *respLen*)

switch slot unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.19 int unpack_uim_UnblockPin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_UnblockPin_t * *pOutput*)

Unblock Pin unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.20 int unpack_uim_VerifyPin (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_VerifyPin_t * *pOutput*)

Verify Pin unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29 wds.h File Reference

Data Structures

- struct [LibPackQosClassID](#)
- struct [LibPackTFTIDParams](#)
- struct [LibPackGPRSRequestedQoS](#)
- struct [LibPackUMTSQoS](#)
- struct [LibPackUMTSReqQoSSigInd](#)
- 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_SLQSMModifyProfile_t](#)
- struct [unpack_wds_SLQSMModifyProfile_t](#)
- struct [pack_wds_SLQSGetProfileSettings_t](#)
- struct [LibpackProfile3GPP](#)
- struct [LibpackProfile3GPP2](#)
- union [unpackWdsProfileParam](#)
- struct [UnPackGetProfileSettingOut](#)
- struct [unpack_wds_SLQSGetProfileSettings_t](#)
- struct [unpack_wds_GetSessionState_t](#)
- struct [pack_wds_GetDefaultProfile_t](#)
- struct [unpack_wds_GetDefaultProfile_t](#)
- struct [unpack_wds_GetConnectionRate_t](#)
- struct [pack_wds_GetPacketStatus_t](#)
- struct [unpack_wds_GetPacketStatus_t](#)
- struct [unpack_wds_GetSessionDuration_t](#)
- struct [pack_wds_GetSessionDuration_t](#)
- struct [unpack_wds_GetDormancyState_t](#)
- struct [pack_wds_GetDormancyState_t](#)
- struct [pack_wds_SLQSDDeleteProfile_t](#)
- struct [unpack_wds_SLQSDDeleteProfile_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 [rmTrasferStaticsReq](#)
- struct [pack_wds_RMSetTransferStatistics_t](#)
- struct [unpack_wds_RMSetTransferStatistics_t](#)
- struct [pack_wds_SetMobileIPProfile_t](#)
- struct [unpack_wds_SetMobileIPProfile_t](#)
- struct [pack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#)
- struct [ipv6AddressInfo](#)
- struct [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#)
- struct [transferStatInd](#)
- struct [pack_wds_SLQSGetDUNCallInfo_t](#)
- struct [connectionStatus](#)

- struct [dunchannelRate](#)
- struct [unpack_wds_SLQSGetDUNCallInfo_t](#)
- struct [qmiWDSDataBearerTechnology](#)
- struct [unpack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [unpack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [pack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [pack_wds_SetDefaultProfileNum_t](#)
- struct [pack_wds_GetDefaultProfileNum_t](#)
- struct [unpack_wds_GetDefaultProfileNum_t](#)
- struct [wdsDhcpv4ProfileId](#)
- struct [wdsDhcpv4HwConfig](#)
- struct [wdsDhcpv4Option](#)
- struct [wdsDhcpv4OptionList](#)
- struct [pack_wds_SLQSSetDHCPv4ClientConfig_t](#)
- struct [unpack_wds_SLQSSetDHCPv4ClientConfig_t](#)

Macros

- [#define IPV6_ADDRESS_ARRAY_SIZE 8](#)
- [#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24](#)
- [#define PACK_WDS_IPV4 4](#)
- [#define PACK_WDS_IPV6 6](#)

Typedefs

- typedef union [unpackWdsProfileParam](#) [UnpackQmiProfileInfo](#)

Functions

- int [pack_wds_SLQSStartDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSStartDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSStartDataSession](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSStartDataSession_t](#) *pOutput)
- int [unpack_wds_SLQSSetPacketSrvStatusCallback](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetPacketSrvStatusCallback_t](#) *pOutput)
- int [pack_wds_SLQSStopDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSStopDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSStopDataSession](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_wds_SLQSGetRuntimeSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetRuntimeSettings_t](#) *pOutput)
- int [unpack_wds_SLQSSetWdsEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetWdsEventCallback_ind_t](#) *pOutput)
- int [unpack_wds_SLQSSetWdsEventCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_SLQSSetWdsEventCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSetWdsEventCallback_t](#) *reqArg)
- int [pack_wds_SLQSGetRuntimeSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetRuntimeSettings_t](#) *reqArg)
- int [pack_wds_SLQSCreateProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSCreateProfile_t](#) *reqArg)
- int [unpack_wds_SLQSCreateProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSCreateProfile_t](#) *pOutput)
- int [pack_wds_SLQSModifyProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSModifyProfile_t](#) *reqArg)

- [int unpack_wds_SLQSModifyProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSModifyProfile_t](#) *pOutput)
- [int pack_wds_SLQSGetProfileSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetProfileSettings_t](#) *reqArg)
- [int unpack_wds_SLQSGetProfileSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetProfileSettings_t](#) *pOutput)
- [int pack_wds_GetSessionState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_wds_GetSessionState](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetSessionState_t](#) *pOutput)
- [int pack_wds_GetDefaultProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDefaultProfile_t](#) *reqParam)
- [int unpack_wds_GetDefaultProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDefaultProfile_t](#) *pOutput)
- [int pack_wds_GetConnectionRate](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_wds_GetConnectionRate](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetConnectionRate_t](#) *pOutput)
- [int pack_wds_GetPacketStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetPacketStatus_t](#) *reqParam)
- [int unpack_wds_GetPacketStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetPacketStatus_t](#) *pOutput)
- [int pack_wds_GetSessionDuration](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetSessionDuration_t](#) *reqParam)
- [int unpack_wds_GetSessionDuration](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetSessionDuration_t](#) *pOutput)
- [int pack_wds_GetDormancyState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDormancyState_t](#) *reqParam)
- [int unpack_wds_GetDormancyState](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDormancyState_t](#) *pOutput)
- [int pack_wds_SLQSDeleteProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSDeleteProfile_t](#) *reqParam)
- [int unpack_wds_SLQSDeleteProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSDeleteProfile_t](#) *pOutput)
- [int pack_wds_SetDefaultProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetDefaultProfile_t](#) *reqParam)
- [int unpack_wds_SetDefaultProfile](#) (uint8_t *pResp, uint16_t respLen)
- [int pack_wds_SLQSGet3GPPConfigItem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_wds_SLQSGet3GPPConfigItem](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGet3GPPConfigItem_t](#) *pOutput)
- [int pack_wds_SLQSSet3GPPConfigItem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSet3GPPConfigItem_t](#) *reqParam)
- [int unpack_wds_SLQSSet3GPPConfigItem](#) (uint8_t *pResp, uint16_t respLen)
- [int pack_wds_GetMobileIP](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetMobileIP_t](#) *pReqParam)
- [int unpack_wds_GetMobileIP](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetMobileIP_t](#) *pOutput)
- [int pack_wds_GetMobileIPProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetMobileIPProfile_t](#) *reqParam)
- [int unpack_wds_GetMobileIPProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetMobileIPProfile_t](#) *pOutput)
- [int pack_wds_SLQSGetCurrDataSystemStat](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetCurrDataSystemStat_t](#) *pReqParam)
- [int unpack_wds_SLQSGetCurrDataSystemStat](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetCurrDataSystemStat_t](#) *pOutput)
- [int pack_wds_GetLastMobileIPError](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetLastMobileIPError_t](#) *pReqParam)
- [int unpack_wds_GetLastMobileIPError](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetLastMobileIPError_t](#) *pOutput)

- int [pack_wds_RMSetTransferStatistics](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_RMSetTransferStatistics_t](#) *reqParam)
- int [unpack_wds_RMSetTransferStatistics](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_RMSetTransferStatistics_t](#) *pOutput)
- int [pack_wds_SetMobileIPProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetMobileIPProfile_t](#) *reqParam)
- int [unpack_wds_SetMobileIPProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SetMobileIPProfile_t](#) *pOutput)
- int [pack_wds_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)
- 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_SLQSSGetDHCPv4ClientConfig](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSGetDHCPv4ClientConfig_t](#) *pReq)
- int [unpack_wds_SLQSSGetDHCPv4ClientConfig](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSGetDHCPv4ClientConfig_t](#) *pOutput)

9.29.1 Macro Definition Documentation

9.29.1.1 `#define IPV6_ADDRESS_ARRAY_SIZE 8`

9.29.1.2 `#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24`

9.29.1.3 `#define PACK_WDS_IPV4 4`

9.29.1.4 `#define PACK_WDS_IPV6 6`

9.29.2 Typedef Documentation

9.29.2.1 `typedef union unpackWdsProfileParam UnpackQmiProfileInfo`

9.29.3 Function Documentation

9.29.3.1 `int pack_wds_GetConnectionRate (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get connection rate pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: No

9.29.3.2 int pack_wds_GetDefaultProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_GetDefaultProfile_t * *reqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.3 int pack_wds_GetDefaultProfileNum (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_GetDefaultProfileNum_t * *pReqParam*)

get default profile number pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.4 `int pack_wds_GetDormancyState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
pack_wds_GetDormancyState_t * reqParam)`

get dormancy state pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: No

9.29.3.5 `int pack_wds_GetLastMobileIPError (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetLastMobileIPError_t * pReqParam)`

get current data system pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: No

9.29.3.6 `int pack_wds_GetMobileIP (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetMobileIP_t * pReqParam)`

get mobile ip mode pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: No

9.29.3.7 `int pack_wds_GetMobileIPProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetMobileIPProfile_t * reqParam)`

get mobile ip profile pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.8 `int pack_wds_GetPacketStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetPacketStatus_t * reqParam)`

get packet status pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.9 `int pack_wds_GetSessionDuration (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetSessionDuration_t * reqParam)`

get session duration pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: No

9.29.3.10 int pack_wds_GetSessionState (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get session state pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.11 int pack_wds_RMSetTransferStatistics (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_RMSetTransferStatistics_t * *reqParam*)

rm set transfer statistics pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

Note

PDN Specific: No

9.29.3.12 `int pack_wds_SetDefaultProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
pack_wds_SetDefaultProfile_t * reqParam)`

set default profile pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.13 int pack_wds_SetDefaultProfileNum (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetDefaultProfileNum_t * *pReqParam*)

set default profile number pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.14 int pack_wds_SetMobileIPProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SetMobileIPProfile_t * *reqParam*)

set mobile ip profile pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.15 `int pack_wds_SLQSCreateProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSCreateProfile_t * reqArg)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.16 `int pack_wds_SLQSDeleteProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSDeleteProfile_t * reqParam)`

delete stored profile pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.17 `int pack_wds_SLQSGet3GPPConfigItem (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get 3Gpp config items pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.18 int pack_wds_SLQSGetCurrDataSystemStat (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSGetCurrDataSystemStat_t * *pReqParam*)

get current data system pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.19 int pack_wds_SLQSGetDataBearerTechnology (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSGetDataBearerTechnology_t * *pReqParam*)

get data bearer technology pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

out	<i>reqParam</i>	request parameter
-----	-----------------	-------------------

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.20 int pack_wds_SLQSGetDUNCallInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSGetDUNCallInfo_t * *reqParam*)

get dun call info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.21 int pack_wds_SLQSGetProfileSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSGetProfileSettings_t * *reqArg*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.22 int pack_wds_SLQSGetRuntimeSettings (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSGetRuntimeSettings_t * *reqArg*)

get runtime settings pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.23 int pack_wds_SLQSModifyProfile (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSModifyProfile_t * *reqArg*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.24 int pack_wds_SLQSSet3GPPConfigItem (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSSet3GPPConfigItem_t * *reqParam*)

set 3Gpp config items pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.25 `int pack_wds_SLQSSetIPFamilyPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSetIPFamilyPreference_t * pReqParam)`

Set IP Family Preference pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.26 `int pack_wds_SLQSSetWdsEventCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSetWdsEventCallback_t * reqArg)`

set event callback pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: No

9.29.3.27 `int pack_wds_SLQSSetDHCPv4ClientConfig (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSetDHCPv4ClientConfig_t * pReq)`

get DHCPv4 Client Config pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReq</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.28 int pack_wds_SLQSStartDataSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSStartDataSession_t * *reqArg*)

Start data session**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.29 int pack_wds_SLQSStopDataSession (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSStopDataSession_t * *reqArg*)

stop data session pack**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.30 `int pack_wds_SLQSWdsSwiPDPRuntimeSettings (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t * reqParam)`

swi pdp runtime settings pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.31 `int unpack_wds_GetConnectionRate (uint8_t * pResp, uint16_t respLen, unpack_wds_GetConnectionRate_t * pOutput)`

get connection rate unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.32 `int unpack_wds_GetDefaultProfile (uint8_t * pResp, uint16_t respLen, unpack_wds_GetDefaultProfile_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.33 int unpack_wds_GetDefaultProfileNum (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDefaultProfile-Num_t * *pOutput*)

get default profile number unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.34 int unpack_wds_GetDormancyState (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetDormancyState_t * *pOutput*)

get dormancy state unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.35 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

9.29.3.36 int unpack_wds_GetMobileIP (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetMobileIP_t * *pOutput*)

get mobile ip mode unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.37 int unpack_wds_GetMobileIPProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetMobileIPProfile_t * *pOutput*)

get mobile ip profile unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.38 int unpack_wds_GetPacketStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatus_t * *pOutput*)

get packet status unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.39 int unpack_wds_GetSessionDuration (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetSessionDuration_t * *pOutput*)

get session duration unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.40 int unpack_wds_GetSessionState (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetSessionState_t * *pOutput*)

get session state unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.41 int unpack_wds_RMSetTransferStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_RMSetTransferStatistics_t * *pOutput*)

rm set transfer statistics unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.42 int unpack_wds_SetDefaultProfile (uint8_t * *pResp*, uint16_t *respLen*)

set default profile unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.43 int unpack_wds_SetDefaultProfileNum (uint8_t * *pResp*, uint16_t *respLen*)

set default profile number unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.44 int unpack_wds_SetMobileIPProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SetMobileIPProfile_t * *pOutput*)

set mobile ip profile unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.45 int unpack_wds_SLQSCreateProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSCreateProfile_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response structure to fill
in	<i>pProfileId</i>	profile id pointer passed in req

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.46 int unpack_wds_SLQSDeleteProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSDeleteProfile_t * *pOutput*)

delete stored profile unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.47 int unpack_wds_SLQSGet3GPPConfigItem (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGet3GPPConfigItem_t * *pOutput*)

get 3GPP config items unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.48 int unpack_wds_SLQSGetCurrDataSystemStat (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSGetCurrDataSystemStat_t * *pOutput*)

get current data system unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.49 int unpack_wds_SLQSGetDataBearerTechnology (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSGetDataBearerTechnology_t * *pOutput*)

get data bearer technology unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.50 int unpack_wds_SLQSGetDUNCallInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetDUNCall-
Info_t * *pOutput*)

get dun call info unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.51 int unpack_wds_SLQSGetProfileSettings (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSGetProfileSettings_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response structure to fill

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.52 int unpack_wds_SLQSGetRuntimeSettings (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSGetRuntimeSettings_t * *pOutput*)

get runtime settings unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.53 int unpack_wds_SLQSModifyProfile (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSModifyProfile_t
* *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.54 int unpack_wds_SLQSSet3GPPConfigItem (uint8_t * *pResp*, uint16_t *respLen*)

set 3GPP config items unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

9.29.3.55 int unpack_wds_SLQSSetIPFamilyPreference (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSSetIPFamilyPreference_t * *pOutput*)

Set IP Family Preference unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.56 int unpack_wds_SLQSSetPacketSrvStatusCallback (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSSetPacketSrvStatusCallback_t * *pOutput*)

set packet srv status callback unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	length

out	<i>pOutput</i>	unpacked response
-----	----------------	-------------------

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.57 int unpack_wds_SLQSSetWdsEventCallback (uint8_t * *pResp*, uint16_t *respLen*)

set event callback unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.58 int unpack_wds_SLQSSetWdsEventCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSetWdsEventCallback_ind_t * *pOutput*)

set event callback unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.59 int unpack_wds_SLQSSetDHCPv4ClientConfig (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSetDHCPv4ClientConfig_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

9.29.3.60 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

9.29.3.61 int unpack_wds_SLQSStopDataSession (uint8_t * *pResp*, uint16_t *respLen*)

stop data session unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.62 int unpack_wds_SLQSWdsSwiPDPRuntimeSettings (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t * *pOutput*)

get current data system unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Index

- [_3gppRelease](#)
 - [pack_wds_SLQSSet3GPPConfigItem_t, 245](#)
 - [unpack_wds_SLQSGet3GPPConfigItem_t, 400](#)
- [AAASPI](#)
 - [unpack_wds_GetMobileIPProfile_t, 396](#)
- [AAASState](#)
 - [unpack_wds_GetMobileIPProfile_t, 396](#)
- [AMSSString](#)
 - [unpack_dms_GetFirmwareRevision_t, 287](#)
 - [unpack_dms_GetFirmwareRevisions_t, 287](#)
- [APNName](#)
 - [unpack_wds_SLQSGetRuntimeSettings_t, 404](#)
- [accolc](#)
 - [pack_nas_SetACCOLC_t, 195](#)
- [ackIndicator](#)
 - [sMSTransferRouteMTMessage, 262](#)
- [acqOrdeLen](#)
 - [nas_acqOrderPref, 83](#)
- [activationStatus](#)
 - [dms_ActivationStatusTlv, 28](#)
- [ActivationStatusTlv](#)
 - [unpack_dms_SetEventReport_ind_t, 293](#)
- [activeBandClass](#)
 - [nas_RFInfoTlv, 145](#)
 - [RFBandInfoElements, 252](#)
- [activeChannel](#)
 - [nas_RFInfoTlv, 145](#)
 - [RFBandInfoElements, 252](#)
- [ActiveTechPref](#)
 - [unpack_nas_GetNetworkPreference_t, 317](#)
- [addr](#)
 - [unpack_qos_IPv4Addr_t, 349](#)
 - [unpack_qos_IPv6Addr_t, 349](#)
- [address](#)
 - [unpack_wds_GetMobileIPProfile_t, 396](#)
- [aid](#)
 - [uim_sessionInformation, 275](#)
 - [uim_UIMSessionInformation, 279](#)
- [aidLength](#)
 - [appStats, 24](#)
 - [uim_appStatus, 267](#)
 - [uim_sessionInformation, 275](#)
 - [uim_UIMSessionInformation, 279](#)
- [aidVal](#)
 - [appStats, 24](#)
 - [uim_appStatus, 267](#)
- [aidingIndicatorMask](#)
 - [loc_sensorDataUsage, 79](#)
- [alertmsg](#)
 - [unpack_omaDmConfigTlv_t, 345](#)
- [alertmsglength](#)
 - [unpack_omaDmConfigTlv_t, 345](#)
- [ambr_dl](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
- [ambr_dl_ext](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
- [ambr_dl_ext2](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
- [ambr_ul](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
- [ambr_ul_ext](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
- [ambr_ul_ext2](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
- [amssSize](#)
 - [unpack_dms_GetFirmwareRevision_t, 287](#)
 - [unpack_dms_GetFirmwareRevisions_t, 287](#)
- [apdoxypages.c, 427](#)
- [apnId](#)
 - [pack_qos_SLQSQosSwiReadApnExtraParams_t, 214](#)
 - [pack_qos_SLQSQosSwiReadDataStats_t, 214](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 356](#)
 - [unpack_qos_SLQSQosSwiReadDataStats_t, 357](#)
- [apnName](#)
 - [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 410](#)
- [apnname](#)
 - [unpack_wds_GetDefaultProfile_t, 393](#)
- [apnsize](#)
 - [unpack_wds_GetDefaultProfile_t, 393](#)
- [appNameLength](#)
 - [loc_LocApplicationInfo, 77](#)
- [appProviderLength](#)
 - [loc_LocApplicationInfo, 77](#)
- [appState](#)
 - [appStats, 24](#)
 - [uim_appStatus, 267](#)
- [appStats, 21](#)
 - [aidLength, 24](#)
 - [aidVal, 24](#)

- appState, [24](#)
- appType, [24](#)
- persoFeature, [24](#)
- persoRetries, [24](#)
- persoState, [24](#)
- persoUnblockRetries, [24](#)
- pin1Retries, [24](#)
- pin1State, [24](#)
- pin2Retries, [24](#)
- pin2State, [24](#)
- puk1Retries, [24](#)
- puk2Retries, [24](#)
- univPin, [25](#)
- AppStatus
 - slotInf, [256](#)
 - uim_slotInfo, [278](#)
- appType
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- appVersionLength
 - loc_LocApplicationInfo, [77](#)
- appVersionValid
 - loc_LocApplicationInfo, [77](#)
- Application
 - unpack_nas_GetCDMANetworkParameters_t, [314](#)
- appversion_str
 - unpack_dms_GetFirmwareInfo_t, [286](#)
- arfcn
 - nas_GERANInfo, [106](#)
 - nas_gsmCellInfo, [108](#)
- auth
 - unpack_wds_GetDefaultProfile_t, [393](#)
- Authentication
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- authentication
 - pack_wds_SetDefaultProfile_t, [238](#)
- Autosdm
 - unpack_swima_SLQSOMADMGetSettings_t, [384](#)
- avgPeriod
 - nas_LTESigRptConfig, [132](#)
- bEnable
 - pack_nas_SLQSSetSignalStrengthsCallback_t, [209](#)
- bForceDownload
 - pack_fms_SetImagesPreference_t, [187](#)
- bICCID
 - slot_t, [253](#)
- bICCIDLength
 - slot_t, [253](#)
- bLogicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, [231](#)
 - slot_t, [253](#)
- bNumberOfPhySlots
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, [388](#)
- BOOL
 - SwiDataTypes.h, [567](#)
- BPTlv
 - NASQmiCbkNasSystemSelPrefInd, [178](#)
- bResetStatistics
 - rmTrasferStaticsReq, [252](#)
- BYTE
 - SwiDataTypes.h, [567](#)
- band
 - nas_LTEInfo, [123](#)
- band1900
 - nas_gsmCellInfo, [108](#)
- band_pref
 - NASBandPreferenceTlv, [171](#)
- BandCapability
 - unpack_dms_GetBandCapability_t, [282](#)
- bandCapability
 - unpack_dms_SLQSGetBandCapability_t, [298](#)
- bandwidth
 - nas_LTEInfo, [123](#)
- baseId
 - nas_CDMAInfo, [88](#)
 - nas_CDMA SysInfo, [93](#)
- baseLat
 - nas_CDMAInfo, [88](#)
 - nas_CDMA SysInfo, [93](#)
- baseLong
 - nas_CDMAInfo, [88](#)
 - nas_CDMA SysInfo, [93](#)
- BasestationID
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- BasestationLatitude
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- BasestationLongitude
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- BearerID
 - unpack_qos_QosFlowInfo_t, [352](#)
- bearerID
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [406](#)
- bearerId
 - unpack_QosFlowStat_t, [372](#)
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [410](#)
- bootSize
 - unpack_dms_GetFirmwareRevisions_t, [287](#)
- BootString
 - unpack_dms_GetFirmwareRevisions_t, [287](#)
- bootversion_str
 - unpack_dms_GetFirmwareInfo_t, [286](#)
- Broadcast
 - unpack_nas_GetCDMANetworkParameters_t, [314](#)
- bsInfoValid
 - nas_CDMA SysInfo, [93](#)
- bsPRev
 - nas_CDMA SysInfo, [93](#)
- bsPRevValid
 - nas_CDMA SysInfo, [93](#)
- bsic
 - nas_GERANInfo, [106](#)
- bsicId

- nas_gsmCellInfo, 108
- bucketSz
 - unpack_qos_tokenBucket_t, 370
- buildID
 - FMSImageIdElement, 34
 - image_info_t, 37
- buildIDLen
 - image_info_t, 37
- buildIDLength
 - FMSImageIdElement, 34
- buildId
 - FMSImageElement, 33
- buildIdLength
 - FMSImageElement, 33
- CDMA_P_Rev
 - unpack_nas_SLQSGetServingSystem_t, 327
- CDMAECIOThreshListLen
 - nas_CDMAECIOThresh, 87
- CDMARSSIThreshListLen
 - nas_CDMARSSIThresh, 89
- CDMASSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 337
- CDMASystemInfoExt
 - unpack_nas_SLQSGetServingSystem_t, 327
- CHAR
 - SwiDataTypes.h, 567
- CQIValueCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, 341
- CQIValueCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, 341
- CSDomain
 - unpack_nas_GetServingNetwork_t, 319
- CallBarStatus
 - unpack_nas_SLQSGetServingSystem_t, 327
- callDuration
 - unpack_wds_GetSessionDuration_t, 398
- callEndReason
 - unpack_wds_SLQSGetDUNCallInfo_t, 402
- cardState
 - slotInf, 256
 - uim_slotInfo, 278
- carrier
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 301
- carrier_str
 - unpack_dms_GetFirmwareInfo_t, 286
- CarrierImage_t, 25
 - m_FwBuildId, 25
 - m_FwImageld, 26
 - m_PriBuildId, 26
 - m_PriImageld, 26
 - m_nCarrierId, 26
 - m_nFolderId, 26
 - m_nStorage, 26
- ccsSupported
 - nas_CDMASysInfo, 93
- ccsSupportedValid
 - nas_CDMASysInfo, 93
- cdmaSSInfo, 26
 - ecio, 26
 - rsi, 26
- cdmaSysIdValid
 - nas_CDMASysInfo, 93
- cell_resel_priority
 - nas_infoInterFreq, 119
- cellBroadcastCap
 - nas_AddSysInfo, 84
- CellID
 - unpack_nas_SLQSGetServingSystem_t, 327
- cellID
 - nas_GERANInfo, 106
 - nas_UMTSInfo, 160
- cellId
 - nas_GSMSysInfo, 113
 - nas_LTESysInfo, 135
 - nas_WCDMASysInfo, 170
- cellIdValid
 - nas_gsmCellInfo, 108
 - nas_GSMSysInfo, 113
 - nas_LTESysInfo, 135
 - nas_WCDMASysInfo, 170
- cellInterFreqParams
 - nas_infoInterFreq, 119
- cellsTDD
 - nas_umtsLTENbrCell, 163
- CellParams
 - nas_LTEInfoIntrafreq, 126
- cellReselPriority
 - nas_lteGsmCellInfo, 121
 - nas_LTEInfoIntrafreq, 126
 - nas_lteWcdmaCellInfo, 137
- cells_len
 - nas_infoInterFreq, 120
 - nas_lteGsmCellInfo, 121
- cellsLen
 - nas_LTEInfoIntrafreq, 126
 - nas_lteWcdmaCellInfo, 137
- chaddr
 - wdsDhcpv4HwConfig, 423
- chaddrLen
 - wdsDhcpv4HwConfig, 423
- changePIN
 - pack_uim_ChangePin_t, 227
- channelRate
 - unpack_wds_SLQSGetDUNCallInfo_t, 402
- common.h
 - eCTL, 429
 - eDMS, 429
 - eIND, 430
 - eLOC, 429
 - eLOG_DEBUG, 429
 - eLOG_FATAL, 429
 - eLOG_INFO, 429
 - eLOG_WARN, 429
 - eNAS, 429
 - eQOS, 429
 - eREQ, 430

- eRSP, [430](#)
- eSMS, [429](#)
- eSWILOC, [429](#)
- eSWIOMA, [429](#)
- eTIMEOUT_10_S, [430](#)
- eTIMEOUT_20_S, [430](#)
- eTIMEOUT_2_S, [430](#)
- eTIMEOUT_300_S, [430](#)
- eTIMEOUT_30_S, [430](#)
- eTIMEOUT_5_S, [430](#)
- eTIMEOUT_60_S, [430](#)
- eTIMEOUT_8_S, [430](#)
- eTIMEOUT_DEFAULT, [430](#)
- eUIM, [429](#)
- eWDS, [429](#)
- common.h, [427](#)
 - eLOG_LEVEL, [429](#)
 - eQMI_SVC, [429](#)
 - eTimeout, [429](#)
 - fill_pack_ctx, [430](#)
 - fill_sdu_hdr, [430](#)
 - get_version, [430](#)
 - glog, [431](#)
 - gloglvl, [431](#)
 - helper_get_resp_ctx, [430](#)
 - helper_get_xid, [430](#)
 - helper_set_log_func, [431](#)
 - helper_set_log_lvl, [431](#)
 - libpack_log, [431](#)
 - logger, [429](#)
 - MINREQBKLEN, [428](#)
 - MSGID_AND_LEN, [429](#)
 - MSGID_DONT_CARE, [429](#)
 - msgtype, [430](#)
 - SDU_HDR_LEN, [429](#)
 - UNUSEDPARAM, [429](#)
 - unpack_result_code_only, [431](#)
- commonInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, [339](#)
- ConcSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- conn_status
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [406](#)
- connectionStatus, [26](#)
 - MDMCallDuration, [27](#)
 - MDMConnStatus, [27](#)
 - unpack_wds_GetSessionState_t, [398](#)
 - unpack_wds_SLQSGetDUNCallInfo_t, [402](#)
- content
 - uim_readResult, [273](#)
- contentLen
 - uim_readResult, [273](#)
- contextId
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [249](#)
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [410](#)
- contextType
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [249](#)
- cpich_ecno
 - nas_wcdmaCellInfo, [164](#)
- cpich_rscp
 - nas_wcdmaCellInfo, [164](#)
- csAttachState
 - nas_servSystem, [149](#)
 - NASServingSystemInfo, [180](#)
- csBarStatus
 - nas_CallBarringSysInfo, [85](#)
 - nas_callBarStatus, [86](#)
- cur_carr_name
 - unpack_dms_GetFirmwareInfo_t, [286](#)
- cur_carr_rev
 - unpack_dms_GetFirmwareInfo_t, [286](#)
- curDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, [401](#)
- curProfile
 - pack_wds_SLQSModifyProfile_t, [244](#)
 - UnPackGetProfileSettingOut, [411](#)
- CurrChanRxRate
 - dunchannelRate, [32](#)
- CurrChanTxRate
 - dunchannelRate, [32](#)
- currDBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- currNWInfo
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- currNetworkInfo, [27](#)
 - NetworkType, [27](#)
 - RATMask, [27](#)
 - SOMask, [27](#)
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [401](#)
- currentChannelRXRate
 - unpack_wds_GetConnectionRate_t, [393](#)
- currentChannelTXRate
 - unpack_wds_GetConnectionRate_t, [393](#)
- currentDataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, [246](#)
- currentNetwork
 - qmiWDSDataBearerTechnology, [251](#)
- CurrentPLMN
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- cust_attr
 - DMScustSettingInfo, [29](#)
- cust_id
 - DMScustSettingInfo, [29](#)
 - DMSgetCustomInput, [31](#)
 - pack_dms_GetCustFeaturesV2_t, [182](#)
 - pack_dms_SetCustFeaturesV2_t, [183](#)
- cust_value
 - DMScustSettingInfo, [29](#)

- pack_dms_SetCustFeaturesV2_t, 183
- custSetting
 - DMScustSettingList, 30
- CustomSCP
 - unpack_nas_GetCDMANetworkParameters_t, 314
- dBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- dBTechnology
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- DHCPRelayEnabled
 - pack_dms_SetCustFeature_t, 182
 - unpack_dms_GetCustFeature_t, 282
- DMS_IMGDETAILS_LEN
 - dms.h, 435
- DMS_PM_FACTORY
 - dms.h, 435
- DMS_PM_LOW
 - dms.h, 435
- DMS_PM_OFFLINE
 - dms.h, 435
- DMS_PM_ONLINE
 - dms.h, 435
- DMS_PM_RESET
 - dms.h, 435
- DMS_PM_SHUT_DOWN
 - dms.h, 435
- DMScustSettingInfo, 29
 - cust_attr, 29
 - cust_id, 29
 - cust_value, 29
 - id_length, 29
 - value_length, 29
- DMScustSettingList, 30
 - custSetting, 30
 - list_type, 30
 - num_instances, 30
- DMSgetCustomFeatureV2, 30
 - pCustSettingInfo, 31
 - pCustSettingList, 31
 - pGetCustomInput, 31
- DMSgetCustomInput, 31
 - cust_id, 31
 - list_type, 31
- DTMInd
 - unpack_nas_SLQSSetServingSystem_t, 327
- data
 - sMSCAddress, 256
 - sMSEtwsMessage, 258
 - sMSTransferRouteMTMessage, 262
- data_buf
 - NASOTAMessageTlv, 173
- data_len
 - NASOTAMessageTlv, 173
- dataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, 246
- dataBearerMask
 - unpack_wds_SLQSSetServingSystem_t, 327
- dataBearerTech
 - unpack_wds_SLQSSetDUNCallInfo_t, 402
- dataCapabilities
 - nas_dataSrvCapabilities, 100
- dataCapabilitiesLen
 - nas_dataSrvCapabilities, 100
- DataCaps
 - unpack_nas_GetServingNetwork_t, 319
 - unpack_nas_GetServingNetworkCapabilities_t, 319
- dataCaps
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 321
- DataCapsLen
 - unpack_nas_GetServingNetwork_t, 319
 - unpack_nas_GetServingNetworkCapabilities_t, 319
- dataCapsSize
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 321
- DataRate
 - unpack_qos_swiQoSFlow_t, 369
- dataRateMax
 - unpack_qos_dataRate_t, 348
- dataServiceCaCapability
 - unpack_dms_GetDeviceCapabilities_t, 284
- DataServiceCapability
 - unpack_dms_GetDeviceCap_t, 283
- DataSrvCapabilities
 - unpack_nas_SLQSSetServingSystem_t, 327
- dataSysStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- dataSystemStatus
 - pack_wds_SLQSSetWdsEventCallback_t, 246
- Date
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 381
- DateLength
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 382
- daylightSavings
 - nas_qaQmi3Gpp2TimeZone, 141
- defaultPDNEnabled
 - pack_wds_SLQSSet3GPPConfigItem_t, 245
 - unpack_wds_SLQSSet3GPPConfigItem_t, 400
- DefaultRoamInd
 - unpack_nas_SLQSSetServingSystem_t, 327
- delayClass
 - LibPackGPRSRequestedQoS, 38
 - wds_GPRSQoS, 416
- deliveryErrSDU
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- description
 - unpack_omaDmFotaTlv_t, 347

- descriptionlength
 - unpack_omaDmFotaTlv_t, 347
- Description
 - nas_QmiNas3GppNetworkInfo, 142
- destPortRangeEnd
 - LibPackTFTIDParams, 66
- destPortRangeStart
 - LibPackTFTIDParams, 66
- DetailedSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, 327
- DevCrashState
 - unpack_dms_GetCrashAction_t, 282
- DisableIMSI
 - pack_dms_SetCustFeature_t, 182
 - unpack_dms_GetCustFeature_t, 282
- dl_bw_value
 - NASPhyCaAggPcellInfo, 174
 - NASPhyCaAggScellIDBw, 175
 - NASPhyCaAggScellInfo, 177
- dms.h, 431
 - DMS_IMGDETAILS_LEN, 435
 - DMS_PM_FACTORY, 435
 - DMS_PM_LOW, 435
 - DMS_PM_OFFLINE, 435
 - DMS_PM_ONLINE, 435
 - DMS_PM_RESET, 435
 - DMS_PM_SHUT_DOWN, 435
 - MAX_BUILD_ID_LEN, 435
 - pack_dms_GetActivationState, 436
 - pack_dms_GetBandCapability, 436
 - pack_dms_GetCrashAction, 436
 - pack_dms_GetCustFeature, 438
 - pack_dms_GetCustFeaturesV2, 438
 - pack_dms_GetDeviceCap, 438
 - pack_dms_GetDeviceCapabilities, 438
 - pack_dms_GetDeviceHardwareRev, 439
 - pack_dms_GetDeviceMfr, 439
 - pack_dms_GetDeviceSerialNumbers, 439
 - pack_dms_GetFSN, 441
 - pack_dms_GetFirmwareInfo, 440
 - pack_dms_GetFirmwareRevision, 440
 - pack_dms_GetFirmwareRevisions, 440
 - pack_dms_GetHardwareRevision, 441
 - pack_dms_GetIMSI, 441
 - pack_dms_GetModelID, 442
 - pack_dms_GetNetworkTime, 442
 - pack_dms_GetPRLVersion, 443
 - pack_dms_GetPower, 442
 - pack_dms_GetSerialNumbers, 443
 - pack_dms_GetUSBComp, 443
 - pack_dms_GetVoiceNumber, 444
 - pack_dms_SLQSDmsSwiGetResetInfo, 446
 - pack_dms_SLQSDmsSwiIndicationRegister, 446
 - pack_dms_SLQSGetBandCapability, 447
 - pack_dms_SLQSSwiClearDyingGaspStatistics, 447
 - pack_dms_SLQSSwiGetDyingGaspCfg, 447
 - pack_dms_SLQSSwiGetDyingGaspStatistics, 448
 - pack_dms_SLQSSwiGetFirmwareCurr, 448
 - pack_dms_SLQSSwiSetDyingGaspCfg, 448
 - pack_dms_SetCustFeature, 444
 - pack_dms_SetCustFeaturesV2, 444
 - pack_dms_SetEventReport, 445
 - pack_dms_SetFirmwarePreference, 445
 - pack_dms_SetPower, 445
 - pack_dms_SetUSBComp, 446
 - pack_dms_UIMGetICCID, 449
 - SLQSFWINFO_SKU_SZ, 436
 - UNIQUE_ID_LEN, 436
 - unpack_dms_GetActivationState, 449
 - unpack_dms_GetBandCapability, 449
 - unpack_dms_GetCrashAction, 450
 - unpack_dms_GetCustFeature, 450
 - unpack_dms_GetCustFeaturesV2, 450
 - unpack_dms_GetDeviceCap, 450
 - unpack_dms_GetDeviceCapabilities, 451
 - unpack_dms_GetDeviceHardwareRev, 451
 - unpack_dms_GetDeviceMfr, 451
 - unpack_dms_GetDeviceSerialNumbers, 452
 - unpack_dms_GetFSN, 453
 - unpack_dms_GetFirmwareInfo, 452
 - unpack_dms_GetFirmwareRevision, 452
 - unpack_dms_GetFirmwareRevisions, 453
 - unpack_dms_GetHardwareRevision, 453
 - unpack_dms_GetIMSI, 454
 - unpack_dms_GetModelID, 454
 - unpack_dms_GetNetworkTime, 454
 - unpack_dms_GetPRLVersion, 455
 - unpack_dms_GetPower, 455
 - unpack_dms_GetSerialNumbers, 455
 - unpack_dms_GetUSBComp, 456
 - unpack_dms_GetVoiceNumber, 456
 - unpack_dms_SLQSDmsSwiGetResetInfo, 459
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 459
 - unpack_dms_SLQSDmsSwiIndicationRegister, 459
 - unpack_dms_SLQSGetBandCapability, 460
 - unpack_dms_SLQSSwiClearDyingGaspStatistics, 460
 - unpack_dms_SLQSSwiGetDyingGaspCfg, 460
 - unpack_dms_SLQSSwiGetDyingGaspStatistics, 461
 - unpack_dms_SLQSSwiGetFirmwareCurr, 461
 - unpack_dms_SLQSSwiSetDyingGaspCfg, 461
 - unpack_dms_SetCustFeature, 456
 - unpack_dms_SetCustFeaturesV2, 457
 - unpack_dms_SetEventReport, 457
 - unpack_dms_SetEventReport_ind, 457
 - unpack_dms_SetFirmwarePreference, 458
 - unpack_dms_SetPower, 458
 - unpack_dms_SetUSBComp, 458
 - unpack_dms_UIMGetICCID, 462
- dms_ActivationStatusTlv, 27
 - activationStatus, 28
 - TlvPresent, 28

- dms_OperatingModeTlv, [28](#)
 - operatingMode, [29](#)
 - TlvPresent, [29](#)
- domain
 - wds_DomainNameList, [414](#)
- domainLen
 - wds_Domain, [414](#)
- DomainList
 - unpack_wds_SLQSSetRuntimeSettings_t, [404](#)
- domainName
 - wds_Domain, [414](#)
- dormancyStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- dormancyState
 - unpack_wds_GetDormancyState_t, [395](#)
- dormancyStatus
 - pack_wds_SLQSSetWdsEventCallback_t, [246](#)
 - unpack_wds_SLQSSetDUNCallInfo_t, [402](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- dtmSupp
 - nas_GSMSSysInfo, [113](#)
- dtmSuppValid
 - nas_GSMSSysInfo, [113](#)
- dunchannelRate, [31](#)
 - CurrChanRxRate, [32](#)
 - CurrChanTxRate, [32](#)
 - MaxChanRxRate, [32](#)
 - MaxChanTxRate, [32](#)
- Duration
 - pack_nas_SetNetworkPreference_t, [196](#)
 - unpack_nas_GetNetworkPreference_t, [317](#)
- eCTL
 - common.h, [429](#)
- eDMS
 - common.h, [429](#)
- eIND
 - common.h, [430](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
 - nas.h, [480](#)
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
 - nas.h, [480](#)
- eLOC
 - common.h, [429](#)
- eLOG_DEBUG
 - common.h, [429](#)
- eLOG_FATAL
 - common.h, [429](#)
- eLOG_INFO
 - common.h, [429](#)
- eLOG_WARN
 - common.h, [429](#)
- eNAS
 - common.h, [429](#)
- eQCWWAN_ERR_API_MUTEX_TIMEOUT
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_BUFFER_SZ
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_CANCEL_OP
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_DRIVER
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_ENUM_BEGIN
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_ENUM_END
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_FILE_COPY
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_FILE_OPEN
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_GENERAL
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_INTERNAL
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_INVALID_ARG
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_INVALID_DEVID
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_INVALID_FILE
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_INVALID_QMI_RSP
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_INVALID_XID
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_MALFORMED_QMI_RSP
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_MEMORY
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_MULTIPLE_DEVICES
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_NO_CANCELABLE_OP
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_NO_CONNECTION
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_NO_DEVICE
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_NO_SIGNAL

- qmerrno.h, [542](#)
- eQCWWAN_ERR_NONE
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_NULL_TLV
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_OFFLINE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_PDU_GENERATION
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_ABORTED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_ACCESS_DENIED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_ACK_NOT_SENT
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_ARG_TOO_LONG
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_CALL_FAILED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_CARD_BUSY_RSP
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_CAT_END
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_CAT_START
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_CAUSE_CODE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_CONNECT
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_DEVICE_IN_USE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_DEVICE_NOT_READY
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_DISABLED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_ENCODING
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_EVENT_REG_FAILED
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_FDN_RESTRICT
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_FLOW_SUSPENDED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_GENERAL
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_IFACE
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INCORRECT_PIN
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_INJECT_TIMEOUT
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCE-
 - S
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INTERNAL
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_ARG
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_INVALID_HANDLE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_ID
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_INDEX
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_OPERATION
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_PINID
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTI-
 - ON

- qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_QMI_CMD
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_INVALID_QOS_ID
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_TECH_PREF
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
 - qmerrno.h, [545](#)
- eQCWWAN_ERR_QMI_INVALID_TRANSITION
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_INVALID_TX_ID
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_MALFORMED_MSG
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_MAX
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_MISSING_ARG
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NO_EFFECT
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NO_ENTRY
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NO_MEMORY
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NO_RADIO
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_OFFSET
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_REQ
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_REQ_SCH
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_REQ_TO
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_RSP
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_RSP_TO
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE
 - qmerrno.h, [544](#)

- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, [545](#)
- eQCWWAN_ERR_RESET
qmerrno.h, [542](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_END
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_START
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIDCS_END
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIDCS_START
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_END
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWIIM_START
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWISM_END
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, [545](#)
- eQCWWAN_ERR_SWISMS_START
qmerrno.h, [545](#)
- eQMI_LOC_SESS_STATUS_FAILURE
loc.h, [470](#)
- eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, [470](#)
- eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, [470](#)
- eQMI_LOC_SESS_STATUS_TIMEOUT

- loc.h, [470](#)
- eQOS
 - common.h, [429](#)
- eREQ
 - common.h, [430](#)
- eRSP
 - common.h, [430](#)
- eSMS
 - common.h, [429](#)
- eSWILOC
 - common.h, [429](#)
- eSWIOMA
 - common.h, [429](#)
- eTIMEOUT_10_S
 - common.h, [430](#)
- eTIMEOUT_20_S
 - common.h, [430](#)
- eTIMEOUT_2_S
 - common.h, [430](#)
- eTIMEOUT_300_S
 - common.h, [430](#)
- eTIMEOUT_30_S
 - common.h, [430](#)
- eTIMEOUT_5_S
 - common.h, [430](#)
- eTIMEOUT_60_S
 - common.h, [430](#)
- eTIMEOUT_8_S
 - common.h, [430](#)
- eTIMEOUT_DEFAULT
 - common.h, [430](#)
- eUIM
 - common.h, [429](#)
- eWDS
 - common.h, [429](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID-
_IDENT_FOR_PROFILE
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT-
_DEFINED
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_
_PROFILES
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFIL-
_E_FAMILY
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_F-
_LAG_SET
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_N-
_OT_SET
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_END
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMI-
_LY
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_HNDL
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_IDENT
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_OP
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_PROFILE_NUM
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_PROFILE_TYPE
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_
_SUBS_ID
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_IN-
_VALID
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NO-
_T_INITED
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL
 - qmerrno.h, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END
 - qmerrno.h, [546](#)
- eLOG_LEVEL
 - common.h, [429](#)
- EMTlv
 - NASQmiCbkNasSystemSelPrefInd, [178](#)
- eQCWWANError
 - qmerrno.h, [541](#)
- eQMI_SVC
 - common.h, [429](#)
- ESNString
 - unpack_dms_GetDeviceSerialNumbers_t, [285](#)
- ETWSPLMNInfo
 - eTWSPMLNInfoTlv, [32](#)
- eTWSPMLNInfoTlv, [32](#)
 - ETWSPLMNInfo, [32](#)
 - TlvPresent, [32](#)
- ETWSPLMNTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [374](#)
- ETWSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [374](#)
- eTimeout
 - common.h, [429](#)
- eValid
 - LibPackTFTIDParams, [66](#)
- earfcn
 - nas_infoInterFreq, [120](#)

- nas_LTEInfoIntraFreq, 126
- nas_umtsLTENbrCell, 163
- ecio
 - cdmaSSInfo, 26
 - hdrSSInfo, 37
 - nas_ecioListElement, 103
 - nas_UMTSInfo, 160
 - tdscdmaSigInfoExt, 262
- ecioDelta
 - nas_SLQSSignalStrengthsIndReq, 150
- ecioInfo
 - nas_SLQSSignalStrengthsInformation, 151
- ecioList
 - unpack_nas_SLQSGetSignalStrength_t, 328
- ecioListLen
 - unpack_nas_SLQSGetSignalStrength_t, 328
- ecioThresholdList
 - nas_SLQSSignalStrengthsIndReq, 150
- ecioThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, 150
- egprsSupp
 - nas_GSMSSysInfo, 113
- egprsSuppValid
 - nas_GSMSSysInfo, 113
- EmerMode
 - NASEmergencyModeTlv, 171
- emmConnState
 - nas_LTEInfo, 123
- emmState
 - nas_LTEInfo, 123
- emmSubState
 - nas_LTEInfo, 124
- enable
 - pack_qos_SLQSSetQosEventCallback_t, 216
- enabled
 - unpack_wds_GetMobileIPProfile_t, 396
- EncryptedPIN1
 - pack_uim_ChangePin_t, 227
 - pack_uim_SetPinProtection_t, 229
 - pack_uim_UnblockPin_t, 232
- engineState
 - unpack_loc_EngineState_Ind_t, 305
- eqmiCbkSetStatus
 - sms.h, 560
- error
 - unpack_wds_GetLastMobileIPError_t, 395
- errorRate
 - nas_errorRateListElement, 104
- errorRateInfo
 - nas_SLQSSignalStrengthsInformation, 151
- errorRateList
 - unpack_nas_SLQSGetSignalStrength_t, 328
- errorRateListLen
 - unpack_nas_SLQSGetSignalStrength_t, 328
- errorState
 - slotInf, 256
 - uim_slotInfo, 278
- esn
 - unpack_dms_GetSerialNumbers_t, 291
- esnSize
 - unpack_dms_GetDeviceSerialNumbers_t, 285
- EspSpi
 - unpack_qos_swiQosFilter_t, 364
- EtwsMessageInfo
 - sMSEtwsMessageTlv, 258
- event
 - unpack_qos_SLQSSetQosPriEventCallback_ind_t, 360
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 362
- eventMask
 - pack_uim_SLQSUIEventRegister_t, 230
 - unpack_uim_SLQSUIEventRegister_t, 389
- eventRegister
 - pack_loc_EventRegister_t, 190
- eventType
 - unpack_swioma_SLQSOMADMAAlertCallback_ind_t, 379
- executingImage
 - FMSImageIDEntries, 35
- exponent
 - unpack_qos_pktErrRate_t, 350
- ExtErrorCode
 - PackCreateProfileOut, 249
- extPowerState
 - pack_loc_SetExtPowerState_t, 190
- extendedErrorCode
 - unpack_wds_SLQSDeleteProfile_t, 399
- FLOAT
 - SwiDataTypes.h, 567
- FMSImageElement, 32
 - buildId, 33
 - buildIdLength, 33
 - imageId, 33
 - imageType, 33
- FMSImageIDEntries, 34
 - executingImage, 35
 - imageIDElement, 35
 - imageIDSize, 35
 - imageType, 35
 - maxImages, 35
- FMSImageIDElement, 33
 - buildID, 34
 - buildIDLength, 34
 - failureCount, 34
 - imageID, 34
 - storageIndex, 34
- FMSImageList, 35
 - imageIDEntries, 36
 - listSize, 36
- FMSPrefImageList, 36
 - listEntries, 36
 - listSize, 36
- FOTAUpdate
 - pack_swioma_SLQSOMADMSetSettings_t, 226
 - unpack_swioma_SLQSOMADMGetSettings_t, 384

- FOTAdownload
 - pack_swiloma_SLQSOMADMSetSettings_t, 226
 - unpack_swiloma_SLQSOMADMGetSettings_t, 384
- failureCount
 - FMSImageIdElement, 34
- family
 - pack_wds_GetDefaultProfileNum_t, 235
 - pack_wds_SetDefaultProfileNum_t, 239
- fileID
 - uim_fileInfo, 271
- fileIndex
 - pack_uim_ReadTransparent_t, 228
- fill_pack_ctx
 - common.h, 430
- fill_sdu_hdr
 - common.h, 430
- filterId
 - LibPackTFTIDParams, 66
- fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, 223
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- fix_rate_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, 223
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- fix_type_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- flowLabel
 - LibPackTFTIDParams, 66
- fms.h, 462
 - GetValidFwPriCombinations, 463
 - pack_fms_GetImagesPreference, 464
 - pack_fms_GetStoredImages, 464
 - pack_fms_SetImagesPreference, 464
 - unpack_fms_GetImagesPreference, 465
 - unpack_fms_GetStoredImages, 465
 - unpack_fms_SetImagesPreference, 465
- Forbidden
 - nas_QmiNas3GppNetworkInfo, 142
- ForceRev0
 - unpack_nas_GetCDMANetworkParameters_t, 314
- format
 - sMSTransferRouteMTMessage, 262
- fqdnAddr
 - wds_PCSCFFQDNAddress, 418
- fqdnLen
 - wds_PCSCFFQDNAddress, 418
- freq
 - NASPhyCaAggPcellInfo, 174
 - NASPhyCaAggScellIndType, 176
 - NASPhyCaAggScellInfo, 177
- freqsLen
 - nas_LTEInfoInterfreq, 124
 - nas_LTEInfoNeighboringGSM, 127
 - nas_LTEInfoNeighboringWCDMA, 128
- function
 - pack_swiloc_SwiLocSetAutoStart_t, 223
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- function_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- FwAutoCheck
 - unpack_swiloma_SLQSOMADMGetSettings_t, 384
- FwAvailability
 - unpack_swiloma_SLQSOMADMStartSession_t, 384
- fwloadsize
 - unpack_omaDmFotaTlv_t, 347
- fwloadComplete
 - unpack_omaDmFotaTlv_t, 347
- fwvers
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 301
- gDIBitRate
 - LibPackQosClassID, 64
- GPRSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, 404
- GPSPMP
 - pack_dms_SetCustFeature_t, 182
 - unpack_dms_GetCustFeature_t, 282
- GPSSel
 - pack_dms_SetCustFeature_t, 182
 - unpack_dms_GetCustFeature_t, 282
- GSMRSSIThreshListLen
 - nas_GSMRSSIThresh, 109
- GSMSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 337
- gUIBitRate
 - LibPackQosClassID, 64
- GWAOPTlv
 - NASQmiCbkNasSystemSelPrefInd, 178
- GWAcqOrderPref
 - NASGWAcqOrderPrefTlv, 172
- GWAddressV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 404
- geoSysIdx
 - nas_AddCDMASysInfo, 84
 - nas_AddSysInfo, 84
- geranArfcn
 - nas_geranInstInfo, 107
- geranBsicBcc
 - nas_geranInstInfo, 107
- geranBsicNcc
 - nas_geranInstInfo, 107
- geranInst
 - nas_UMTSInfo, 160
- GeranInstInfo
 - nas_UMTSInfo, 160
- geranRssi
 - nas_geranInstInfo, 107
- get_version
 - common.h, 430
- GetCustomFeatureV2
 - unpack_dms_GetCustFeaturesV2_t, 283
- GetValidFwPriCombinations
 - fms.h, 463
- globalCellId

- nas_LTEInfoIntrafreq, [127](#)
- glog
 - common.h, [431](#)
- gloglvl
 - common.h, [431](#)
- gnssSvUsedList
 - loc_svUsedforFix, [81](#)
- gnssSvUsedList_len
 - loc_svUsedforFix, [81](#)
- Gpp2TimeZone
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- GppNetworkDSTAdjustment
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- GppTimeZone
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- GpsEnable
 - pack_dms_SetCustFeature_t, [182](#)
 - unpack_dms_GetCustFeature_t, [282](#)
- gpsTimeOfWeekMs
 - loc_gpsTime, [76](#)
- gpsWeek
 - loc_gpsTime, [76](#)
- grntDownlinkBitrate
 - LibPackUMTSQoS, [69](#)
 - wds_UMTSMInQoS, [422](#)
- grntUplinkBitrate
 - LibPackUMTSQoS, [69](#)
 - wds_UMTSMInQoS, [422](#)
- GsmCellInfo
 - nas_lteGsmCellInfo, [121](#)
- gsmUmtsDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- gsmUmtsUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- guaranteedRate
 - unpack_qos_dataRate_t, [348](#)
- gwAddressV6
 - wds_IPV6GWAddressInfo, [417](#)
- gwV6PrefixLen
 - wds_IPV6GWAddressInfo, [417](#)
- HASPI
 - unpack_wds_GetMobileIPProfile_t, [396](#)
- HASState
 - unpack_wds_GetMobileIPProfile_t, [396](#)
- HDOP
 - loc_precisionDilution, [78](#)
- HDRECIOTreshListLen
 - nas_HDRECIOTresh, [114](#)
- HDRIOTreshListLen
 - nas_HDRIOTresh, [114](#)
- HDRRSSITreshListLen
 - nas_HDRRSSITresh, [115](#)
- HDRSINRThreshListLen
 - nas_HDRSINRThreshold, [116](#)
- HDRSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [337](#)
- HardwareControlledMode
 - unpack_dms_GetPower_t, [290](#)
- hdrActiveProt
 - nas_HDRSysInfo, [118](#)
- hdrActiveProtValid
 - nas_HDRSysInfo, [118](#)
- hdrHybrid
 - nas_detailSvcInfo, [102](#)
- HdrPersonality
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- hdrPersonality
 - nas_HDRSysInfo, [118](#)
 - NASServingSystemInfo, [180](#)
- hdrPersonalityValid
 - nas_HDRSysInfo, [118](#)
- hdrSSInfo, [36](#)
 - ecio, [37](#)
 - io, [37](#)
 - rsi, [37](#)
 - sinr, [37](#)
- hdrSrvStatus
 - nas_detailSvcInfo, [102](#)
- helper_get_resp_ctx
 - common.h, [430](#)
- helper_get_xid
 - common.h, [430](#)
- helper_set_log_func
 - common.h, [431](#)
- helper_set_log_lvl
 - common.h, [431](#)
- hotSwap
 - uim_hotSwapStatus, [272](#)
- hotSwapLength
 - uim_hotSwapStatus, [272](#)
- hsCallStatus
 - nas_WCDMASysInfo, [170](#)
- hsCallStatusValid
 - nas_WCDMASysInfo, [170](#)
- hsInd
 - nas_WCDMASysInfo, [170](#)
- hsIndValid
 - nas_WCDMASysInfo, [170](#)
- hwType
 - wdsDhcv4HwConfig, [423](#)
- hwVer
 - unpack_dms_GetHardwareRevision_t, [288](#)
- iLTEbandValue
 - NASPhyCaAggPcellInfo, [174](#)
 - NASPhyCaAggScellInfo, [177](#)
- IMCNflag
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- IMEIString
 - unpack_dms_GetDeviceSerialNumbers_t, [285](#)
- IMSInfo
 - sMSOnIMSTlv, [260](#)
- IMSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [374](#)
- INT32

- SwiDataTypes.h, [567](#)
- INT8
 - SwiDataTypes.h, [567](#)
- IPAddressV6
 - ipv6AddressInfo, [38](#)
 - wds_IPV6AddressInfo, [417](#)
- IPFamSupport
 - pack_dms_SetCustFeature_t, [182](#)
 - unpack_dms_GetCustFeature_t, [282](#)
- IPFamilyPreference
 - pack_wds_SLQSSetIPFamilyPreference_t, [246](#)
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- IPSECSPI
 - LibPackTFTIDParams, [66](#)
- IPv6AddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- IPv6GWAAddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- IPv6PrefixLen
 - ipv6AddressInfo, [38](#)
 - wds_IPV6AddressInfo, [417](#)
- IPv4
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- IPv4DstAddr
 - unpack_qos_swiQosFilter_t, [364](#)
- IPv4SrcAddr
 - unpack_qos_swiQosFilter_t, [364](#)
- IPv4Tos
 - unpack_qos_swiQosFilter_t, [364](#)
- IPv6DstAddr
 - unpack_qos_swiQosFilter_t, [364](#)
- IPv6Label
 - unpack_qos_swiQosFilter_t, [364](#)
- IPv6SrcAddr
 - unpack_qos_swiQosFilter_t, [364](#)
- IPv6TrafCls
 - unpack_qos_swiQosFilter_t, [364](#)
- Id
 - unpack_qos_swiQosFilter_t, [364](#)
- id
 - loc_BdsSV, [70](#)
 - loc_SV, [80](#)
 - nas_CSGID, [98](#)
 - unpack_qos_QosFlowInfoState_t, [353](#)
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, [362](#)
- id_length
 - DMScustSettingInfo, [29](#)
- image_info_t, [37](#)
 - buildID, [37](#)
 - buildIDLen, [37](#)
 - imageType, [37](#)
 - uniqueID, [37](#)
- imageID
 - FMSImageIDElement, [34](#)
- imageIDElement
 - FMSImageIDEntries, [35](#)
- imageIDEntries
 - FMSImageList, [36](#)
- imageIDSize
 - FMSImageIDEntries, [35](#)
- imageId
 - FMSImageElement, [33](#)
- imageList
 - unpack_fms_GetStoredImages_t, [303](#)
- ImageListSize
 - unpack_fms_GetImagesPreference_t, [303](#)
- imageListSize
 - pack_fms_SetImagesPreference_t, [187](#)
- imageType
 - FMSImageElement, [33](#)
 - FMSImageIDEntries, [35](#)
 - image_info_t, [37](#)
- ImageTypes
 - unpack_fms_SetImagesPreference_t, [304](#)
- ImageTypesSize
 - unpack_fms_SetImagesPreference_t, [304](#)
- imagelistSize
 - unpack_fms_GetStoredImages_t, [303](#)
- imei_no
 - unpack_dms_GetSerialNumbers_t, [291](#)
- imeiSize
 - unpack_dms_GetDeviceSerialNumbers_t, [285](#)
- imeiSvnSize
 - unpack_dms_GetDeviceSerialNumbers_t, [285](#)
- ImeiSvnString
 - unpack_dms_GetDeviceSerialNumbers_t, [285](#)
- imeisv_svn
 - unpack_dms_GetSerialNumbers_t, [291](#)
- imsRegState
 - nas_CommInfo, [97](#)
- imsi
 - unpack_dms_GetIMSI_t, [288](#)
- imsi_11_12
 - nas_CDMA SysInfoExt, [94](#)
- InUse
 - nas_QmiNas3GppNetworkInfo, [142](#)
- includes_pcs_digit
 - nas_QmisNasPcsDigit, [144](#)
- index
 - pack_wds_GetMobileIPProfile_t, [237](#)
 - pack_wds_SetDefaultProfileNum_t, [239](#)
 - pack_wds_SetMobileIPProfile_t, [239](#)
 - unpack_qos_swiQosFilter_t, [364](#)
 - unpack_qos_swiQosFlow_t, [369](#)
 - unpack_wds_GetDefaultProfileNum_t, [394](#)
- index1xPri
 - uim_cardStatus, [269](#)
- index1xSec
 - uim_cardStatus, [269](#)
- indexGwPri
 - uim_cardStatus, [269](#)
- indexGwSec
 - uim_cardStatus, [269](#)
- Info

- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 339
- unpack_nas_SLQSSetSysSelectionPrefCallback_ind_t, 340
- InfoInterfreq
 - nas_LTEInfoInterfreq, 124
- insNmrCellInfo
 - nas_GERANInfo, 106
- instancesSize
 - unpack_nas_GetRFInfo_t, 318
- interval
 - pack_wds_SLQSSetWdsEventCallback_t, 246
- Io
 - unpack_nas_SLQSGetSignalStrength_t, 328
- io
 - hdrSSInfo, 37
 - nas_SLQSSignalStrengthsInformation, 151
- ioDelta
 - nas_SLQSSignalStrengthsIndReq, 150
- ipAddress
 - pack_wds_SetDefaultProfile_t, 238
- ipFamily
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 406
- ipVersion
 - LibPackTFTIDParams, 66
- ipaddr
 - unpack_wds_GetDefaultProfile_t, 394
- ipaddrv6
 - unpack_wds_GetDefaultProfile_t, 394
- ipv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 410
- ipv4GWAddress
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 410
- ipv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 410
- ipv6AddressInfo, 37
 - IPAddressV6, 38
 - IPv6PrefixLen, 38
- ipv6GWAddress
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 411
- is856SysId
 - nas_HDRSysInfo, 118
- is856SysIdValid
 - nas_HDRSysInfo, 118
- is_DataRate_Available
 - unpack_qos_swiQosFlow_t, 369
- is_EspSpi_Available
 - unpack_qos_swiQosFilter_t, 364
- is_IPv4DstAddr_Available
 - unpack_qos_swiQosFilter_t, 364
- is_IPv4SrcAddr_Available
 - unpack_qos_swiQosFilter_t, 365
- is_IPv4Tos_Available
 - unpack_qos_swiQosFilter_t, 365
- is_IPv6DstAddr_Available
 - unpack_qos_swiQosFilter_t, 365
- is_IPv6Label_Available
 - unpack_qos_swiQosFilter_t, 365
- is_IPv6SrcAddr_Available
 - unpack_qos_swiQosFilter_t, 365
- is_IPv6TrafCls_Available
 - unpack_qos_swiQosFilter_t, 365
- is_Id_Available
 - unpack_qos_swiQosFilter_t, 364
- is_Jitter_Available
 - unpack_qos_swiQosFlow_t, 369
- is_Latency_Available
 - unpack_qos_swiQosFlow_t, 369
- is_LteBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, 298
- is_LteQci_Available
 - unpack_qos_swiQosFlow_t, 369
- is_MaxAllowedPktSz_Available
 - unpack_qos_swiQosFlow_t, 369
- is_MinPolicedPktSz_Available
 - unpack_qos_swiQosFlow_t, 369
- is_NxtHdrProto_Available
 - unpack_qos_swiQosFilter_t, 365
- is_PktErrRate_Available
 - unpack_qos_swiQosFlow_t, 369
- is_Precedence_Available
 - unpack_qos_swiQosFilter_t, 365
- is_ProfileId3GPP2_Available
 - unpack_qos_swiQosFlow_t, 369
- is_RxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, 352
- is_TCPDstPort_Available
 - unpack_qos_swiQosFilter_t, 365
- is_TCPSrcPort_Available
 - unpack_qos_swiQosFilter_t, 365
- is_TdsBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, 298
- is-TokenBucket_Available
 - unpack_qos_swiQosFlow_t, 369
- is_TrafficClass_Available
 - unpack_qos_swiQosFlow_t, 369
- is_TranDstPort_Available
 - unpack_qos_swiQosFilter_t, 365
- is_TranSrcPort_Available
 - unpack_qos_swiQosFilter_t, 365
- is_TxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, 352
- is_UDPDstPort_Available
 - unpack_qos_swiQosFilter_t, 365
- is_UDPSrcPort_Available
 - unpack_qos_swiQosFilter_t, 365
- is_val_3GPP2Pri_Available
 - unpack_qos_swiQosFlow_t, 369
- is_val_3GPPIImCn_Available
 - unpack_qos_swiQosFlow_t, 369
- is_val_3GPPResResidualBER_Available

- unpack_qos_swiQosFlow_t, [369](#)
- is_val_3GPPSigInd_Available
 - unpack_qos_swiQosFlow_t, [369](#)
- is_val_3GPPTraHdlPri_Available
 - unpack_qos_swiQosFlow_t, [369](#)
- isNewFlow
 - unpack_qos_QosFlowInfoState_t, [353](#)
- isPrefDataPath
 - nas_GSMSrvStatusInfo, [110](#)
 - nas_SrvStatusInfo, [153](#)
- isSysForbidden
 - nas_detailSvcInfo, [102](#)
 - nas_sysInfoCommon, [156](#)
- isSysForbiddenValid
 - nas_sysInfoCommon, [156](#)
- isSysPriMatch
 - nas_CDMASysInfo, [93](#)
 - nas_HDRSysInfo, [118](#)
- isSysPriMatchValid
 - nas_CDMASysInfo, [93](#)
 - nas_HDRSysInfo, [118](#)
- IsVoiceEnabled
 - pack_dms_SetCustFeature_t, [182](#)
 - unpack_dms_GetCustFeature_t, [282](#)
- Jitter
 - unpack_qos_swiQosFlow_t, [370](#)
- LIBPACK_QMI_CBK_PARAM_NOCHANGE
 - sms.h, [560](#)
- LIBPACK_QMI_CBK_PARAM_RESET
 - sms.h, [560](#)
- LIBPACK_QMI_CBK_PARAM_SET
 - sms.h, [560](#)
- LBPTlv
 - NASQmiCbkNasSystemSelPrefInd, [178](#)
- LOCEVENTMASKGNSSSVINFO
 - loc.h, [468](#)
- LOCEVENTMASKNMEA
 - loc.h, [469](#)
- LOCEVENTMASKWIFIREQ
 - loc.h, [469](#)
- LPCSTR
 - SwiDataTypes.h, [567](#)
- LTEAttachProfileList
 - pack_wds_SLQSSet3GPPConfigItem_t, [245](#)
 - unpack_wds_SLQSGet3GPPConfigItem_t, [400](#)
- LTEAttachProfileListLen
 - pack_wds_SLQSSet3GPPConfigItem_t, [245](#)
 - unpack_wds_SLQSGet3GPPConfigItem_t, [400](#)
- LTEBandPref
 - NASLTEBandPreferenceTlv, [172](#)
- LTECphyCAInfo
 - unpack_nas_SlqsGetLTECphyCAInfo_t, [324](#)
- LTERSRRPThreshListLen
 - nas_LTERSRRPThresh, [129](#)
- LTERSRRQThreshListLen
 - nas_LTERSRRQThresh, [130](#)
- LTERSSIThreshListLen
 - nas_LTERSSIThresh, [130](#)
- LTESNRThreshListLen
 - nas_LTESNRThreshold, [133](#)
- LTESInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [337](#)
- LTEWCDMACellInfo
 - nas_LTEInfoNeighboringWCDMA, [128](#)
- Lac
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- lac
 - nas_GERANInfo, [106](#)
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_UMTSInfo, [160](#)
 - nas_WCDMASysInfo, [170](#)
- lacValid
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_WCDMASysInfo, [170](#)
- lastCallDataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, [402](#)
- lastCallDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, [401](#)
- lastCallRXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, [402](#)
- lastCallTXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, [402](#)
- Latency
 - unpack_qos_swiQosFlow_t, [370](#)
- leapSeconds
 - nas_qaQmi3Gpp2TimeZone, [141](#)
- len
 - loc_BdsSVInfo, [71](#)
 - loc_SVInfo, [81](#)
 - unpack_nas_GetSignalStrengths_t, [320](#)
- length
 - sMSCAddress, [257](#)
 - sMSEtwSMessage, [258](#)
 - sMSTransferRouteMTMessage, [262](#)
 - uim_readTransparentInfo, [273](#)
- LibPackGPRSRequestedQoS, [38](#)
 - delayClass, [38](#)
 - meanThroughputClass, [38](#)
 - peakThroughputClass, [38](#)
 - precedenceClass, [38](#)
 - reliabilityClass, [38](#)
- LibPackQosClassID, [63](#)
 - gDIBitRate, [64](#)
 - gUIBitRate, [64](#)
 - maxDIBitRate, [64](#)
 - maxUIBitRate, [64](#)
 - QCI, [64](#)
- LibPackTFTIDParams, [64](#)
 - destPortRangeEnd, [66](#)
 - destPortRangeStart, [66](#)
 - eValid, [66](#)
 - filterId, [66](#)

- flowLabel, 66
- IPSECSPi, 66
- ipVersion, 66
- nextHeader, 66
- pSourceIP, 66
- sourceIPMask, 66
- srcPortRangeEnd, 66
- srcPortRangeStart, 66
- tosMask, 66
- LibPackUMTSQoS, 66
 - deliveryErrSDU, 69
 - grntDownlinkBitrate, 69
 - grntUplinkBitrate, 69
 - maxDownlinkBitrate, 69
 - maxSDUSize, 69
 - maxUplinkBitrate, 69
 - qosDeliveryOrder, 69
 - resBerRatio, 69
 - sduErrorRatio, 69
 - trafficClass, 69
 - trafficPriority, 69
 - transferDelay, 69
- LibPackUMTSReqQoSsigInd, 69
 - SigInd, 70
 - UMTSReqQoS, 70
- LibPackprofile_3GPP, 51
 - pAPNClass, 56
 - pAPNDisabledFlag, 56
 - pAPNName, 56
 - pAPNnameSize, 56
 - pAddrAllocPref, 56
 - pAuthenticationPref, 56
 - pGPRSMinimumQoS, 56
 - pGPRSRequestedQoS, 56
 - pIPv4AddrPref, 56
 - pIPv6AddPref, 56
 - plmCnFlag, 56
 - pPDNInactivTimeout, 56
 - pPDType, 57
 - pPassword, 56
 - pPasswordSize, 56
 - pPcscfAddrUsingDhcp, 56
 - pPcscfAddrUsingPCO, 56
 - pPdpAccessConFlag, 56
 - pPdpContext, 56
 - pPdpDataCompType, 57
 - pPdpHdrCompType, 57
 - pPriDNSIPv4AddPref, 57
 - pPriDNSIPv6addpref, 57
 - pPrimaryID, 57
 - pProfileName, 57
 - pProfileNameSize, 57
 - pQoSClassID, 57
 - pSecDNSIPv4AddPref, 57
 - pSecDNSIPv6addpref, 57
 - pSecondaryFlag, 57
 - pTFTID1Params, 57
 - pTFTID2Params, 57
 - pUMTSMinQoS, 57
 - pUMTSMinQoSsigInd, 57
 - pUMTSReqQoS, 57
 - pUMTSReqQoSsigInd, 57
 - pUsername, 57
 - pUsernameSize, 57
- LibPackprofile_3GPP2, 57
 - pAPNClass3GPP2, 62
 - pAPNEnabled3GPP2, 62
 - pAllowLinger, 62
 - pApnString, 62
 - pApnStringSize, 62
 - pAppPriority, 62
 - pAppType, 62
 - pAuthPassword, 62
 - pAuthPassword_tSize, 62
 - pAuthProtocol, 62
 - pAuthRetryCount, 62
 - pAuthTimeout, 62
 - pDataMode, 63
 - pDataRate, 63
 - plpcpAckTimeout, 63
 - plpcpCreqRetryCount, 63
 - plsPcscfAddressNedded, 63
 - pLcpAckTimeout, 63
 - pLcpCreqRetryCount, 63
 - pNegoDnsSrvrPref, 63
 - pPDNInactivTimeout3GPP2, 63
 - pPdnType, 63
 - pPppSessCloseTimer1x, 63
 - pPppSessCloseTimerDO, 63
 - pPriV6DnsAddress, 63
 - pPrimaryV4DnsAddress, 63
 - pRATType, 63
 - pSecV6DnsAddress, 63
 - pSecondaryV4DnsAddress, 63
 - pUserId, 63
 - pUserIdSize, 63
- libpack_log
 - common.h, 431
- LibpackProfile3GPP, 38
 - pAPNClass, 43
 - pAPNDisabledFlag, 43
 - pAPNName, 43
 - pAPNnameSize, 44
 - pAddrAllocPref, 43
 - pAuthenticationPref, 44
 - pGPRSMinimumQoS, 44
 - pGPRSRequestedQoS, 44
 - pIPv4AddrPref, 44
 - pIPv6AddPref, 44
 - plmCnFlag, 44
 - pPDNInactivTimeout, 44
 - pPDType, 44
 - pPassword, 44
 - pPasswordSize, 44
 - pPcscfAddrUsingDhcp, 44
 - pPcscfAddrUsingPCO, 44

- pPdpAccessConFlag, [44](#)
- pPdpContext, [44](#)
- pPdpDataCompType, [44](#)
- pPdpHdrCompType, [44](#)
- pPriDNSIPv4AddPref, [44](#)
- pPriDNSIPv6addpref, [44](#)
- pPrimaryID, [44](#)
- pProfilename, [44](#)
- pProfilenameSize, [44](#)
- pQosClassID, [44](#)
- pSecDNSIPv4AddPref, [44](#)
- pSecDNSIPv6addpref, [44](#)
- pSecondaryFlag, [44](#)
- pTFTID1Params, [44](#)
- pTFTID2Params, [44](#)
- pUMTSMinQoS, [45](#)
- pUMTSMinQoSsigInd, [45](#)
- pUMTSReqQoS, [45](#)
- pUMTSReqQoSsigInd, [45](#)
- pUsername, [45](#)
- pUsernameSize, [45](#)
- LibpackProfile3GPP2, [45](#)
 - pAPNClass3GPP2, [50](#)
 - pAPNEnabled3GPP2, [50](#)
 - pAllowLinger, [50](#)
 - pApnString, [50](#)
 - pApnStringSize, [50](#)
 - pAppPriority, [50](#)
 - pAppType, [50](#)
 - pAuthPassword, [50](#)
 - pAuthPasswordSize, [50](#)
 - pAuthProtocol, [50](#)
 - pAuthRetryCount, [50](#)
 - pAuthTimeout, [50](#)
 - pDataMode, [50](#)
 - pDataRate, [50](#)
 - plpcpAckTimeout, [50](#)
 - plpcpCreqRetryCount, [50](#)
 - plsPscsfAddressNedded, [50](#)
 - pLcpAckTimeout, [50](#)
 - pLcpCreqRetryCount, [50](#)
 - pNegoDnsSrvrPref, [50](#)
 - pPDNInactivTimeout3GPP2, [50](#)
 - pPdnType, [50](#)
 - pPppSessCloseTimer1x, [50](#)
 - pPppSessCloseTimerDO, [51](#)
 - pPriV6DnsAddress, [51](#)
 - pPrimaryV4DnsAddress, [51](#)
 - pRATType, [51](#)
 - pSecV6DnsAddress, [51](#)
 - pSecondaryV4DnsAddress, [51](#)
 - pUserId, [51](#)
 - pUserIdSize, [51](#)
- list_type
 - DMScustSettingList, [30](#)
 - DMSgetCustomInput, [31](#)
 - pack_dms_GetCustFeaturesV2_t, [182](#)
- listEntries
 - FMSPrefImageList, [36](#)
- listSize
 - FMSImageList, [36](#)
 - FMSPrefImageList, [36](#)
- loc.h
 - eQMI_LOC_SESS_STATUS_FAILURE, [470](#)
 - eQMI_LOC_SESS_STATUS_IN_PROGRESS, [470](#)
 - eQMI_LOC_SESS_STATUS_SUCCESS, [470](#)
 - eQMI_LOC_SESS_STATUS_TIMEOUT, [470](#)
- loc.h, [465](#)
 - LOCEVENTMASKNMEA, [469](#)
 - LOCEVENTMASKWIFIREQ, [469](#)
 - pack_loc_DeleteAssistData, [470](#)
 - pack_loc_EventRegister, [470](#)
 - pack_loc_SetExtPowerState, [470](#)
 - pack_loc_SetOperationMode, [471](#)
 - pack_loc_Start, [471](#)
 - pack_loc_Stop, [471](#)
 - unpack_loc_DeleteAssistData, [472](#)
 - unpack_loc_EngineState_Ind, [472](#)
 - unpack_loc_EventRegister, [472](#)
 - unpack_loc_PositionRpt_Ind, [473](#)
 - unpack_loc_SetExtPowerState, [473](#)
 - unpack_loc_SetOperationMode, [473](#)
 - unpack_loc_Start, [474](#)
 - unpack_loc_Stop, [474](#)
- loc_BdsSV, [70](#)
 - id, [70](#)
 - mask, [70](#)
- loc_BdsSVInfo, [70](#)
 - len, [71](#)
 - pSV, [71](#)
- loc_CellDb, [71](#)
 - mask, [72](#)
- loc_ClkInfo, [72](#)
 - mask, [73](#)
- loc_GnssData, [74](#)
 - mask, [76](#)
- loc_LocApplicationInfo, [76](#)
 - appNameLength, [77](#)
 - appProviderLength, [77](#)
 - appVersionLength, [77](#)
 - appVersionValid, [77](#)
 - pAppName, [77](#)
 - pAppProvider, [77](#)
 - pAppVersion, [77](#)
- loc_SV, [79](#)
 - id, [80](#)
 - mask, [80](#)
 - system, [80](#)
- loc_SVInfo, [80](#)
 - len, [81](#)
 - pSV, [81](#)
- loc_gpsTime, [76](#)
 - gpsTimeOfWeekMs, [76](#)
 - gpsWeek, [76](#)
- loc_precisionDilution, [77](#)

- HDOP, [78](#)
- PDOP, [78](#)
- VDOP, [78](#)
- loc_sensorDataUsage, [78](#)
 - aidingIndicatorMask, [79](#)
 - usageMask, [79](#)
- loc_svUsedforFix, [81](#)
 - gnssSvUsedList, [81](#)
 - gnssSvUsedList_len, [81](#)
- localTimeOffset
 - nas_qaQmi3Gpp2TimeZone, [141](#)
- logger
 - common.h, [429](#)
- longName
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- longNameCI
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- longNameEn
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- longNameLen
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- longNameSB
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- LteBandCapability
 - unpack_dms_SLQSGetBandCapability_t, [298](#)
- lteEmmDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- lteEmmUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- lteEsmDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- lteEsmUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- LteGsmCellInfo
 - nas_LTEInfoNeighboringGSM, [127](#)
- LteQci
 - unpack_qos_swiQosFlow_t, [370](#)
- lteRsrpDelta
 - nas_SLQSSignalStrengthsIndReq, [150](#)
- lteRsrpinfo
 - nas_SLQSSignalStrengthsInformation, [151](#)
- lteSSInfo, [81](#)
 - rsrp, [82](#)
 - rsrq, [82](#)
 - rsqi, [82](#)
 - snr, [82](#)
- lteSnrDelta
 - nas_SLQSSignalStrengthsIndReq, [151](#)
- lteSnrinfo
 - nas_SLQSSignalStrengthsInformation, [151](#)
- ltersrp
 - unpack_nas_SLQSGetSignalStrength_t, [328](#)
- ltersnr
 - unpack_nas_SLQSGetSignalStrength_t, [328](#)
- m_FwBuildId
 - CarrierImage_t, [25](#)
- m_FwImageld
 - CarrierImage_t, [26](#)
- m_PriBuildId
 - CarrierImage_t, [26](#)
- m_Prilmageld
 - CarrierImage_t, [26](#)
- m_nCarrierId
 - CarrierImage_t, [26](#)
- m_nFolderId
 - CarrierImage_t, [26](#)
- m_nStorage
 - CarrierImage_t, [26](#)
- MAX_BUILD_ID_LEN
 - dms.h, [435](#)
- MAX_ICCID_LENGTH
 - uim.h, [580](#)
- MAX_MSE_TWS_MSG
 - sms.h, [558](#)
- MAX_NO_OF_SLOTS
 - uim.h, [580](#)
- MAX_SLOTS_STATUS
 - uim.h, [580](#)
- MAX_SMS_LIST_SIZE
 - sms.h, [559](#)
- MCC
 - nas_CDMASysInfo, [93](#)
 - nas_CDMASysInfoExt, [94](#)
 - nas_currentPLMN, [99](#)
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_QmiNas3GppNetworkInfo, [142](#)
 - nas_QmiNas3GppNetworkRAT, [143](#)
 - nas_QmisNasPcsDigit, [144](#)
 - nas_WCDMASysInfo, [170](#)
 - unpack_nas_GetServingNetwork_t, [319](#)
- MDMCallDuration
 - connectionStatus, [27](#)
- MDMConnStatus
 - connectionStatus, [27](#)
- MEIDString
 - unpack_dms_GetDeviceSerialNumbers_t, [286](#)
- MIN
 - unpack_dms_GetVoiceNumber_t, [291](#)
- MINREQBKLEN
 - common.h, [428](#)
- MMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [374](#)
- MNC
 - nas_CDMASysInfo, [93](#)
 - nas_currentPLMN, [99](#)
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_QmiNas3GppNetworkInfo, [142](#)
 - nas_QmiNas3GppNetworkRAT, [143](#)
 - nas_QmisNasPcsDigit, [144](#)
 - nas_WCDMASysInfo, [170](#)

- unpack_nas_GetServingNetwork_t, 319
- MPTlv
 - NASQmiCbkNasSystemSelPrefInd, 178
- MSGID_AND_LEN
 - common.h, 429
- MSGID_DONT_CARE
 - common.h, 429
- MTMessageInfo
 - newMTMessageTlv, 181
- Mask
 - pack_wds_SLQSGetDUNCallInfo_t, 242
- mask
 - loc_BdsSV, 70
 - loc_CellDb, 72
 - loc_ClkInfo, 73
 - loc_GnssData, 76
 - loc_SV, 80
 - unpack_qos_IPv6TrafCls_t, 350
 - unpack_qos_Tos_t, 371
- max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, 223
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- max_dist_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- max_time
 - pack_swiloc_SwiLocSetAutoStart_t, 223
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- max_time_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 378
- MaxAllowedPktSz
 - unpack_qos_swiQosFlow_t, 370
- MaxChanRxRate
 - dunchannelRate, 32
- MaxChanTxRate
 - dunchannelRate, 32
- maxChannelRXRate
 - unpack_wds_GetConnectionRate_t, 393
- maxChannelTXRate
 - unpack_wds_GetConnectionRate_t, 393
- maxDIBitRate
 - LibPackQosClassID, 64
- maxDownlinkBitrate
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- maxImages
 - FMSImageIDEntries, 35
- MaxRXChannelRate
 - unpack_dms_GetDeviceCap_t, 283
- maxRxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, 284
- maxSDUSize
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- MaxTXChannelRate
 - unpack_dms_GetDeviceCap_t, 283
- maxTxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, 284
- maxUIBitRate
 - LibPackQosClassID, 64
- maxUplinkBitrate
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- mcc
 - nas_CSGID, 98
 - nas_MNRInfo, 138
 - nas_netSelectionPref, 138
 - pack_nas_SLQSGetPLMNName_t, 197
 - unpack_nas_GetHomeNetwork_t, 315
- mdmCallDurationActive
 - unpack_wds_SLQSGetDUNCallInfo_t, 402
- meanThroughputClass
 - LibPackGPRSRequestedQoS, 38
 - wds_GPRSQoS, 416
- meid
 - unpack_dms_GetSerialNumbers_t, 291
- meidSize
 - unpack_dms_GetDeviceSerialNumbers_t, 285
- message
 - unpack_sms_SLQSGetSMS_t, 375
- message_type
 - NASOTAMessageTlv, 173
- messageFailureCode
 - unpack_sms_SendSMS_t, 372
- messageFormat
 - pack_sms_SendSMS_t, 216
 - unpack_sms_SLQSGetSMS_t, 375
- messageID
 - unpack_sms_SendSMS_t, 372
- messageIndex
 - pack_sms_SLQSGetSMS_t, 219
 - pack_sms_SLQSMModifySMSStatus_t, 221
 - qmiSmsMessageList, 251
 - sMSMTMessage, 259
- messageList
 - unpack_sms_SLQSGetSMSList_t, 376
- messageListSize
 - unpack_sms_SLQSGetSMSList_t, 376
- messageMode
 - sMSMessageMode, 259
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind-
_t, 376
- MessageModelInfo
 - messageModeTlv, 82
- messageModeTlv, 82
 - MessageModelInfo, 82
 - TlvPresent, 82
- messageSize
 - pack_sms_SendSMS_t, 216
 - unpack_sms_SLQSGetSMS_t, 375
- messageTag
 - pack_sms_SLQSMModifySMSStatus_t, 221
 - qmiSmsMessageList, 251
 - unpack_sms_SLQSGetSMS_t, 375
- MinPolicedPktSz
 - unpack_qos_swiQosFlow_t, 370
- minSize

- unpack_dms_GetVoiceNumber_t, 291
- mipMode
 - unpack_wds_GetMobileIP_t, 395
- mipStatus
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- mipstatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- mnc
 - nas_CSGID, 98
 - nas_MNRInfo, 138
 - nas_netSelectionPref, 138
 - pack_nas_SLQSGetPLMNName_t, 197
 - unpack_nas_GetHomeNetwork_t, 315
- mncPcsDigits
 - nas_CSGID, 98
- mobileCountryCode
 - sMSEtwsPlmn, 259
- mobileIP
 - pack_wds_SLQSSetWdsEventCallback_t, 247
- mobileNetworkCode
 - sMSEtwsPlmn, 259
- mode
 - pack_dms_SetEventReport_t, 183
 - pack_dms_SetPower_t, 184
 - pack_loc_SetOperationMode_t, 191
- ModePref
 - NASModePreferenceTlv, 172
- modelid
 - unpack_dms_GetModelID_t, 289
- modelid_str
 - unpack_dms_GetFirmwareInfo_t, 286
- modemMode
 - nas_CommInfo, 97
- modemindex
 - pack_fms_SetImagesPreference_t, 187
- msgid
 - pack_qmi_t, 213
 - unpack_qmi_t, 348
- msgtype
 - common.h, 430
- Mtu
 - unpack_wds_SLQSGetRuntimeSettings_t, 404
- multiplier
 - unpack_qos_pktErrRate_t, 350
- NAI
 - unpack_wds_GetMobileIPProfile_t, 396
- NAS_PLMN_LENGTH
 - nas.h, 480
- NASBandPreferenceTlv, 170
 - band_pref, 171
 - TlvPresent, 171
- NASEmergencyModeTlv, 171
 - EmerMode, 171
 - TlvPresent, 171
- NASGWAcqOrderPrefTlv, 171
 - GWAcqOrderPref, 172
- TlvPresent, 172
- NASLTEBandPreferenceTlv, 172
 - LTEBandPref, 172
 - TlvPresent, 172
- NASLteNasReleaseInfoTlv, 172
 - nas_major, 172
 - nas_minor, 172
 - nas_release, 172
 - TlvPresent, 172
- NASModePreferenceTlv, 172
 - ModePref, 172
 - TlvPresent, 172
- NASNetSelPreferenceTlv, 173
 - NetSelPref, 173
 - TlvPresent, 173
- NASOTAMessageTlv, 173
 - data_buf, 173
 - data_len, 173
 - message_type, 173
 - TlvPresent, 173
- NASPRLPreferenceTlv, 177
 - PRLPref, 177
 - TlvPresent, 177
- NASPhyCaAggPcellInfo, 173
 - dl_bw_value, 174
 - freq, 174
 - iLTEbandValue, 174
 - pci, 174
 - TlvPresent, 174
- NASPhyCaAggScellIDIBw, 174
 - dl_bw_value, 175
 - TlvPresent, 175
- NASPhyCaAggScellIndType, 175
 - freq, 176
 - pci, 176
 - scell_state, 176
 - TlvPresent, 176
- NASPhyCaAggScellIndex, 175
 - scell_idx, 175
 - TlvPresent, 175
- NASPhyCaAggScellInfo, 176
 - dl_bw_value, 177
 - freq, 177
 - iLTEbandValue, 177
 - pci, 177
 - scell_state, 177
 - TlvPresent, 177
- NASQmiCbkNasSwiOTAMessageInd, 177
 - nasRelInfoTlv, 178
 - otaMsgTlv, 178
 - timeTlv, 178
- NASQmiCbkNasSystemSelPrefInd, 178
 - BPTlv, 178
 - EMTlv, 178
 - GWAOPTlv, 178
 - LBPTlv, 178
 - MPTlv, 178
 - NSPTlv, 178

- PRLPTlv, [178](#)
- RPTlv, [178](#)
- SDPTlv, [178](#)
- NASRoamPreferenceTlv, [178](#)
 - RoamPref, [179](#)
 - TlvPresent, [179](#)
- NASServDomainPrefTlv, [179](#)
 - SrvDomainPref, [179](#)
 - TlvPresent, [179](#)
- NASServingSystemInfo, [179](#)
 - csAttachState, [180](#)
 - hdrPersonality, [180](#)
 - psAttachState, [180](#)
 - radiInterfaceList, [180](#)
 - radiInterfaceNo, [180](#)
 - registrationState, [181](#)
 - selectedNetwork, [181](#)
- NASTimeInfoTlv, [181](#)
 - time, [181](#)
 - TlvPresent, [181](#)
- NSPTlv
 - NASQmiCbkNasSystemSelPrefInd, [178](#)
- NWQoSStatus
 - unpack_qos_SLQSQosGetNetworkStatus_t, [355](#)
- naiSize
 - unpack_wds_GetMobileIPProfile_t, [396](#)
- Name
 - unpack_nas_GetServingNetwork_t, [319](#)
- name
 - unpack_nas_GetHomeNetwork_t, [315](#)
 - unpack_wds_GetDefaultProfile_t, [394](#)
- nameSize
 - unpack_nas_GetServingNetwork_t, [319](#)
- namelength
 - unpack_omaDmFotaTlv_t, [347](#)
- namesize
 - unpack_wds_GetDefaultProfile_t, [394](#)
- nas.h
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_C-CONFIGURED_ACTIVATED, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_C-CONFIGURED_DEACTIVATED, [480](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_D-ECONFIGURED, [480](#)
- nas.h, [474](#)
 - NAS_PLMN_LENGTH, [480](#)
 - pack_nas_GetACCOLC, [480](#)
 - pack_nas_GetANAAAAAuthenticationStatus, [480](#)
 - pack_nas_GetCDMANetworkParameters, [481](#)
 - pack_nas_GetHomeNetwork, [481](#)
 - pack_nas_GetNetworkPreference, [481](#)
 - pack_nas_GetRFInfo, [482](#)
 - pack_nas_GetServingNetwork, [482](#)
 - pack_nas_GetServingNetworkCapabilities, [482](#)
 - pack_nas_GetSignalStrengths, [482](#)
 - pack_nas_PerformNetworkScan, [484](#)
 - pack_nas_SLQSGetPLMNName, [485](#)
 - pack_nas_SLQSGetServingSystem, [486](#)
 - pack_nas_SLQSGetSignalStrength, [486](#)
 - pack_nas_SLQSGetSysInfo, [486](#)
 - pack_nas_SLQSGetSysSelectionPref, [487](#)
 - pack_nas_SLQSIInitiateNetworkRegistration, [487](#)
 - pack_nas_SLQSNasConfigSigInfo2, [487](#)
 - pack_nas_SLQSNasGetCellLocationInfo, [488](#)
 - pack_nas_SLQSNasGetSigInfo, [488](#)
 - pack_nas_SLQSNasIndicationRegisterExt, [488](#)
 - pack_nas_SLQSNasSwtModemStatus, [489](#)
 - pack_nas_SLQSNasSwtOTAMessageCallback, [489](#)
 - pack_nas_SLQSSetBandPreference, [489](#)
 - pack_nas_SLQSSetSignalStrengthsCallback, [490](#)
 - pack_nas_SLQSSetSysSelectionPref, [490](#)
 - pack_nas_SLQSSwtGetLteCQI, [490](#)
 - pack_nas_SetACCOLC, [484](#)
 - pack_nas_SetLURejectCallback, [484](#)
 - pack_nas_SetNetworkPreference, [485](#)
 - pack_nas_SetRFInfoCallback, [485](#)
 - pack_nas_SlqsGetLTECphyCAInfo, [485](#)
 - unpack_nas_GetACCOLC, [491](#)
 - unpack_nas_GetANAAAAAuthenticationStatus, [491](#)
 - unpack_nas_GetCDMANetworkParameters, [491](#)
 - unpack_nas_GetHomeNetwork, [492](#)
 - unpack_nas_GetNetworkPreference, [492](#)
 - unpack_nas_GetRFInfo, [492](#)
 - unpack_nas_GetServingNetwork, [492](#)
 - unpack_nas_GetServingNetworkCapabilities, [493](#)
 - unpack_nas_GetSignalStrengths, [493](#)
 - unpack_nas_PerformNetworkScan, [493](#)
 - unpack_nas_SLQSGetPLMNName, [495](#)
 - unpack_nas_SLQSGetServingSystem, [496](#)
 - unpack_nas_SLQSGetSignalStrength, [496](#)
 - unpack_nas_SLQSGetSysInfo, [496](#)
 - unpack_nas_SLQSGetSysSelectionPref, [497](#)
 - unpack_nas_SLQSIInitiateNetworkRegistration, [497](#)
 - unpack_nas_SLQSNasConfigSigInfo2, [497](#)
 - unpack_nas_SLQSNasGetCellLocationInfo, [498](#)
 - unpack_nas_SLQSNasGetSigInfo, [498](#)
 - unpack_nas_SLQSNasIndicationRegisterExt, [498](#)
 - unpack_nas_SLQSNasSigInfoCallback, [499](#)
 - unpack_nas_SLQSNasSwtModemStatus, [499](#)
 - unpack_nas_SLQSNasSwtOTAMessageCallback, [499](#)

- unpack_nas_SLQSNasSwiOTAMessageCallback_ind, 500
- unpack_nas_SLQSNasSysInfoCallback, 500
- unpack_nas_SLQSSetBandPreference, 500
- unpack_nas_SLQSSetSignalStrengthsCallback, 501
- unpack_nas_SLQSSetSysSelectionPref, 501
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind, 501
- unpack_nas_SLQSSwiGetLteCQI, 501
- unpack_nas_SetACCOLC, 494
- unpack_nas_SetDataCapabilitiesCallback_ind, 494
- unpack_nas_SetEventReportInd, 494
- unpack_nas_SetLURejectCallback, 494
- unpack_nas_SetNetworkPreference, 495
- unpack_nas_SetRFInfoCallback, 495
- unpack_nas_SetRoamingIndicatorCallback_ind, 495
- unpack_nas_SetServingSystemCallback_ind, 495
- unpack_nas_SlqsGetLTECphyCAInfo, 495
- nas_AddCDMASysInfo, 83
 - geoSysIdx, 84
 - regPrd, 84
- nas_AddSysInfo, 84
 - cellBroadcastCap, 84
 - geoSysIdx, 84
- nas_CDMAECIOThresh, 86
 - CDMAECIOThreshListLen, 87
 - pCDMAECIOThreshList, 87
- nas_CDMAInfo, 87
 - baseId, 88
 - baseLat, 88
 - baseLong, 88
 - nid, 88
 - refpn, 88
 - sid, 88
- nas_CDMARSSIThresh, 88
 - CDMARSSIThreshListLen, 89
 - pCDMARSSIThreshList, 89
- nas_CDMA SysInfo, 89
 - baseId, 93
 - baseLat, 93
 - baseLong, 93
 - bsInfoValid, 93
 - bsPRev, 93
 - bsPRevValid, 93
 - ccsSupported, 93
 - ccsSupportedValid, 93
 - cdmaSysIdValid, 93
 - isSysPriMatch, 93
 - isSysPriMatchValid, 93
 - MCC, 93
 - MNC, 93
 - networkID, 93
 - networkIdValid, 93
 - pRevInUse, 93
 - pRevInUseValid, 93
 - packetZone, 93
 - packetZoneValid, 93
 - sysInfoCDMA, 93
 - systemID, 93
- nas_CDMASysInfoExt, 93
 - imsi_11_12, 94
 - MCC, 94
- nas_CSGID, 97
 - id, 98
 - mcc, 98
 - mnc, 98
 - mncPcsDigits, 98
 - rat, 98
- nas_CallBarringSysInfo, 84
 - csBarStatus, 85
 - psBarStatus, 85
- nas_CommInfo, 95
 - imsRegState, 97
 - modemMode, 97
 - psState, 97
 - systemMode, 97
 - temperature, 97
- nas_GERANInfo, 104
 - arfcn, 106
 - bsic, 106
 - cellID, 106
 - insNmrCellInfo, 106
 - lac, 106
 - nmrInst, 106
 - plmn, 106
 - rxLev, 106
 - timingAdvance, 106
- nas_GSMRSSIThresh, 108
 - GSMRSSIThreshListLen, 109
 - pGSMRSSIThreshList, 109
- nas_GSMSrvStatusInfo, 109
 - isPrefDataPath, 110
 - srvStatus, 110
 - trueSrvStatus, 110
- nas_GSMSysInfo, 110
 - cellId, 113
 - cellIdValid, 113
 - dtmSupp, 113
 - dtmSuppValid, 113
 - egprsSupp, 113
 - egprsSuppValid, 113
 - lac, 113
 - lacValid, 113
 - MCC, 113
 - MNC, 113
 - networkIdValid, 113
 - regRejectInfoValid, 113
 - rejCause, 113
 - rejectSrvDomain, 113
 - sysInfoGSM, 113
- nas_HDRECIOThresh, 113
 - HDRECIOThreshListLen, 114
 - pHDRECIOThreshList, 114

- nas_HDRIOTthresh, 114
 - HDRIOTthreshListLen, 114
 - pHDRIOTthreshList, 114
- nas_HDRRSSIThresh, 114
 - HDRRSSIThreshListLen, 115
 - pHRRSSIThreshList, 115
- nas_HDRSINRThreshold, 115
 - HDRSINRThreshListLen, 116
 - pHRSINRThreshList, 116
- nas_HDRSysInfo, 116
 - hdrActiveProt, 118
 - hdrActiveProtValid, 118
 - hdrPersonality, 118
 - hdrPersonalityValid, 118
 - is856SysId, 118
 - is856SysIdValid, 118
 - isSysPrIMatch, 118
 - isSysPrIMatchValid, 118
 - sysInfoHDR, 118
- nas_LTEInfo, 121
 - band, 123
 - bandwidth, 123
 - emmConnState, 123
 - emmState, 123
 - emmSubState, 124
 - RXChan, 124
 - TXChan, 124
- nas_LTEInfoInterfreq, 124
 - freqsLen, 124
 - InfoInterfreq, 124
 - ueInIdle, 124
- nas_LTEInfoIntrafreq, 124
 - CellParams, 126
 - cellReselPriority, 126
 - cellsLen, 126
 - earfcn, 126
 - globalCellId, 127
 - plmn, 127
 - sIntraSearch, 127
 - sNonIntraSearch, 127
 - servingCellId, 127
 - tac, 127
 - threshServingLow, 127
 - ueInIdle, 127
- nas_LTEInfoNeighboringGSM, 127
 - freqsLen, 127
 - LteGsmCellInfo, 127
 - ueInIdle, 127
- nas_LTEInfoNeighboringWCDMA, 128
 - freqsLen, 128
 - ueInIdle, 128
- nas_LTERSRRPThresh, 129
 - LTERSRRPThreshListLen, 129
 - pLTERSRRPThreshList, 129
- nas_LTERSRRQThresh, 129
 - LTERSRRQThreshListLen, 130
 - pLTERSRRQThreshList, 130
- nas_LTERSSIThresh, 130
 - LTERSSIThreshListLen, 130
 - pLTERSSIThreshList, 130
- nas_LTESNRThreshold, 132
 - LTESNRThreshListLen, 133
 - pLTESNRThreshList, 133
- nas_LTESigRptConfig, 130
 - avgPeriod, 132
 - rptRate, 132
- nas_LTESysInfo, 133
 - cellId, 135
 - cellIdValid, 135
 - lac, 135
 - lacValid, 135
 - MCC, 135
 - MNC, 135
 - networkIdValid, 135
 - regRejectInfoValid, 135
 - rejCause, 135
 - rejectSrvDomain, 135
 - sysInfoLTE, 135
 - tac, 135
 - tacValid, 136
- nas_MNRInfo, 137
 - mcc, 138
 - mnc, 138
 - rat, 138
- nas_QmiNas3GppNetworkInfo, 141
 - Description, 142
 - Forbidden, 142
 - InUse, 142
 - MCC, 142
 - MNC, 142
 - Preferred, 142
 - Roaming, 142
- nas_QmiNas3GppNetworkRAT, 142
 - MCC, 143
 - MNC, 143
 - RAT, 143
- nas_QmisNasPcsDigit, 143
 - includes_pcs_digit, 144
 - MCC, 144
 - MNC, 144
- nas_RFInfoTlv, 144
 - activeBandClass, 145
 - activeChannel, 145
 - radioInterface, 145
 - radioInterfaceSize, 145
 - TlvPresent, 145
- nas_RejectReasonTlv, 144
 - rejectCause, 144
 - serviceDomain, 144
 - TlvPresent, 144
- nas_SLQSSignalStrengthsIndReq, 150
 - ecioDelta, 150
 - ecioThresholdList, 150
 - ecioThresholdListLen, 150
 - ioDelta, 150
 - lteRsrpDelta, 150

- lteSnrDelta, 151
- rsrqDelta, 151
- rxSignalStrengthDelta, 151
- sinrDelta, 151
- sinrThresholdList, 151
- sinrThresholdListLen, 151
- nas_SLQSSignalStrengthsInformation, 151
 - ecioInfo, 151
 - errorRateInfo, 151
 - io, 151
 - lteRsrpinfo, 151
 - lteSnrinfo, 151
 - rsrqInfo, 151
 - rxSignalStrengthInfo, 151
 - sinr, 152
- nas_SLQSSignalStrengthsTlv, 152
 - sSLQSSignalStrengthsInfo, 152
 - TlvPresent, 152
- nas_SignalStrengthTlv, 149
 - radiolInterface, 150
 - signalStrength, 150
 - TlvPresent, 150
- nas_SrvStatusInfo, 152
 - isPrefDataPath, 153
 - srvStatus, 153
- nas_TDSCDMAECIOThresh, 156
- nas_TDSCDMARSCPTThresh, 157
- nas_TDSCDMARSSIThresh, 157
- nas_TDSCDMASINRThresh, 158
- nas_UMTSInfo, 158
 - cellID, 160
 - ecio, 160
 - geranInst, 160
 - GeranInstInfo, 160
 - lac, 160
 - plmn, 160
 - psc, 160
 - rscp, 160
 - UMTSInstInfo, 160
 - uarfcn, 160
 - umtsInst, 160
- nas_UMTSinstInfo, 160
 - umtsEcio, 162
 - umtsPsc, 162
 - umtsRscp, 162
 - umtsUarfcn, 162
- nas_WCDMAECIOThresh, 164
- nas_WCDMAInfoLTENeighborCell, 165
 - umtsLTENbrCellLen, 166
 - wcdmaRRCTest, 166
- nas_WCDMARSSIThresh, 166
- nas_WCDMASysInfo, 166
 - cellId, 170
 - cellIdValid, 170
 - hsCallStatus, 170
 - hsCallStatusValid, 170
 - hsInd, 170
 - hsIndValid, 170
 - lac, 170
 - lacValid, 170
 - MCC, 170
 - MNC, 170
 - networkIdValid, 170
 - psc, 170
 - pscValid, 170
 - regRejectInfoValid, 170
 - rejCause, 170
 - rejectSrvDomain, 170
 - sysInfoWCDMA, 170
- nas_acqOrderPref, 82
 - acqOrdeLen, 83
 - pAcqOrder, 83
- nas_callBarStatus, 85
 - csBarStatus, 86
 - psBarStatus, 86
- nas_cellParams, 94
 - pci, 95
 - rsrp, 95
 - rsrq, 95
 - rsi, 95
 - srxlev, 95
- nas_currentPLMN, 98
 - MCC, 99
 - MNC, 99
 - netDescr, 99
 - netDescrLength, 99
- nas_dataSrvCapabilities, 99
 - dataCapabilities, 100
 - dataCapabilitiesLen, 100
- nas_detailSvcInfo, 100
 - hdrHybrid, 102
 - hdrSrvStatus, 102
 - isSysForbidden, 102
 - srvCapability, 102
 - srvStatus, 102
- nas_ecioListElement, 102
 - ecio, 103
 - radiolf, 103
- nas_errorRateListElement, 103
 - errorRate, 104
 - radiolf, 104
- nas_geranInstInfo, 106
 - geranArfcn, 107
 - geranBsicBcc, 107
 - geranBsicNcc, 107
 - geranRssi, 107
- nas_gsmCellInfo, 107
 - arfcn, 108
 - band1900, 108
 - bsicld, 108
 - cellIdValid, 108
 - rssi, 108
 - srxlev, 108
- nas_infoInterFreq, 119
 - cell_resel_priority, 119
 - cellInterFreqParams, 119

- cells_len, [120](#)
- earfcn, [120](#)
- threshXHigh, [120](#)
- threshXLow, [120](#)
- nas_lteGsmCellInfo, [120](#)
 - cellReselPriority, [121](#)
 - cells_len, [121](#)
 - GsmCellInfo, [121](#)
 - nccPermitted, [121](#)
 - threshGsmHigh, [121](#)
 - threshGsmLow, [121](#)
- nas_lteRsrpInformation, [128](#)
 - rsrpLevel, [129](#)
- nas_lteSnrInformation, [132](#)
 - snrLevel, [132](#)
- nas_lteWcdmaCellInfo, [136](#)
 - cellReselPriority, [137](#)
 - cellsLen, [137](#)
 - threshXhigh, [137](#)
 - threshXlow, [137](#)
 - uarfcn, [137](#)
 - WCDMACellInfo, [137](#)
- nas_major
 - NASLteNasReleaseInfoTlv, [172](#)
- nas_minor
 - NASLteNasReleaseInfoTlv, [172](#)
- nas_netSelectionPref, [138](#)
 - mcc, [138](#)
 - mnc, [138](#)
 - netReg, [138](#)
- nas_nmrCellInfo, [138](#)
 - nmrArfcn, [140](#)
 - nmrBsic, [140](#)
 - nmrCellID, [140](#)
 - nmrLac, [140](#)
 - nmrPlmn, [141](#)
 - nmrRxLev, [141](#)
- nas_qaQmi3Gpp2TimeZone, [141](#)
 - daylightSavings, [141](#)
 - leapSeconds, [141](#)
 - localTimeOffset, [141](#)
- nas_release
 - NASLteNasReleaseInfoTlv, [172](#)
- nas_roamIndList, [145](#)
 - numInstances, [146](#)
 - radioInterface, [146](#)
 - roamIndicator, [146](#)
- nas_rsrqInformation, [146](#)
 - radioIf, [146](#)
 - rsrq, [146](#)
- nas_rxSignalStrengthListElement, [146](#)
 - radioIf, [147](#)
 - rxSignalStrength, [147](#)
- nas_servSystem, [147](#)
 - csAttachState, [149](#)
 - numRadioInterfaces, [149](#)
 - psAttachState, [149](#)
 - radioInterface, [149](#)
 - regState, [149](#)
 - selNetwork, [149](#)
- nas_sysInfoCommon, [153](#)
 - isSysForbidden, [156](#)
 - isSysForbiddenValid, [156](#)
 - roamStatus, [156](#)
 - roamStatusValid, [156](#)
 - srvCapability, [156](#)
 - srvCapabilityValid, [156](#)
 - srvDomain, [156](#)
 - srvDomainValid, [156](#)
- nas_umtsLTENbrCell, [162](#)
 - cellsTDD, [163](#)
 - earfcn, [163](#)
 - pci, [163](#)
 - rsrp, [163](#)
 - rsrq, [163](#)
 - srxlev, [163](#)
- nas_wcdmaCellInfo, [163](#)
 - cpich_ecno, [164](#)
 - cpich_rscp, [164](#)
 - psc, [164](#)
 - srxlev, [164](#)
- NasGetLTECphyCaInfo, [171](#)
 - PhyCaAggPcellInfo, [171](#)
 - PhyCaAggScellIDBw, [171](#)
 - PhyCaAggScellIndType, [171](#)
 - PhyCaAggScellIndex, [171](#)
 - PhyCaAggScellInfo, [171](#)
- nasRelInfoTlv
 - NASQmiCbkNasSwiOTAMessageInd, [178](#)
- nccPermitted
 - nas_lteGsmCellInfo, [121](#)
- netDescr
 - nas_currentPLMN, [99](#)
- netDescrLength
 - nas_currentPLMN, [99](#)
- netInfoLen
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- netReg
 - nas_netSelectionPref, [138](#)
- NetSelPref
 - NASNetSelPreferenceTlv, [173](#)
- NetworkID
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- networkID
 - nas_CDMA SysInfo, [93](#)
- networkIdValid
 - nas_CDMA SysInfo, [93](#)
 - nas_GSM SysInfo, [113](#)
 - nas_LTE SysInfo, [135](#)
 - nas_WCDMA SysInfo, [170](#)
- networkInfoLen
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [401](#)
- NetworkType
 - currNetworkInfo, [27](#)

- wds_currNetworkInfo, 413
- NewMMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 374
- newMTMessageTlv, 181
 - MTMessageInfo, 181
 - TlvPresent, 181
- newPINLen
 - uim_unblockUIMPIN, 280
- newPINVal
 - uim_unblockUIMPIN, 280
- nextHeader
 - LibPackTFTIDParams, 66
- nid
 - nas_CDMAInfo, 88
 - unpack_nas_GetHomeNetwork_t, 315
- nmrArfcn
 - nas_nmrCellInfo, 140
- nmrBsic
 - nas_nmrCellInfo, 140
- nmrCellID
 - nas_nmrCellInfo, 140
- nmrInst
 - nas_GERANInfo, 106
- nmrLac
 - nas_nmrCellInfo, 140
- nmrPlmn
 - nas_nmrCellInfo, 141
- nmrRxLev
 - nas_nmrCellInfo, 141
- notification
 - unpack_omaDmNotificationsTlv_t, 348
- notificationType
 - sMSEtwsMessage, 258
- num_instances
 - DMScustSettingList, 30
- numApp
 - slotInf, 256
 - uim_slotInfo, 278
- numEntries
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 301
- NumFlows
 - unpack_qos_SLQSSetQosEventCallback_ind_t, 358
- numInstances
 - nas_roamIndList, 146
 - wds_DomainNameList, 414
 - wds_PCSCFFQDNAddressList, 418
 - wds_PCSCFIPv4ServerAddressList, 419
- numOpt
 - wdsDhcpv4OptionList, 424
- numQosFlow
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- numRadiolInterfaces
 - nas_servSystem, 149
- NumRxFilters
 - unpack_qos_QosFlowInfo_t, 352
- numSlot
 - uim_cardStatus, 269
- NumSupUSBComps
 - unpack_dms_GetUSBComp_t, 291
- NumTxFilters
 - unpack_qos_QosFlowInfo_t, 352
- NxtHdrProto
 - unpack_qos_swiQosFilter_t, 365
- OMADMEEnabled
 - unpack_swioma_SLQSOMADMGetSettings_t, 384
- OfflineReason
 - unpack_dms_GetPower_t, 290
- offset
 - uim_readTransparentInfo, 273
- oldPINLen
 - uim_changeUIMPIN, 270
- oldPINVal
 - uim_changeUIMPIN, 270
- operatingMode
 - dms_OperatingModeTlv, 29
- OperatingModeTlv
 - unpack_dms_SetEventReport_ind_t, 293
- OperationMode
 - unpack_dms_GetPower_t, 290
- optCode
 - wdsDhcpv4Option, 424
- optVal
 - wdsDhcpv4Option, 424
- optValLen
 - wdsDhcpv4Option, 424
- otaMsgTlv
 - NASQmiCbkNasSwiOTAMessageInd, 178
- p3GppNetworkInfoInstances
 - unpack_nas_PerformNetworkScan_t, 320
- p3GppNetworkInstanceSize
 - unpack_nas_PerformNetworkScan_t, 320
- pAAASPI
 - pack_wds_SetMobileIPProfile_t, 239
- PACK_WDS_IPV4
 - wds.h, 592
- PACK_WDS_IPV6
 - wds.h, 592
- pAPNClass
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 56
- pAPNClass3GPP2
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 62
- pAPNDisabledFlag
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 56
- pAPNEnabled3GPP2
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 62
- pAPNName
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 56
- pAPNnameSize
 - LibpackProfile3GPP, 44

- LibPackprofile_3GPP, [56](#)
- pAcqOrder
 - nas_acqOrderPref, [83](#)
- pAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, [212](#)
- pAddCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- pAddGSMSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- pAddHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- pAddLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- pAddWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- pAddrAllocPref
 - LibpackProfile3GPP, [43](#)
 - LibPackprofile_3GPP, [56](#)
- pAddress
 - pack_wds_SetMobileIPProfile_t, [239](#)
- pAllowLinger
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAltitudeAssumed
 - unpack_loc_PositionRpt_Ind_t, [311](#)
- pAltitudeWrtEllipsoid
 - unpack_loc_PositionRpt_Ind_t, [311](#)
- pAltitudeWrtMeanSeaLevel
 - unpack_loc_PositionRpt_Ind_t, [311](#)
- pApnString
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pApnStringSize
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pApnname
 - pack_wds_SetDefaultProfile_t, [238](#)
- pAppName
 - loc_LocApplicationInfo, [77](#)
- pAppPriority
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAppProvider
 - loc_LocApplicationInfo, [77](#)
- pAppType
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAppVersion
 - loc_LocApplicationInfo, [77](#)
- pApplicationInfo
 - pack_loc_Start_t, [193](#)
- pAuth
 - pack_wds_SLQSStartDataSession_t, [248](#)
- pAuthPassword
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAuthPassword_tSize
 - LibPackprofile_3GPP2, [62](#)
- pAuthPasswordSize
 - LibpackProfile3GPP2, [50](#)
- pAuthProtocol
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAuthRetryCount
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAuthTimeout
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [62](#)
- pAuthenticationPref
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [56](#)
- pAutosdm
 - pack_swima_SLQSOMADMSetSettings_t, [226](#)
- pBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, [212](#)
 - unpack_nas_SLQSGetSysSelectionPref_t, [335](#)
- pBdsSVInfo
 - pack_loc_Delete_Assist_Data_t, [188](#)
- pCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [202](#)
- pCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [202](#)
- pCDMAECIOThreshList
 - nas_CDMAECIOThresh, [87](#)
- pCDMAInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [336](#)
- pCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [202](#)
- pCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [202](#)
- pCDMARSSIThreshList
 - nas_CDMARSSIThresh, [89](#)
- pCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_t, [338](#)
- pCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- pCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [343](#)
- PCSCFAddrPCO
 - unpack_wds_SLQSGetRuntimeSettings_t, [404](#)
- PCSCFFQDNAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- pCSGID
 - pack_nas_SLQSSetSysSelectionPref_t, [213](#)
- pCardResult
 - unpack_uim_ReadTransparent_t, [387](#)
- pCardStatus
 - unpack_uim_GetCardStatus_t, [386](#)

- unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 390
- pCellDb
 - pack_loc_Delete_Assist_Data_t, 188
- pChangeDuration
 - pack_nas_SLQSInitiateNetworkRegistration_t, 198
- pChgDuration
 - pack_nas_SLQSSetSysSelectionPref_t, 212
- pClkInfo
 - pack_loc_Delete_Assist_Data_t, 188
- pConfigAltitudeAssumed
 - pack_loc_Start_t, 193
- pCreateProfileOut
 - unpack_wds_SLQSCreateProfile_t, 399
- pCurProfile
 - pack_wds_SLQSCreateProfile_t, 240
- pCurImgInfo
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 301
- pCustSettingInfo
 - DMSgetCustomFeatureV2, 31
- pCustSettingList
 - DMSgetCustomFeatureV2, 31
- pDDTMInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 206
- PDOP
 - loc_precisionDilution, 78
- PDType
 - unpack_wds_SLQSGetRuntimeSettings_t, 405
- pDataMode
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pDataRate
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pDeferTime
 - pack_swima_SLQSOMADMSSendSelection_t, 225
- pDestSMSContent
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, 185
 - packgetDyingGaspCfg, 250
- pDestSMSNum
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, 185
 - packgetDyingGaspCfg, 250
- pDualStandByPrefInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 206
- pEmerMode
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSGetSysSelectionPref_t, 335
- pEnabled
 - pack_wds_SetMobileIPProfile_t, 239
- pEncryptData
 - pack_uim_ReadTransparent_t, 228
- pEncryptedData
 - unpack_uim_ReadTransparent_t, 387
- pEncryptedPIN1
 - pack_uim_VerifyPin_t, 234
 - unpack_uim_ChangePin_t, 385
 - unpack_uim_SetPinProtection_t, 388
- unpack_uim_UnblockPin_t, 390
- unpack_uim_VerifyPin_t, 392
- pErrorRateInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 206
- pExtErrCode
 - UnPackGetProfileSettingOut, 411
- pExtErrorCode
 - unpack_wds_SLQSMModifyProfile_t, 405
- pFailureReason
 - unpack_wds_SLQSStartDataSession_t, 409
- pFixId
 - unpack_loc_PositionRpt_Ind_t, 311
- pFwAutoCheck
 - pack_swima_SLQSOMADMSetSettings_t, 226
- pGERANInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 336
- pGPRSMInimumQoS
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pGPRSRequestedQos
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pGSMCallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 343
- pGSMCipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 343
- pGSMRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 202
- pGSMRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 202
- pGSMRSSIThreshList
 - nas_GSMRSSIThresh, 109
- pGSMSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_t, 338
- pGSMSrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 343
- pGSMSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pGWAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSGetSysSelectionPref_t, 335
- pGetCustomInput
 - DMSgetCustomFeatureV2, 31
- pGetDyingGaspCfg
 - unpack_dms_SLQSSwiGetDyingGaspCfg_t, 300
- pGetDyingGaspStatistics
 - unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 300
- pGnssData
 - pack_loc_Delete_Assist_Data_t, 188
- pGpsTime
 - unpack_loc_PositionRpt_Ind_t, 311
- pHASPI
 - pack_wds_SetMobileIPProfile_t, 239

- pHDRECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 202
- pHDRECIOTresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 202
- pHDRECIOTreshList
 - nas_HDRECIOTresh, 114
- pHDRIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 202
- pHDRIOTresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pHDRIOTreshList
 - nas_HDRIOTresh, 114
- pHDRNewUATIAssInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 206
- pHRRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pHRRSSITresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pHRRSSITreshList
 - nas_HDRRSSITresh, 115
- pHRSINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pHRSINRTresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pHRSINRTreshList
 - nas_HRSINRThreshold, 116
- pHDRSessionCloseInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 206
- pHDRSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_t, 338
- pHDRSrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pHeading
 - unpack_loc_PositionRpt_Ind_t, 311
- pHeadingUnc
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorConfidence
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorReliability
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorUncCircular
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorUncEllipseOrientAzimuth
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorUncEllipseSemiMajor
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorUncEllipseSemiMinor
 - unpack_loc_PositionRpt_Ind_t, 311
- pHorizontalAccuracyLvl
 - pack_loc_Start_t, 193
- pHotSwapStatus
 - unpack_uim_GetCardStatus_t, 386
- pHwConfig
 - unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 409
- pIPv4AddrPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pIPv6AddrPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pImCnFlag
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pImageList
 - pack_fms_SetImagesPreference_t, 187
 - unpack_fms_GetImagesPreference_t, 303
- pIndicationToken
 - pack_uim_ChangePin_t, 227
 - pack_uim_ReadTransparent_t, 228
 - pack_uim_SetPinProtection_t, 229
 - pack_uim_UnblockPin_t, 232
 - pack_uim_VerifyPin_t, 234
 - unpack_uim_ChangePin_t, 385
 - unpack_uim_ReadTransparent_t, 387
 - unpack_uim_SetPinProtection_t, 388
 - unpack_uim_UnblockPin_t, 390
 - unpack_uim_VerifyPin_t, 392
- pIntermediateReportState
 - pack_loc_Start_t, 193
- plpcpAckTimeout
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- plpcpCreqRetryCount
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- plsPcscfAddressNedded
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pKeyReferenceID
 - pack_uim_ChangePin_t, 227
 - pack_uim_SetPinProtection_t, 230
 - pack_uim_UnblockPin_t, 232
 - pack_uim_VerifyPin_t, 234
- pLTEBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSGetSysSelectionPref_t, 335
- pLTECphyCa
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pLTEInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, 339
- pLTEInfoInterfreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 336
- pLTEInfoIntrafreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 337
- pLTEInfoNeighboringGSM
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 337
- pLTEInfoNeighboringWCDMA
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 337
- pLTERSRPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203

- pLTERSRPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTERSRPThreshList
 - nas_LTERSRPThresh, 129
- pLTERSRQDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTERSRQThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTERSRQThreshList
 - nas_LTERSRQThresh, 130
- pLTERSSIDDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTERSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTERSSIThreshList
 - nas_LTERSSIThresh, 130
- pLTESNRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTESNRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTESNRThreshList
 - nas_LTESNRThreshold, 133
- pLTESigInfo
 - unpack_nas_SLQSNasSigInfoCallback_t, 338
- pLTESigRptConfig
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pLTESrvStatusInfo
 - unpack_nas_SLQSSetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pLTESysInfo
 - unpack_nas_SLQSSetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pLTEVoiceSupportSysInfo
 - unpack_nas_SLQSSetSysInfo_t, 331
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pLatitude
 - unpack_loc_PositionRpt_Ind_t, 311
- pLcpAckTimeout
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pLcpCreqRetryCount
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pLeapSeconds
 - unpack_loc_PositionRpt_Ind_t, 311
- pLinktimer
 - pack_sms_SendSMS_t, 216
- pLongitude
 - unpack_loc_PositionRpt_Ind_t, 311
- pMNAAA
 - pack_wds_SetMobileIPProfile_t, 239
- pMNCIncPCSDigStat
 - pack_nas_SLQSSetSysSelectionPref_t, 213
- pMNHA
 - pack_wds_SetMobileIPProfile_t, 240
- pMNRInfo
 - pack_nas_SLQSNInitiateNetworkRegistration_t, 198
- pMagneticDeviation
 - unpack_loc_PositionRpt_Ind_t, 311
- pManagedRoamingInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pMessage
 - pack_sms_SendSMS_t, 216
- pMessageIndex
 - pack_sms_SLQSDDeleteSMS_t, 218
- pMessageMode
 - pack_sms_SLQSDDeleteSMS_t, 218
 - pack_sms_SLQSGetSMS_t, 219
 - pack_sms_SLQSGetSMSList_t, 220
 - pack_sms_SLQSMModifySMSStatus_t, 221
- pMessageTag
 - pack_sms_SLQSDDeleteSMS_t, 218
- pMinIntervalTime
 - pack_loc_Start_t, 193
- pMncPcsDigitStatus
 - pack_nas_SLQSNInitiateNetworkRegistration_t, 198
- pMncPcsStatus
 - pack_nas_SLQSGetPLMNName_t, 197
- pModePref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSGetSysSelectionPref_t, 335
- pNAI
 - pack_wds_SetMobileIPProfile_t, 240
- pName
 - pack_wds_SetDefaultProfile_t, 238
- pNegoDnsSrvrPref
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pNetSelPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSGetSysSelectionPref_t, 335
- pNetworkTimeInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pNumberOfPhySlot
 - unpack_uim_SLQSUIMGetSlotsStatus_t, 389
- pOptList
 - wdsDhcpv4OptionList, 424
- pPcSInstance
 - unpack_nas_PerformNetworkScan_t, 320
- pPcSInstanceSize
 - unpack_nas_PerformNetworkScan_t, 321
- pPDNInactivTimeout
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pPDNInactivTimeout3GPP2
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 63
- pPDPTtype
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 57
- pPRLPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSGetSysSelectionPref_t, 335
- pPass
 - pack_wds_SLQSSStartDataSession_t, 248
- pPassword

- LibpackProfile3GPP, [44](#)
- LibPackprofile_3GPP, [56](#)
- pack_wds_SetDefaultProfile_t, [238](#)
- pPasswordSize
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [56](#)
- pPcscfAddrUsingDhcp
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [56](#)
- pPcscfAddrUsingPCO
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [56](#)
- pPdnType
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [63](#)
- pPdpAccessConFlag
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [56](#)
- pPdpContext
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [56](#)
- pPdpDataCompType
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pPdpHdrCompType
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pPppSessCloseTimer1x
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [63](#)
- pPppSessCloseTimerDO
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [63](#)
- pPrecisionDilution
 - unpack_loc_PositionRpt_Ind_t, [311](#)
- pPriDNSIPv4AddPref
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pPriDNSIPv6addpref
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pPriV6DnsAddress
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [63](#)
- pPrimaryHA
 - pack_wds_SetMobileIPProfile_t, [240](#)
- pPrimaryID
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pPrimaryV4DnsAddress
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [63](#)
- pProfileID
 - unpack_wds_SLQSCreateProfile_t, [399](#)
- pProfileId
 - pack_wds_SLQSCreateProfile_t, [240](#)
 - pack_wds_SLQSModifyProfile_t, [244](#)
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, [247](#)
- pProfileSettings
 - unpack_wds_SLQSGetProfileSettings_t, [403](#)
- pProfileType
 - pack_wds_SLQSCreateProfile_t, [240](#)
 - pack_wds_SLQSModifyProfile_t, [244](#)
- pProfilename
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pProfilenameSize
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pQosClassID
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [57](#)
- pRAT
 - pack_nas_SLQSSetSysSelectionPref_t, [213](#)
- pRATInstance
 - unpack_nas_PerformNetworkScan_t, [321](#)
- pRATInstanceSize
 - unpack_nas_PerformNetworkScan_t, [321](#)
- pRATType
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [63](#)
- PRIStrng
 - unpack_dms_GetFirmwareRevisions_t, [288](#)
- PRLInd
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- PRLPTlv
 - NASQmiCbkNasSystemSelPrefInd, [178](#)
- PRLPref
 - NASPRLPreferenceTlv, [177](#)
- pRankIndicatorInd
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [208](#)
- pReadResult
 - unpack_uim_ReadTransparent_t, [387](#)
- pRecurrenceType
 - pack_loc_Start_t, [193](#)
- pRejectReason
 - pack_swioa_SLQSOMADMSendSelection_t, [225](#)
- pRemainingRetries
 - unpack_uim_ChangePin_t, [385](#)
 - unpack_uim_SetPinProtection_t, [388](#)
 - unpack_uim_UnblockPin_t, [390](#)
 - unpack_uim_VerifyPin_t, [392](#)
- pReportChannelRate
 - pack_wds_SLQSGetDUNCallInfo_t, [242](#)
- pReportConnStatus
 - pack_wds_SLQSGetDUNCallInfo_t, [242](#)
- pReportDataBearerTech
 - pack_wds_SLQSGetDUNCallInfo_t, [242](#)
- pReportDormStatus
 - pack_wds_SLQSGetDUNCallInfo_t, [242](#)
- pReqSettings
 - pack_wds_SLQSGetRuntimeSettings_t, [244](#)
- pRequestOptionList

- unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 409
- pRequestedTag
 - pack_sms_SLQSSGetSMSList_t, 220
- pRevInUse
 - nas_CDMA SysInfo, 93
- pRevInUseValid
 - nas_CDMA SysInfo, 93
- pRevTunneling
 - pack_wds_SetMobileIPProfile_t, 240
- pRoamPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSSetSysSelectionPref_t, 335
- pRscp
 - unpack_nas_SLQSNasSigInfoCallback_t, 338
- PSDomain
 - unpack_nas_GetServingNetwork_t, 319
- pSMSAttemptedFlag
 - packgetDyingGaspStatistics, 250
- pSV
 - loc_BdsSVInfo, 71
 - loc_SVInfo, 81
- pSVInfo
 - pack_loc_Delete_Assist_Data_t, 188
- pScanResult
 - unpack_nas_PerformNetworkScan_t, 321
- pSecDNSIPv4AddPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 57
- pSecDNSIPv6addpref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 57
- pSecV6DnsAddress
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 63
- pSecondaryFlag
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 57
- pSecondaryHA
 - pack_wds_SetMobileIPProfile_t, 240
- pSecondaryV4DnsAddress
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 63
- pSensorDataUsage
 - unpack_loc_PositionRpt_Ind_t, 312
- pServingSystemInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pSigIndReq
 - pack_nas_SLQSSetSignalStrengthsCallback_t, 209
- pSignalStrengthInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pSourceIP
 - LibPackTFTIDParams, 66
- pSpeedHorizontal
 - unpack_loc_PositionRpt_Ind_t, 312
- pSpeedUnc
 - unpack_loc_PositionRpt_Ind_t, 312
- pSpeedVertical
 - unpack_loc_PositionRpt_Ind_t, 312
- pSrvDomainPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
 - unpack_nas_SLQSSetSysSelectionPref_t, 335
- pSrvRegRestriction
 - pack_nas_SLQSSetSysSelectionPref_t, 213
- pSubscriptionInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pSvUsedforFix
 - unpack_loc_PositionRpt_Ind_t, 312
- pSysInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pSysInfoNoChange
 - unpack_nas_SLQSSysInfoCallback_t, 344
- pSystemSelectionInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 207
- pTDSCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMAECIOThreshList
 - nas_TDSCDMAECIOThresh, 157
- pTDSCDMARSCPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMARSCPTthresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMARSCPTthreshList
 - nas_TDSCDMARSCPTthresh, 157
- pTDSCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMARSSIThreshList
 - nas_TDSCDMARSSIThresh, 158
- pTDSCDMASINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMASINRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 203
- pTDSCDMASINRThreshList
 - nas_TDSCDMASINRThresh, 158
- pTDSCDMASigInfoExt
 - unpack_nas_SLQSNasSigInfoCallback_t, 338
- pTFTID1Params
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 57
- pTFTID2Params
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 57
- pTdsdmaBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 213
- pTech
 - pack_wds_SLQSSetDataSession_t, 248
- pTechnologyMask
 - unpack_loc_PositionRpt_Ind_t, 312
- pTimeSrc
 - unpack_loc_PositionRpt_Ind_t, 312
- pTimeStamp

- packgetDyingGaspStatistics, [250](#)
- pTimeUnc
 - unpack_loc_PositionRpt_Ind_t, [312](#)
- pTimestampUtc
 - unpack_loc_PositionRpt_Ind_t, [312](#)
- pTransferStatInd
 - pack_wds_SLQSGetDUNCallInfo_t, [242](#)
- pUMTSCellID
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [337](#)
- pUMTSInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [337](#)
- pUMTSMinQoS
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUMTSMinQoSSigInd
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUMTSReqQoS
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUMTSReqQoSSigInd
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUimSlotsStatus
 - unpack_uim_SLQSUIMGetSlotsStatus_t, [389](#)
- pUser
 - pack_wds_SLQSStartDataSession_t, [248](#)
- pUserId
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [63](#)
- pUserIdSize
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [63](#)
- pUsername
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
 - pack_wds_SetDefaultProfile_t, [238](#)
- pUsernameSize
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pVerboseFailReasonType
 - unpack_wds_SLQSStartDataSession_t, [409](#)
- pVerboseFailureReason
 - unpack_wds_SLQSStartDataSession_t, [409](#)
- pVertConfidence
 - unpack_loc_PositionRpt_Ind_t, [312](#)
- pVertReliability
 - unpack_loc_PositionRpt_Ind_t, [312](#)
- pVertUnc
 - unpack_loc_PositionRpt_Ind_t, [312](#)
- pWCDMACallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [344](#)
- pWCDMACipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [344](#)
- pWCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [203](#)
- pWCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [203](#)
- pWCDMAECIOThreshList
 - nas_WCDMAECIOThresh, [164](#)
- pWCDMAInfoLTENeighborCell
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [337](#)
- pWCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [203](#)
- pWCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [203](#)
- pWCDMARSSIThreshList
 - nas_WCDMARSSIThresh, [166](#)
- pWCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_t, [338](#)
- pWCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [344](#)
- pWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [331](#)
 - unpack_nas_SLQSSysInfoCallback_t, [344](#)
- pack_dms_GetActivationState
 - dms.h, [436](#)
- pack_dms_GetBandCapability
 - dms.h, [436](#)
- pack_dms_GetCrashAction
 - dms.h, [436](#)
- pack_dms_GetCustFeature
 - dms.h, [438](#)
- pack_dms_GetCustFeaturesV2
 - dms.h, [438](#)
- pack_dms_GetCustFeaturesV2_t, [181](#)
 - cust_id, [182](#)
 - list_type, [182](#)
 - Tlvresult, [182](#)
- pack_dms_GetDeviceCap
 - dms.h, [438](#)
- pack_dms_GetDeviceCapabilities
 - dms.h, [438](#)
- pack_dms_GetDeviceHardwareRev
 - dms.h, [439](#)
- pack_dms_GetDeviceMfr
 - dms.h, [439](#)
- pack_dms_GetDeviceSerialNumbers
 - dms.h, [439](#)
- pack_dms_GetFSN
 - dms.h, [441](#)
- pack_dms_GetFirmwareInfo
 - dms.h, [440](#)
- pack_dms_GetFirmwareRevision
 - dms.h, [440](#)
- pack_dms_GetFirmwareRevisions
 - dms.h, [440](#)
- pack_dms_GetHardwareRevision
 - dms.h, [441](#)
- pack_dms_GetIMSI
 - dms.h, [441](#)
- pack_dms_GetModelID
 - dms.h, [442](#)

- pack_dms_GetNetworkTime
 - dms.h, [442](#)
- pack_dms_GetPRLVersion
 - dms.h, [443](#)
- pack_dms_GetPower
 - dms.h, [442](#)
- pack_dms_GetSerialNumbers
 - dms.h, [443](#)
- pack_dms_GetUSBComp
 - dms.h, [443](#)
- pack_dms_GetVoiceNumber
 - dms.h, [444](#)
- pack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [446](#)
- pack_dms_SLQSDmsSwiIndicationRegister
 - dms.h, [446](#)
- pack_dms_SLQSDmsSwiIndicationRegister_t, [184](#)
 - resetInfoInd, [184](#)
- pack_dms_SLQSGetBandCapability
 - dms.h, [447](#)
- pack_dms_SLQSSwiClearDyingGaspStatistics
 - dms.h, [447](#)
- pack_dms_SLQSSwiGetDyingGaspCfg
 - dms.h, [447](#)
- pack_dms_SLQSSwiGetDyingGaspStatistics
 - dms.h, [448](#)
- pack_dms_SLQSSwiGetFirmwareCurr
 - dms.h, [448](#)
- pack_dms_SLQSSwiSetDyingGaspCfg
 - dms.h, [448](#)
- pack_dms_SLQSSwiSetDyingGaspCfg_t, [185](#)
 - pDestSMSContent, [185](#)
 - pDestSMSNum, [185](#)
- pack_dms_SetCustFeature
 - dms.h, [444](#)
- pack_dms_SetCustFeature_t, [182](#)
 - DHCPRelayEnabled, [182](#)
 - DisableIMSI, [182](#)
 - GPSPMP, [182](#)
 - GPSSel, [182](#)
 - GpsEnable, [182](#)
 - IPFamSupport, [182](#)
 - IsVoiceEnabled, [182](#)
 - RMAutoConnect, [183](#)
 - SMSSupport, [183](#)
- pack_dms_SetCustFeaturesV2
 - dms.h, [444](#)
- pack_dms_SetCustFeaturesV2_t, [183](#)
 - cust_id, [183](#)
 - cust_value, [183](#)
 - Tlvresult, [183](#)
 - value_length, [183](#)
- pack_dms_SetEventReport
 - dms.h, [445](#)
- pack_dms_SetEventReport_t, [183](#)
 - mode, [183](#)
- pack_dms_SetFirmwarePreference
 - dms.h, [445](#)
- pack_dms_SetPower
 - dms.h, [445](#)
- pack_dms_SetPower_t, [184](#)
 - mode, [184](#)
 - Tlvresult, [184](#)
- pack_dms_SetUSBComp
 - dms.h, [446](#)
- pack_dms_SetUSBComp_t, [184](#)
 - Tlvresult, [184](#)
 - USBComp, [184](#)
- pack_dms_UIMGetICCID
 - dms.h, [449](#)
- pack_dms_UIMGetICCID_t, [185](#)
 - Tlvresult, [185](#)
- pack_fms_GetImagesPreference
 - fms.h, [464](#)
- pack_fms_GetImagesPreference_t, [185](#)
 - Tlvresult, [186](#)
- pack_fms_GetStoredImages
 - fms.h, [464](#)
- pack_fms_GetStoredImages_t, [186](#)
 - Tlvresult, [186](#)
- pack_fms_SetImagesPreference
 - fms.h, [464](#)
- pack_fms_SetImagesPreference_t, [186](#)
 - bForceDownload, [187](#)
 - imageListSize, [187](#)
 - modemindex, [187](#)
 - pImageList, [187](#)
 - Tlvresult, [187](#)
- pack_loc_Delete_Assist_Data_t, [187](#)
 - pBdsSVInfo, [188](#)
 - pCellIdb, [188](#)
 - pClkInfo, [188](#)
 - pGnssData, [188](#)
 - pSVInfo, [188](#)
 - Tlvresult, [188](#)
- pack_loc_DeleteAssistData
 - loc.h, [470](#)
- pack_loc_EventRegister
 - loc.h, [470](#)
- pack_loc_EventRegister_t, [188](#)
 - eventRegister, [190](#)
 - Tlvresult, [190](#)
- pack_loc_SetExtPowerState
 - loc.h, [470](#)
- pack_loc_SetExtPowerState_t, [190](#)
 - extPowerState, [190](#)
 - Tlvresult, [190](#)
- pack_loc_SetOperationMode
 - loc.h, [471](#)
- pack_loc_SetOperationMode_t, [190](#)
 - mode, [191](#)
 - Tlvresult, [191](#)
- pack_loc_Start
 - loc.h, [471](#)
- pack_loc_Start_t, [191](#)
 - pApplicationInfo, [193](#)

- pConfigAltitudeAssumed, [193](#)
 - pHorizontalAccuracyLvl, [193](#)
 - pIntermediateReportState, [193](#)
 - pMinIntervalTime, [193](#)
 - pRecurrenceType, [193](#)
 - SessionId, [193](#)
 - Tlvresult, [193](#)
- pack_loc_Stop
 - loc.h, [471](#)
- pack_loc_Stop_t, [193](#)
 - SessionId, [193](#)
 - Tlvresult, [193](#)
- pack_nas_GetACCOLC
 - nas.h, [480](#)
- pack_nas_GetANAAAAAuthenticationStatus
 - nas.h, [480](#)
- pack_nas_GetCDMANetworkParameters
 - nas.h, [481](#)
- pack_nas_GetHomeNetwork
 - nas.h, [481](#)
- pack_nas_GetNetworkPreference
 - nas.h, [481](#)
- pack_nas_GetRFInfo
 - nas.h, [482](#)
- pack_nas_GetServingNetwork
 - nas.h, [482](#)
- pack_nas_GetServingNetworkCapabilities
 - nas.h, [482](#)
- pack_nas_GetSignalStrengths
 - nas.h, [482](#)
- pack_nas_PerformNetworkScan
 - nas.h, [484](#)
- pack_nas_SLQSGetPLMNName
 - nas.h, [485](#)
- pack_nas_SLQSGetPLMNName_t, [196](#)
 - mcc, [197](#)
 - mnc, [197](#)
 - pMncPcsStatus, [197](#)
- pack_nas_SLQSGetServingSystem
 - nas.h, [486](#)
- pack_nas_SLQSGetSignalStrength
 - nas.h, [486](#)
- pack_nas_SLQSGetSysInfo
 - nas.h, [486](#)
- pack_nas_SLQSGetSysSelectionPref
 - nas.h, [487](#)
- pack_nas_SLQSInitiateNetworkRegistration
 - nas.h, [487](#)
- pack_nas_SLQSInitiateNetworkRegistration_t, [197](#)
 - pChangeDuration, [198](#)
 - pMNRIInfo, [198](#)
 - pMncPcsDigitStatus, [198](#)
 - regAction, [198](#)
- pack_nas_SLQSNasConfigSigInfo2
 - nas.h, [487](#)
- pack_nas_SLQSNasConfigSigInfo2_t, [198](#)
 - pHDRIODelta, [202](#)
 - pHDRIOThresh, [203](#)
 - pLTESigRptConfig, [203](#)
- pack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [488](#)
- pack_nas_SLQSNasGetSigInfo
 - nas.h, [488](#)
- pack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [488](#)
- pack_nas_SLQSNasIndicationRegisterExt_t, [203](#)
 - pDDTMInd, [206](#)
 - pDualStandByPrefInd, [206](#)
 - pErrorRateInd, [206](#)
 - pHDRSessionCloseInd, [206](#)
 - pLTECphyCa, [207](#)
 - pManagedRoamingInd, [207](#)
 - pNetworkTimeInd, [207](#)
 - pServingSystemInd, [207](#)
 - pSignalStrengthInd, [207](#)
 - pSubscriptionInfoInd, [207](#)
 - pSysInfoInd, [207](#)
 - pSystemSelectionInd, [207](#)
- pack_nas_SLQSNasSwiModemStatus
 - nas.h, [489](#)
- pack_nas_SLQSNasSwiOTAMessageCallback
 - nas.h, [489](#)
- pack_nas_SLQSNasSwiOTAMessageCallback_t, [207](#)
 - gsmUmtsDI, [208](#)
 - gsmUmtsUI, [208](#)
 - lteEmmDI, [208](#)
 - lteEmmUI, [208](#)
 - lteEsmDI, [208](#)
 - lteEsmUI, [208](#)
 - pRankIndicatorInd, [208](#)
- pack_nas_SLQSSetBandPreference
 - nas.h, [489](#)
- pack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [490](#)
- pack_nas_SLQSSetSignalStrengthsCallback_t, [208](#)
 - bEnable, [209](#)
 - pSigIndReq, [209](#)
- pack_nas_SLQSSetSysSelectionPref
 - nas.h, [490](#)
- pack_nas_SLQSSetSysSelectionPref_t, [209](#)
 - pAcqOrderPref, [212](#)
 - pBandPref, [212](#)
 - pCSGID, [213](#)
 - pChgDuration, [212](#)
 - pEmerMode, [213](#)
 - pGWAOrderPref, [213](#)
 - pLTEBandPref, [213](#)
 - pModePref, [213](#)
 - pNetSelPref, [213](#)
 - pPRLPref, [213](#)
 - pRAT, [213](#)
 - pRoamPref, [213](#)
 - pSrvDomainPref, [213](#)
 - pSrvRegRestriction, [213](#)
 - pTdsdmaBandPref, [213](#)
- pack_nas_SLQSSwiGetLteCQI

- nas.h, [490](#)
- pack_nas_SetACCOLC
 - nas.h, [484](#)
- pack_nas_SetACCOLC_t, [193](#)
 - accolc, [195](#)
 - spc, [195](#)
- pack_nas_SetLURejectCallback
 - nas.h, [484](#)
- pack_nas_SetNetworkPreference
 - nas.h, [485](#)
- pack_nas_SetNetworkPreference_t, [195](#)
 - Duration, [196](#)
 - TechnologyPref, [196](#)
 - Tlvresult, [196](#)
- pack_nas_SetRFInfoCallback
 - nas.h, [485](#)
- pack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [485](#)
- pack_qmi_t, [213](#)
 - msgid, [213](#)
 - svc, [213](#)
 - timeout, [213](#)
 - xid, [214](#)
- pack_qos_SLQSQosGetNetworkStatus
 - qos.h, [548](#)
- pack_qos_SLQSQosSwiReadApnExtraParams
 - qos.h, [548](#)
- pack_qos_SLQSQosSwiReadApnExtraParams_t, [214](#)
 - apnId, [214](#)
- pack_qos_SLQSQosSwiReadDataStats
 - qos.h, [549](#)
- pack_qos_SLQSQosSwiReadDataStats_t, [214](#)
 - apnId, [214](#)
- pack_qos_SLQSSetQosEventCallback
 - qos.h, [550](#)
- pack_qos_SLQSSetQosEventCallback_t, [214](#)
 - enable, [216](#)
- pack_sms_SLQSDeleteSMS
 - sms.h, [562](#)
- pack_sms_SLQSDeleteSMS_t, [217](#)
 - pMessageIndex, [218](#)
 - pMessageMode, [218](#)
 - pMessageTag, [218](#)
 - storageType, [218](#)
- pack_sms_SLQSGetSMS
 - sms.h, [562](#)
- pack_sms_SLQSGetSMS_t, [218](#)
 - messageIndex, [219](#)
 - pMessageMode, [219](#)
 - storageType, [219](#)
- pack_sms_SLQSGetSMSList
 - sms.h, [563](#)
- pack_sms_SLQSGetSMSList_t, [219](#)
 - pMessageMode, [220](#)
 - pRequestedTag, [220](#)
 - storageType, [220](#)
- pack_sms_SLQSModifySMSStatus
 - sms.h, [563](#)
- pack_sms_SLQSModifySMSStatus_t, [220](#)
 - messageIndex, [221](#)
 - messageTag, [221](#)
 - pMessageMode, [221](#)
 - storageType, [221](#)
- pack_sms_SendSMS
 - sms.h, [560](#)
- pack_sms_SendSMS_t, [216](#)
 - messageFormat, [216](#)
 - messageSize, [216](#)
 - pLinktimer, [216](#)
 - pMessage, [216](#)
- pack_sms_SetNewSMSCallback
 - sms.h, [562](#)
- pack_sms_SetNewSMSCallback_t, [217](#)
 - status, [217](#)
- pack_swiloc_SwiLocGetAutoStart
 - swiloc.h, [568](#)
- pack_swiloc_SwiLocSetAutoStart
 - swiloc.h, [568](#)
- pack_swiloc_SwiLocSetAutoStart_t, [221](#)
 - fix_rate, [223](#)
 - fix_type, [223](#)
 - function, [223](#)
 - max_dist, [223](#)
 - max_time, [223](#)
 - set_fix_rate, [223](#)
 - set_fix_type, [223](#)
 - set_function, [223](#)
 - set_max_dist, [223](#)
 - set_max_time, [223](#)
- pack_swioima_SLQSOMADMAAlertCallback
 - swioima.h, [570](#)
- pack_swioima_SLQSOMADMCancelSession
 - swioima.h, [571](#)
- pack_swioima_SLQSOMADMCancelSession_t, [223](#)
 - sessionType, [223](#)
- pack_swioima_SLQSOMADMGetSessionInfo
 - swioima.h, [571](#)
- pack_swioima_SLQSOMADMGetSessionInfo_t, [223](#)
 - SessionType, [224](#)
- pack_swioima_SLQSOMADMGetSettings
 - swioima.h, [572](#)
- pack_swioima_SLQSOMADMSendSelection
 - swioima.h, [573](#)
- pack_swioima_SLQSOMADMSendSelection_t, [224](#)
 - pDeferTime, [225](#)
 - pRejectReason, [225](#)
 - selection, [225](#)
- pack_swioima_SLQSOMADMSetSettings
 - swioima.h, [573](#)
- pack_swioima_SLQSOMADMSetSettings_t, [225](#)
 - FOTAdownload, [226](#)
 - pAutosdm, [226](#)
 - pFwAutoCheck, [226](#)
- pack_swioima_SLQSOMADMStartSession
 - swioima.h, [574](#)
- pack_swioima_SLQSOMADMStartSession_t, [226](#)

- sessionType, [226](#)
- pack_uim_ChangePin
 - uim.h, [581](#)
- pack_uim_ChangePin_t, [226](#)
 - changePIN, [227](#)
 - EncryptedPIN1, [227](#)
 - pIndicationToken, [227](#)
 - pKeyReferenceID, [227](#)
 - sessionInfo, [227](#)
 - Tlvresult, [227](#)
- pack_uim_GetCardStatus
 - uim.h, [581](#)
- pack_uim_ReadTransparent
 - uim.h, [581](#)
- pack_uim_ReadTransparent_t, [227](#)
 - fileIndex, [228](#)
 - pEncryptData, [228](#)
 - pIndicationToken, [228](#)
 - readTransparent, [228](#)
 - sessionInfo, [228](#)
 - Tlvresult, [229](#)
- pack_uim_SLQSUIMEventRegister
 - uim.h, [582](#)
- pack_uim_SLQSUIMEventRegister_t, [230](#)
 - eventMask, [230](#)
- pack_uim_SLQSUIGetSlotsStatus
 - uim.h, [582](#)
- pack_uim_SLQSUISSwitchSlot
 - uim.h, [582](#)
- pack_uim_SLQSUISSwitchSlot_t, [230](#)
 - bLogicalSlot, [231](#)
 - ulPhysicalSlot, [231](#)
- pack_uim_SetPinProtection
 - uim.h, [581](#)
- pack_uim_SetPinProtection_t, [229](#)
 - EncryptedPIN1, [229](#)
 - pIndicationToken, [229](#)
 - pKeyReferenceID, [230](#)
 - pinProtection, [230](#)
 - sessionInfo, [230](#)
 - Tlvresult, [230](#)
- pack_uim_UnblockPin
 - uim.h, [584](#)
- pack_uim_UnblockPin_t, [231](#)
 - EncryptedPIN1, [232](#)
 - pIndicationToken, [232](#)
 - pKeyReferenceID, [232](#)
 - pinProtection, [232](#)
 - sessionInfo, [232](#)
 - Tlvresult, [232](#)
- pack_uim_VerifyPin
 - uim.h, [584](#)
- pack_uim_VerifyPin_t, [232](#)
 - pEncryptedPIN1, [234](#)
 - pIndicationToken, [234](#)
 - pKeyReferenceID, [234](#)
 - sessionInfo, [234](#)
 - Tlvresult, [234](#)
- verifyPIN, [234](#)
- pack_wds_GetConnectionRate
 - wds.h, [592](#)
- pack_wds_GetDefaultProfile
 - wds.h, [593](#)
- pack_wds_GetDefaultProfile_t, [234](#)
 - profiletype, [235](#)
- pack_wds_GetDefaultProfileNum
 - wds.h, [593](#)
- pack_wds_GetDefaultProfileNum_t, [235](#)
 - family, [235](#)
 - type, [235](#)
- pack_wds_GetDormancyState
 - wds.h, [593](#)
- pack_wds_GetDormancyState_t, [235](#)
- pack_wds_GetLastMobileIPError
 - wds.h, [595](#)
- pack_wds_GetLastMobileIPError_t, [235](#)
- pack_wds_GetMobileIP
 - wds.h, [595](#)
- pack_wds_GetMobileIP_t, [235](#)
- pack_wds_GetMobileIPProfile
 - wds.h, [596](#)
- pack_wds_GetMobileIPProfile_t, [235](#)
 - index, [237](#)
- pack_wds_GetPacketStatus
 - wds.h, [596](#)
- pack_wds_GetPacketStatus_t, [237](#)
 - statmask, [237](#)
- pack_wds_GetSessionDuration
 - wds.h, [596](#)
- pack_wds_GetSessionDuration_t, [237](#)
- pack_wds_GetSessionState
 - wds.h, [597](#)
- pack_wds_RMSetTransferStatistics
 - wds.h, [597](#)
- pack_wds_RMSetTransferStatistics_t, [237](#)
 - RmTrasnferStaticsReq, [237](#)
- pack_wds_SLQSCreateProfile
 - wds.h, [600](#)
- pack_wds_SLQSCreateProfile_t, [240](#)
 - pCurProfile, [240](#)
 - pProfileId, [240](#)
 - pProfileType, [240](#)
- pack_wds_SLQSDeleteProfile
 - wds.h, [600](#)
- pack_wds_SLQSDeleteProfile_t, [241](#)
 - profileIndex, [241](#)
 - profileType, [241](#)
- pack_wds_SLQSGet3GPPConfigItem
 - wds.h, [600](#)
- pack_wds_SLQSGetCurrDataSystemStat
 - wds.h, [601](#)
- pack_wds_SLQSGetCurrDataSystemStat_t, [241](#)
- pack_wds_SLQSGetDUNCallInfo
 - wds.h, [602](#)
- pack_wds_SLQSGetDUNCallInfo_t, [241](#)
 - Mask, [242](#)

- pReportChannelRate, [242](#)
 - pReportConnStatus, [242](#)
 - pReportDataBearerTech, [242](#)
 - pReportDormStatus, [242](#)
 - pTransferStatInd, [242](#)
- pack_wds_SLQSGetDataBearerTechnology
 - wds.h, [601](#)
- pack_wds_SLQSGetDataBearerTechnology_t, [241](#)
- pack_wds_SLQSGetProfileSettings
 - wds.h, [602](#)
- pack_wds_SLQSGetProfileSettings_t, [242](#)
 - ProfileId, [243](#)
 - ProfileType, [243](#)
- pack_wds_SLQSGetRuntimeSettings
 - wds.h, [602](#)
- pack_wds_SLQSGetRuntimeSettings_t, [243](#)
 - pReqSettings, [244](#)
- pack_wds_SLQSModifyProfile
 - wds.h, [603](#)
- pack_wds_SLQSModifyProfile_t, [244](#)
 - curProfile, [244](#)
 - pProfileId, [244](#)
 - pProfileType, [244](#)
- pack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [604](#)
- pack_wds_SLQSSetDHCPv4ClientConfig_t, [247](#)
 - pProfileId, [247](#)
- pack_wds_SLQSSet3GPPConfigItem
 - wds.h, [603](#)
- pack_wds_SLQSSet3GPPConfigItem_t, [244](#)
 - _3gppRelease, [245](#)
 - defaultPDNEnabled, [245](#)
 - profileList, [245](#)
- pack_wds_SLQSSetIPFamilyPreference
 - wds.h, [604](#)
- pack_wds_SLQSSetIPFamilyPreference_t, [245](#)
 - IPFamilyPreference, [246](#)
- pack_wds_SLQSSetWdsEventCallback
 - wds.h, [604](#)
- pack_wds_SLQSSetWdsEventCallback_t, [246](#)
 - currentDataBearer, [246](#)
 - dataBearer, [246](#)
 - dataSystemStatus, [246](#)
 - dormancyStatus, [246](#)
 - interval, [246](#)
 - mobileIP, [247](#)
 - transferStats, [247](#)
- pack_wds_SLQSStartDataSession
 - wds.h, [605](#)
- pack_wds_SLQSStartDataSession_t, [247](#)
 - pAuth, [248](#)
 - pPass, [248](#)
 - pTech, [248](#)
 - pUser, [248](#)
 - pprofileid3gpp, [248](#)
 - pprofileid3gpp2, [248](#)
- pack_wds_SLQSStopDataSession
 - wds.h, [605](#)
- pack_wds_SLQSStopDataSession_t, [248](#)
 - psid, [249](#)
- pack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, [606](#)
- pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [249](#)
 - contextId, [249](#)
 - contextType, [249](#)
- pack_wds_SetDefaultProfile
 - wds.h, [597](#)
- pack_wds_SetDefaultProfile_t, [237](#)
 - authentication, [238](#)
 - ipAddress, [238](#)
 - pApnname, [238](#)
 - pName, [238](#)
 - pPassword, [238](#)
 - pUsername, [238](#)
 - pdpType, [238](#)
 - primaryDNS, [238](#)
 - profileType, [238](#)
 - secondaryDNS, [238](#)
- pack_wds_SetDefaultProfileNum
 - wds.h, [599](#)
- pack_wds_SetDefaultProfileNum_t, [238](#)
 - family, [239](#)
 - index, [239](#)
 - type, [239](#)
- pack_wds_SetMobileIPProfile
 - wds.h, [599](#)
- pack_wds_SetMobileIPProfile_t, [239](#)
 - index, [239](#)
 - pAAASPI, [239](#)
 - pAddress, [239](#)
 - pEnabled, [239](#)
 - pHASPI, [239](#)
 - pMNAAA, [239](#)
 - pMNHA, [240](#)
 - pNAI, [240](#)
 - pPrimaryHA, [240](#)
 - pRevTunneling, [240](#)
 - pSecondaryHA, [240](#)
 - spc, [240](#)
- PackCreateProfileOut, [249](#)
 - ExtErrorCode, [249](#)
 - ProfileIndex, [249](#)
 - ProfileType, [249](#)
- package_name
 - unpack_omaDmFotaTlv_t, [347](#)
- packageid_str
 - unpack_dms_GetFirmwareInfo_t, [286](#)
- packetZone
 - nas_CDMA SysInfo, [93](#)
- packetZoneValid
 - nas_CDMA SysInfo, [93](#)
- packgetDyingGaspCfg, [249](#)
 - pDestSMSContent, [250](#)
 - pDestSMSNum, [250](#)
- packgetDyingGaspStatistics, [250](#)
 - pSMSAttemptedFlag, [250](#)

- pTimeStamp, [250](#)
- path
 - uim_fileInfo, [271](#)
- pathLen
 - uim_fileInfo, [271](#)
- pci
 - nas_cellParams, [95](#)
 - nas_umtsLTENbrCell, [163](#)
 - NASPhyCaAggPcellInfo, [174](#)
 - NASPhyCaAggScellIndType, [176](#)
 - NASPhyCaAggScellInfo, [177](#)
- pcsfQDNAddress
 - wds_PCSCFFQDNAddressList, [418](#)
- pdpType
 - pack_wds_SetDefaultProfile_t, [238](#)
- pdptype
 - unpack_wds_GetDefaultProfile_t, [394](#)
- peakRate
 - unpack_qos_tokenBucket_t, [370](#)
- peakThroughputClass
 - LibPackGPRSRequestedQoS, [38](#)
 - wds_GPRSQoS, [416](#)
- PersistentTechPref
 - unpack_nas_GetNetworkPreference_t, [317](#)
- persoFeature
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- persoRetries
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- persoState
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- persoUnblockRetries
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- PhyCaAggPcellInfo
 - NasGetLTECphyCaInfo, [171](#)
- PhyCaAggScellIDBW
 - NasGetLTECphyCaInfo, [171](#)
- PhyCaAggScellIndType
 - NasGetLTECphyCaInfo, [171](#)
- PhyCaAggScellIndex
 - NasGetLTECphyCaInfo, [171](#)
- PhyCaAggScellInfo
 - NasGetLTECphyCaInfo, [171](#)
- pin1Len
 - uim_encryptedPIN1, [271](#)
- pin1Retries
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- pin1State
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- pin1Val
 - uim_encryptedPIN1, [271](#)
- pin2Retries
 - appStats, [24](#)
- uim_appStatus, [267](#)
- pin2State
 - appStats, [24](#)
 - uim_appStatus, [267](#)
- pinID
 - uim_changeUIMPIN, [270](#)
 - uim_setPINProtection, [276](#)
 - uim_unblockUIMPIN, [280](#)
 - uim_verifyUIMPIN, [281](#)
- pinLen
 - uim_changeUIMPIN, [270](#)
 - uim_verifyUIMPIN, [281](#)
- pinLength
 - uim_setPINProtection, [276](#)
- pinOperation
 - uim_setPINProtection, [276](#)
- pinProtection
 - pack_uim_SetPinProtection_t, [230](#)
 - pack_uim_UnblockPin_t, [232](#)
- pinVal
 - uim_changeUIMPIN, [270](#)
 - uim_verifyUIMPIN, [281](#)
- pinValue
 - uim_setPINProtection, [276](#)
- PkgDescLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- PkgDescription
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- PkgName
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- PkgNameLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- pkgver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [301](#)
- PktErrRate
 - unpack_qos_swiQoSFlow_t, [370](#)
- plmn
 - nas_GERANInfo, [106](#)
 - nas_LTEInfoIntrafreq, [127](#)
 - nas_UMTSInfo, [160](#)
- port
 - unpack_qos_Port_t, [351](#)
- pprofileid3gpp
 - pack_wds_SLQSSStartDataSession_t, [248](#)
- pprofileid3gpp2
 - pack_wds_SLQSSStartDataSession_t, [248](#)
- prDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [411](#)
- prDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [411](#)
- prPCSCFIPv4Address

- unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 411
- prPCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 411
- Precedence
 - unpack_qos_swiQosFilter_t, 365
- precedenceClass
 - LibPackGPRSRequestedQoS, 38
 - wds_GPRSQoS, 416
- prefNetwork
 - unpack_wds_SLQSGetCurrDataSystemStat_t, 401
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- Preferred
 - nas_QmiNas3GppNetworkInfo, 142
- prefixLen
 - unpack_qos_IPv6Addr_t, 349
- priSize
 - unpack_dms_GetFirmwareRevisions_t, 288
- pridns
 - unpack_wds_GetDefaultProfile_t, 394
- pridnsV6
 - unpack_wds_GetDefaultProfile_t, 394
- primaryDNS
 - pack_wds_SetDefaultProfile_t, 238
- PrimaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 405
- PrimaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, 405
- primaryHA
 - unpack_wds_GetMobileIPProfile_t, 396
- priver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 301
- priversion_str
 - unpack_dms_GetFirmwareInfo_t, 287
- ProfileID
 - unpack_wds_SLQSGetRuntimeSettings_t, 405
- ProfileId
 - pack_wds_SLQSGetProfileSettings_t, 243
- profileId
 - wdsDhcpv4ProfileId, 424
- ProfileId3GPP2
 - unpack_qos_swiQosFlow_t, 370
- ProfileIndex
 - PackCreateProfileOut, 249
- profileIndex
 - pack_wds_SLQSDeleteProfile_t, 241
 - wds_ProfileIdentifier, 419
- profileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 245
 - unpack_wds_SLQSGet3GPPConfigItem_t, 400
- ProfileName
 - unpack_wds_SLQSGetRuntimeSettings_t, 405
- ProfileType
 - pack_wds_SLQSGetProfileSettings_t, 243
 - PackCreateProfileOut, 249
- unpack_wds_SLQSGetProfileSettings_t, 403
- profileType
 - pack_wds_SetDefaultProfile_t, 238
 - pack_wds_SLQSDeleteProfile_t, 241
 - wds_ProfileIdentifier, 419
 - wdsDhcpv4ProfileId, 425
- profiletype
 - pack_wds_GetDefaultProfile_t, 235
- Protocol
 - unpack_nas_GetCDMANetworkParameters_t, 314
- psAttachState
 - nas_servSystem, 149
 - NASServingSystemInfo, 180
- psBarStatus
 - nas_CallBarringSysInfo, 85
 - nas_callBarStatus, 86
- psState
 - nas_CommInfo, 97
- psc
 - nas_UMTSInfo, 160
 - nas_wcdmaCellInfo, 164
 - nas_WCDMASysInfo, 170
- pscValid
 - nas_WCDMASysInfo, 170
- pscsfIPv4Addr
 - wds_PCSCFIPv4ServerAddressList, 419
- psid
 - pack_wds_SLQSStopDataSession_t, 249
 - unpack_wds_SLQSStartDataSession_t, 409
- puk1Retries
 - appStats, 24
 - uim_appStatus, 267
- puk2Retries
 - appStats, 24
 - uim_appStatus, 267
- pukLen
 - uim_unblockUIMPIN, 280
- pukVal
 - uim_unblockUIMPIN, 280
- QCI
 - LibPackQosClassID, 64
- QFlowState
 - unpack_qos_QosFlowInfo_t, 352
- QMI pack/unpack (pack), 17
- qaGobiApiTableBandClasses.h, 502
- qaGobiApiTableCallControlReturnReasons.h, 505
- qaGobiApiTableCallEndReasons.h, 506
- qaGobiApiTableCarrierCodes.h, 521
- qaGobiApiTableCodingScheme.h, 523
- qaGobiApiTableGpsCapabilityCodes.h, 526
- qaGobiApiTablePowerModes.h, 526
- qaGobiApiTableRadioInterfaces.h, 527
- qaGobiApiTableRegionCodes.h, 527
- qaGobiApiTableServiceOptions.h, 528
- qaGobiApiTableSupServiceInfoClasses.h, 530
- qaGobiApiTableSwiAudio.h, 531
- qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 531

qaGobiApiTableVoiceCallEndReasons.h, [533](#)

qm_wds_ds_profile_extended_err_codes

qmerrno.h, [545](#)

qmerrno.h

eQCWWAN_ERR_API_MUTEX_TIMEOUT, [542](#)

eQCWWAN_ERR_BUFFER_SZ, [541](#)

eQCWWAN_ERR_CANCEL_OP, [542](#)

eQCWWAN_ERR_DRIVER, [542](#)

eQCWWAN_ERR_ENUM_BEGIN, [541](#)

eQCWWAN_ERR_ENUM_END, [542](#)

eQCWWAN_ERR_FILE_COPY, [541](#)

eQCWWAN_ERR_FILE_OPEN, [541](#)

eQCWWAN_ERR_GENERAL, [541](#)

eQCWWAN_ERR_INTERNAL, [541](#)

eQCWWAN_ERR_INVALID_ARG, [541](#)

eQCWWAN_ERR_INVALID_DEVID, [541](#)

eQCWWAN_ERR_INVALID_FILE, [541](#)

eQCWWAN_ERR_INVALID_QMI_RSP, [541](#)

eQCWWAN_ERR_INVALID_XID, [542](#)

eQCWWAN_ERR_MALFORMED_QMI_RSP, [541](#)

eQCWWAN_ERR_MEMORY, [541](#)

eQCWWAN_ERR_MULTIPLE_DEVICES, [542](#)

eQCWWAN_ERR_NO_CANCELABLE_OP, [542](#)

eQCWWAN_ERR_NO_CONNECTION, [541](#)

eQCWWAN_ERR_NO_DEVICE, [541](#)

eQCWWAN_ERR_NO_SIGNAL, [542](#)

eQCWWAN_ERR_NONE, [541](#)

eQCWWAN_ERR_NULL_TLV, [545](#)

eQCWWAN_ERR_OFFLINE, [542](#)

eQCWWAN_ERR_PDU_GENERATION, [542](#)

eQCWWAN_ERR_QMI_ABORTED, [542](#)

eQCWWAN_ERR_QMI_ACCESS_DENIED, [544](#)

eQCWWAN_ERR_QMI_ACK_NOT_SENT, [544](#)

eQCWWAN_ERR_QMI_ARG_TOO_LONG, [542](#)

eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, [543](#)

eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, [543](#)

eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, [544](#)

eQCWWAN_ERR_QMI_CALL_FAILED, [542](#)

eQCWWAN_ERR_QMI_CARD_BUSY_RSP, [545](#)

eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, [544](#)

eQCWWAN_ERR_QMI_CAT_END, [545](#)

eQCWWAN_ERR_QMI_CAT_START, [545](#)

eQCWWAN_ERR_QMI_CAUSE_CODE, [543](#)

eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, [542](#)

eQCWWAN_ERR_QMI_CONNECT, [541](#)

eQCWWAN_ERR_QMI_DEVICE_IN_USE, [542](#)

eQCWWAN_ERR_QMI_DEVICE_NOT_READY, [543](#)

eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, [543](#)

eQCWWAN_ERR_QMI_DISABLED, [543](#)

eQCWWAN_ERR_QMI_ENCODING, [543](#)

eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, [545](#)

eQCWWAN_ERR_QMI_EVENT_REG_FAILED, [545](#)

eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, [544](#)

eQCWWAN_ERR_QMI_FDN_RESTRICT, [544](#)

eQCWWAN_ERR_QMI_FLOW_SUSPENDED, [543](#)

eQCWWAN_ERR_QMI_GENERAL, [543](#)

eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, [544](#)

eQCWWAN_ERR_QMI_IFACE, [541](#)

eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, [544](#)

eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, [543](#)

eQCWWAN_ERR_QMI_INCORRECT_PIN, [542](#)

eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, [544](#)

eQCWWAN_ERR_QMI_INJECT_TIMEOUT, [544](#)

eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, [543](#)

eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, [543](#)

eQCWWAN_ERR_QMI_INTERNAL, [542](#)

eQCWWAN_ERR_QMI_INVALID_ARG, [543](#)

eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, [542](#)

eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, [543](#)

eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, [545](#)

eQCWWAN_ERR_QMI_INVALID_HANDLE, [542](#)

eQCWWAN_ERR_QMI_INVALID_ID, [543](#)

eQCWWAN_ERR_QMI_INVALID_INDEX, [543](#)

eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, [543](#)

eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, [543](#)

eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, [543](#)

eQCWWAN_ERR_QMI_INVALID_OPERATION, [543](#)

eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, [542](#)

eQCWWAN_ERR_QMI_INVALID_PINID, [542](#)

eQCWWAN_ERR_QMI_INVALID_PROFILE, [542](#)

eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, [542](#)

eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, [543](#)

eQCWWAN_ERR_QMI_INVALID_QMI_CMD, [544](#)

eQCWWAN_ERR_QMI_INVALID_QOS_ID, [543](#)

eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, [542](#)

eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, [542](#)

- eQCWWAN_ERR_QMI_INVALID_TECH_PREF, [542](#)
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, [545](#)
- eQCWWAN_ERR_QMI_INVALID_TRANSITION, [543](#)
- eQCWWAN_ERR_QMI_INVALID_TX_ID, [542](#)
- eQCWWAN_ERR_QMI_MALFORMED_MSG, [542](#)
- eQCWWAN_ERR_QMI_MAX, [544](#)
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, [543](#)
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, [543](#)
- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, [543](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, [543](#)
- eQCWWAN_ERR_QMI_MISSING_ARG, [542](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED, [544](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED, [544](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY, [543](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, [543](#)
- eQCWWAN_ERR_QMI_NO_EFFECT, [542](#)
- eQCWWAN_ERR_QMI_NO_ENTRY, [543](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE, [542](#)
- eQCWWAN_ERR_QMI_NO_MEMORY, [542](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, [542](#)
- eQCWWAN_ERR_QMI_NO_RADIO, [544](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, [544](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS, [542](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, [543](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED, [542](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED, [544](#)
- eQCWWAN_ERR_QMI_OFFSET, [542](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, [542](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, [542](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, [544](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL, [542](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED, [543](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, [543](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH, [544](#)
- eQCWWAN_ERR_QMI_REQ, [541](#)
- eQCWWAN_ERR_QMI_REQ_SCH, [541](#)
- eQCWWAN_ERR_QMI_REQ_TO, [541](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, [543](#)
- eQCWWAN_ERR_QMI_RSP, [541](#)
- eQCWWAN_ERR_QMI_RSP_TO, [541](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER, [544](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, [544](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE, [543](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID, [543](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, [543](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, [544](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, [543](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR, [544](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, [544](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE, [544](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, [542](#)
- eQCWWAN_ERR_QMI_UNKNOWN, [543](#)
- eQCWWAN_ERR_QMI_WIDTH, [545](#)
- eQCWWAN_ERR_RESET, [542](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR, [544](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, [544](#)
- eQCWWAN_ERR_SWICM_END, [545](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, [544](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, [544](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, [544](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, [544](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, [544](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, [544](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, [544](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, [544](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE, [544](#)
- eQCWWAN_ERR_SWICM_START, [544](#)
- eQCWWAN_ERR_SWICM_TIMEOUT, [544](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN, [544](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP, [544](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN, [544](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP, [544](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, [545](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, [545](#)

- eQCWWAN_ERR_SWIDCS_END, [545](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR, [545](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR, [545](#)
- eQCWWAN_ERR_SWIDCS_START, [545](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, [545](#)
- eQCWWAN_ERR_SWIIM_END, [545](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, [545](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, [545](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, [545](#)
- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, [545](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, [545](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, [545](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, [545](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, [545](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH, [545](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR, [545](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE, [545](#)
- eQCWWAN_ERR_SWIIM_START, [545](#)
- eQCWWAN_ERR_SWISM_END, [545](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, [545](#)
- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, [545](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, [545](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, [545](#)
- eQCWWAN_ERR_SWISMS_START, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, [546](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, [546](#)
- eWDS_ERR_PROFILE_REG_END, [546](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL, [546](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END, [546](#)
- qmerrno.h, [539](#)
 - eQCWWANError, [541](#)
 - qm_wds_ds_profile_extended_err_codes, [545](#)
- qmiSmsMessageList, [250](#)
 - messageIndex, [251](#)
 - messageTag, [251](#)
- qmiWDSDataBearerTechnology, [251](#)
 - currentNetwork, [251](#)
 - ratMask, [251](#)
 - soMask, [251](#)
- qos.h, [546](#)
 - pack_qos_SLQSQosGetNetworkStatus, [548](#)
 - pack_qos_SLQSQosSmiReadApnExtraParams, [548](#)
 - pack_qos_SLQSQosSmiReadDataStats, [549](#)
 - pack_qos_SLQSSetQosEventCallback, [550](#)
 - unpack_qos_SLQSQosGetNetworkStatus, [550](#)
 - unpack_qos_SLQSQosSmiReadApnExtraParams, [552](#)
 - unpack_qos_SLQSQosSmiReadDataStats, [552](#)
 - unpack_qos_SLQSSetQosEventCallback, [553](#)
 - unpack_qos_SLQSSetQosEventCallback_ind, [553](#)
 - unpack_qos_SLQSSetQosNWStatusCallback_ind, [554](#)
 - unpack_qos_SLQSSetQosPriEventCallback_ind, [554](#)
 - unpack_qos_SLQSSetQosStatusCallback_ind, [556](#)
- qosDeliveryOrder
 - LibPackUMTSQoS, [69](#)
 - wds_UMTSMInQoS, [422](#)
- qosFlow
 - unpack_qos_SLQSQosSmiReadDataStats_t, [358](#)
- QosFlowInfo
 - unpack_qos_SLQSSetQosEventCallback_ind_t, [358](#)
- RAN
 - unpack_nas_GetServingNetwork_t, [319](#)
- RAT
 - nas_QmiNas3GppNetworkRAT, [143](#)
- RATMask

- currNetworkInfo, [27](#)
- wds_currNetworkInfo, [413](#)
- RFBandInfoElements, [251](#)
 - activeBandClass, [252](#)
 - activeChannel, [252](#)
 - radiolInterface, [252](#)
 - unpack_nas_GetRFInfo_t, [318](#)
- RFTlv
 - unpack_nas_SetEventReportInd_t, [321](#)
- RMAutoConnect
 - pack_dms_SetCustFeature_t, [183](#)
 - unpack_dms_GetCustFeature_t, [283](#)
- RPTlv
 - NASQmiCbkNasSystemSelPrefInd, [178](#)
- RRTlv
 - unpack_nas_SetEventReportInd_t, [321](#)
- RXChan
 - nas_LTEInfo, [124](#)
- rXDroppedCount
 - unpack_wds_GetPacketStatus_t, [397](#)
- rXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, [397](#)
- rXOkBytesCount
 - unpack_wds_GetPacketStatus_t, [397](#)
- rXPacketErrors
 - unpack_wds_GetPacketStatus_t, [397](#)
- rXPacketOverflows
 - unpack_wds_GetPacketStatus_t, [397](#)
- rXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, [397](#)
- radio
 - unpack_nas_GetSignalStrengths_t, [320](#)
- radiolf
 - nas_ecioListElement, [103](#)
 - nas_errorRateListElement, [104](#)
 - nas_rsrqInformation, [146](#)
 - nas_rxSignalStrengthListElement, [147](#)
- Radiolfaces
 - unpack_dms_GetDeviceCap_t, [283](#)
 - unpack_dms_GetDeviceCapabilities_t, [284](#)
 - unpack_nas_GetServingNetwork_t, [319](#)
- RadiolfacesSize
 - unpack_dms_GetDeviceCap_t, [283](#)
 - unpack_nas_GetServingNetwork_t, [319](#)
- radiolfacesSize
 - unpack_dms_GetDeviceCapabilities_t, [284](#)
- radiolInterface
 - nas_RFInfoTlv, [145](#)
 - nas_roamIndList, [146](#)
 - nas_servSystem, [149](#)
 - nas_SignalStrengthTlv, [150](#)
 - RFBandInfoElements, [252](#)
- radiolInterfaceList
 - NASServingSystemInfo, [180](#)
- radiolInterfaceNo
 - NASServingSystemInfo, [180](#)
- radiolInterfaceSize
 - nas_RFInfoTlv, [145](#)
- range
 - unpack_qos_Port_t, [351](#)
- rat
 - nas_CSGID, [98](#)
 - nas_MNRInfo, [138](#)
- ratMask
 - qmiWSDDataBearerTechnology, [251](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- readTransparent
 - pack_uim_ReadTransparent_t, [228](#)
- reason
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, [362](#)
- reconfigReqd
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [406](#)
- refpn
 - nas_CDMAInfo, [88](#)
- regAction
 - pack_nas_SLQSInitiateNetworkRegistration_t, [198](#)
- RegForeignNID
 - unpack_nas_GetCDMANetworkParameters_t, [314](#)
- RegForeignSID
 - unpack_nas_GetCDMANetworkParameters_t, [315](#)
- RegHomeSID
 - unpack_nas_GetCDMANetworkParameters_t, [315](#)
- regPrd
 - nas_AddCDMASysInfo, [84](#)
- regRejectInfoValid
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_WCDMASysInfo, [170](#)
- regState
 - nas_servSystem, [149](#)
- RegistrationState
 - unpack_nas_GetServingNetwork_t, [319](#)
- registrationState
 - NASServingSystemInfo, [181](#)
- rejCause
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_WCDMASysInfo, [170](#)
- rejectCause
 - nas_RejectReasonTlv, [144](#)
- rejectSrvDomain
 - nas_GSMSysInfo, [113](#)
 - nas_LTESysInfo, [135](#)
 - nas_WCDMASysInfo, [170](#)
- reliabilityClass
 - LibPackGPRSRequestedQoS, [38](#)
 - wds_GPRSQoS, [416](#)
- resBerRatio
 - LibPackUMTSQoS, [69](#)
 - wds_UMTSMInQoS, [422](#)
- resetInfoInd
 - pack_dms_SLQSDmsSwiIndicationRegister_t, [184](#)
- RetryCount

- unpack_swima_SLQSOMADMGetSessionInfo_t, 382
- revTunneling
 - unpack_wds_GetMobileIPProfile_t, 396
- RmTrasnferStaticsReq
 - pack_wds_RMSetTransferStatistics_t, 237
- rmTrasnferStaticsReq, 252
 - bResetStatistics, 252
 - ulMask, 252
- roamIndicator
 - nas_roamIndList, 146
- RoamIndicatorVal
 - unpack_nas_SLQSGetServingSystem_t, 327
- RoamPref
 - NASRoamPreferenceTlv, 179
- roamStatus
 - nas_sysInfoCommon, 156
- roamStatusValid
 - nas_sysInfoCommon, 156
- Roaming
 - nas_QmiNas3GppNetworkInfo, 142
 - unpack_nas_GetCDMANetworkParameters_t, 315
 - unpack_nas_GetServingNetwork_t, 319
- roaming
 - unpack_nas_SetRoamingIndicatorCallback_ind_t, 323
- RoamingIndicatorList
 - unpack_nas_SLQSGetServingSystem_t, 327
- rptRate
 - nas_LTESigRptConfig, 132
- rscp
 - nas_UMTSInfo, 160
 - tdscdmaSigInfoExt, 263
- rsrp
 - lteSSInfo, 82
 - nas_cellParams, 95
 - nas_umtsLTENbrCell, 163
- rsrplevel
 - nas_lteRsrpinformation, 129
- rsrq
 - lteSSInfo, 82
 - nas_cellParams, 95
 - nas_rsrqInformation, 146
 - nas_umtsLTENbrCell, 163
- rsrqDelta
 - nas_SLQSSignalStrengthsIndReq, 151
- rsrqInfo
 - nas_SLQSSignalStrengthsInformation, 151
 - unpack_nas_SLQSGetSignalStrength_t, 329
- rssti
 - cdmaSSInfo, 26
 - hdrSSInfo, 37
 - lteSSInfo, 82
 - nas_cellParams, 95
 - nas_gsmCellInfo, 108
 - tdscdmaSigInfoExt, 263
 - unpack_nas_GetSignalStrengths_t, 320
- rx_bytes
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- rx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- rxLev
 - nas_GERANInfo, 106
- rxOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, 403
- RxQFilter
 - unpack_qos_QosFlowInfo_t, 352
- RxQFlowGranted
 - unpack_qos_QosFlowInfo_t, 353
- rxSignalStrength
 - nas_rxSignalStrengthListElement, 147
- rxSignalStrengthDelta
 - nas_SLQSSignalStrengthsIndReq, 151
- rxSignalStrengthInfo
 - nas_SLQSSignalStrengthsInformation, 151
- rxSignalStrengthList
 - unpack_nas_SLQSGetSignalStrength_t, 329
- rxSignalStrengthListLen
 - unpack_nas_SLQSGetSignalStrength_t, 329
- SCI
 - unpack_nas_GetCDMANetworkParameters_t, 315
- SCM
 - unpack_nas_GetCDMANetworkParameters_t, 315
- SDPTlv
 - NASQmiCbkNasSystemSelPrefInd, 178
- SDU_HDR_LEN
 - common.h, 429
- SHORT
 - SwiDataTypes.h, 567
- sIntraSearch
 - nas_LTEInfoIntrafreq, 127
- SLQSFWINFO_SKU_SZ
 - dms.h, 436
- SLQSSSTlv
 - unpack_nas_SetEventReportInd_t, 322
- sMSCAddress, 256
 - data, 256
 - length, 257
- sMSCAddressInfo
 - sms.h, 559
- sMSCAddressTlv, 257
 - SMSCInfo, 257
 - TlvPresent, 257
- SMSCInfo
 - sMSCAddressTlv, 257
- SMSCtlv
 - unpack_sms_SetNewSMSCallback_ind_t, 374
- sMSEtwmsMessage, 257
 - data, 258
 - length, 258
 - notificationType, 258
- sMSEtwmsMessageInfo
 - sms.h, 559
- sMSEtwmsMessageTlv, 258

- EtwsMessageInfo, [258](#)
- TlvPresent, [258](#)
- sMSEtwsPlmn, [258](#)
 - mobileCountryCode, [259](#)
 - mobileNetworkCode, [259](#)
- sMSEtwsPlmnInfo
 - sms.h, [559](#)
- sMSMTMessage, [259](#)
 - messageIndex, [259](#)
 - storageType, [259](#)
- sMSMTMessageInfo
 - sms.h, [560](#)
- sMSMessageMode, [259](#)
 - messageMode, [259](#)
- sMSMessageModelInfo
 - sms.h, [559](#)
- sMSOnIMS, [260](#)
 - smsOnIMS, [260](#)
- sMSOnIMSInfo
 - sms.h, [560](#)
- sMSOnIMSTlv, [260](#)
 - IMSInfo, [260](#)
 - TlvPresent, [260](#)
- SMSSupport
 - pack_dms_SetCustFeature_t, [183](#)
 - unpack_dms_GetCustFeature_t, [283](#)
- sMSTransferRouteMTMessage, [260](#)
 - ackIndicator, [262](#)
 - data, [262](#)
 - format, [262](#)
 - length, [262](#)
 - transactionID, [262](#)
- sMSTransferRouteMTMessageInfo
 - sms.h, [560](#)
- sNonIntraSearch
 - nas_LTEInfoIntrafreq, [127](#)
- SOMask
 - currNetworkInfo, [27](#)
 - wds_currNetworkInfo, [413](#)
- SSInfo
 - unpack_nas_SetServingSystemCallback_ind_t, [324](#)
- sSLQSSignalStrengthsInfo
 - nas_SLQSSignalStrengthsTlv, [152](#)
- SSTlv
 - unpack_nas_SetEventReportInd_t, [322](#)
- SWI_API
 - SwiDataTypes.h, [567](#)
- SWIWWANCMAPI.h, [579](#)
- scell_idx
 - NASPhyCaAggScellIndex, [175](#)
- scell_state
 - NASPhyCaAggScellIndType, [176](#)
 - NASPhyCaAggScellInfo, [177](#)
- sduErrorRatio
 - LibPackUMTSQoS, [69](#)
 - wds_UMTSMInQoS, [422](#)
- seDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [411](#)
- seDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [411](#)
- sePCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [411](#)
- sePCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [411](#)
- secdns
 - unpack_wds_GetDefaultProfile_t, [394](#)
- secdnsv6
 - unpack_wds_GetDefaultProfile_t, [394](#)
- secondaryDNS
 - pack_wds_SetDefaultProfile_t, [238](#)
- SecondaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- SecondaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- secondaryHA
 - unpack_wds_GetMobileIPProfile_t, [396](#)
- selNetwork
 - nas_servSystem, [149](#)
- selectedNetwork
 - NASServingSystemInfo, [181](#)
- selection
 - pack_swioama_SLQSOMADMSendSelection_t, [225](#)
- ServerAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- serviceDomain
 - nas_RejectReasonTlv, [144](#)
- servingCellId
 - nas_LTEInfoIntrafreq, [127](#)
- ServingSystem
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- sessionEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [406](#)
- SessionId
 - pack_loc_Start_t, [193](#)
 - pack_loc_Stop_t, [193](#)
- sessionId
 - unpack_loc_PositionRpt_Ind_t, [312](#)
- sessionInfo
 - pack_uim_ChangePin_t, [227](#)
 - pack_uim_ReadTransparent_t, [228](#)
 - pack_uim_SetPinProtection_t, [230](#)
 - pack_uim_UnblockPin_t, [232](#)
 - pack_uim_VerifyPin_t, [234](#)
- SessionInfoConfig
 - unpack_swioama_SLQSOMADMAAlertCallback_ind_t, [379](#)
- SessionInfoFota
 - unpack_swioama_SLQSOMADMAAlertCallback_ind_t, [379](#)

- SessionInfoNotification
 - unpack_swima_SLQSOMADMAAlertCallback_ind_t, [379](#)
- SessionState
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- sessionStatus
 - unpack_loc_PositionRpt_Ind_t, [312](#)
 - unpack_omaDmNotificationsTlv_t, [348](#)
- SessionType
 - pack_swima_SLQSOMADMGetSessionInfo_t, [224](#)
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- sessionType
 - pack_swima_SLQSOMADMCancelSession_t, [223](#)
 - pack_swima_SLQSOMADMStartSession_t, [226](#)
 - uim_sessionInformation, [275](#)
 - uim_UIMSessionInformation, [279](#)
 - unpack_omaDmFotaTlv_t, [347](#)
- set_fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, [223](#)
- set_fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, [223](#)
- set_function
 - pack_swiloc_SwiLocSetAutoStart_t, [223](#)
- set_max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, [223](#)
- set_max_time
 - pack_swiloc_SwiLocSetAutoStart_t, [223](#)
- Severity
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- severity
 - unpack_omaDmFotaTlv_t, [347](#)
- shortName
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- shortNameCI
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- shortNameEn
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- shortNameLen
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- shortNameSB
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- sid
 - nas_CDMAInfo, [88](#)
 - unpack_nas_GetHomeNetwork_t, [315](#)
- SigInd
 - LibPackUMTSReqQoSsigInd, [70](#)
- signalStrength
 - nas_SignalStrengthTlv, [150](#)
- signalStrengthReqMask
 - unpack_nas_SLQSGetSignalStrength_t, [329](#)
- SimCapability
 - unpack_dms_GetDeviceCap_t, [284](#)
- simCapability
 - unpack_dms_GetDeviceCapabilities_t, [284](#)
- sinr
 - hdrSSInfo, [37](#)
 - nas_SLQSSignalStrengthsInformation, [152](#)
 - tdscdmaSigInfoExt, [263](#)
 - unpack_nas_SLQSGetSignalStrength_t, [329](#)
- sinrDelta
 - nas_SLQSSignalStrengthsIndReq, [151](#)
- sinrThresholdList
 - nas_SLQSSignalStrengthsIndReq, [151](#)
- sinrThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, [151](#)
- sku_str
 - unpack_dms_GetFirmwareInfo_t, [287](#)
- slot_t, [252](#)
 - bICCID, [253](#)
 - bICCIDLength, [253](#)
 - bLogicalSlot, [253](#)
 - uPhyCardStatus, [253](#)
 - uPhySlotStatus, [253](#)
- slotInf, [253](#)
 - AppStatus, [256](#)
 - cardState, [256](#)
 - errorState, [256](#)
 - numApp, [256](#)
 - upinRetries, [256](#)
 - upinState, [256](#)
 - upukRetries, [256](#)
- SlotInfo
 - uim_cardStatus, [269](#)
- slots_t, [256](#)
 - uimSlotStatus, [256](#)
- slotsstatusChange
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, [388](#)
- SlqsProfile3GPP
 - unpackWdsProfileParam, [411](#)
 - wds_profileInfo, [419](#)
- SlqsProfile3GPP2
 - unpackWdsProfileParam, [411](#)
 - wds_profileInfo, [419](#)
- sms.h
 - LIBPACK_QMI_CBK_PARAM_NOCHANGE, [560](#)
 - LIBPACK_QMI_CBK_PARAM_RESET, [560](#)
 - LIBPACK_QMI_CBK_PARAM_SET, [560](#)
- sms.h, [557](#)
 - eqmiCbKsetStatus, [560](#)
 - MAX_MSE_TWS_MSG, [558](#)
 - MAX_SMS_LIST_SIZE, [559](#)
 - pack_sms_SLQSDeleteSMS, [562](#)
 - pack_sms_SLQSGetSMS, [562](#)
 - pack_sms_SLQSGetSMSList, [563](#)
 - pack_sms_SLQSModifySMSStatus, [563](#)
 - pack_sms_SendSMS, [560](#)
 - pack_sms_SetNewSMSCallback, [562](#)
 - sMSCAddressInfo, [559](#)
 - sMSEtwsMessageInfo, [559](#)
 - sMSEtwsPlmnInfo, [559](#)

- sMSMTMessageInfo, [560](#)
- sMSMessageModelInfo, [559](#)
- sMSOnIMSInfo, [560](#)
- sMSTransferRouteMTMessageInfo, [560](#)
- unpack_sms_SLQSDDeleteSMS, [564](#)
- unpack_sms_SLQSGetSMS, [565](#)
- unpack_sms_SLQSGetSMSList, [565](#)
- unpack_sms_SLQSModifySMSStatus, [565](#)
- unpack_sms_SLQSWmsMemoryFullCallback_ind, [566](#)
- unpack_sms_SendSMS, [563](#)
- unpack_sms_SetNewSMSCallback, [564](#)
- unpack_sms_SetNewSMSCallback_ind, [564](#)
- smsOnIMS
 - sMSOnIMS, [260](#)
- snr
 - lteSSInfo, [82](#)
- snrlevel
 - nas_lteSnrinformation, [132](#)
- soMask
 - qmiWSDDataBearerTechnology, [251](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [408](#)
- Source
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- source
 - unpack_dms_GetNetworkTime_t, [289](#)
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, [295](#)
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, [296](#)
- sourceIPMask
 - LibPackTFTIDParams, [66](#)
- SourceLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- spc
 - pack_nas_SetACCOLC_t, [195](#)
 - pack_wds_SetMobileIPProfile_t, [240](#)
- spn
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- spnEncoding
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- spnLength
 - unpack_nas_SLQSGetPLMNName_t, [325](#)
- srcPortRangeEnd
 - LibPackTFTIDParams, [66](#)
- srcPortRangeStart
 - LibPackTFTIDParams, [66](#)
- srvCapability
 - nas_detailSvcInfo, [102](#)
 - nas_sysInfoCommon, [156](#)
- srvCapabilityValid
 - nas_sysInfoCommon, [156](#)
- srvDomain
 - nas_sysInfoCommon, [156](#)
- SrvDomainPref
 - NASServDomainPrefTlv, [179](#)
- srvDomainValid
 - nas_sysInfoCommon, [156](#)
- srvStatus
 - nas_detailSvcInfo, [102](#)
 - nas_GSMSrvStatusInfo, [110](#)
 - nas_SrvStatusInfo, [153](#)
- srxlev
 - nas_cellParams, [95](#)
 - nas_gsmCellInfo, [108](#)
 - nas_umtsLTENbrCell, [163](#)
 - nas_wcdmaCellInfo, [164](#)
- state
 - unpack_dms_GetActivationState_t, [281](#)
 - unpack_omaDmConfigTlv_t, [345](#)
 - unpack_omaDmFotaTlv_t, [347](#)
 - unpack_qos_QosFlowInfoState_t, [353](#)
- statmask
 - pack_wds_GetPacketStatus_t, [237](#)
- StatsMask
 - transferStatInd, [263](#)
- StatsPeriod
 - transferStatInd, [263](#)
- Status
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- status
 - pack_sms_SetNewSMSCallback_t, [217](#)
 - unpack_qos_SLQSSetQosNWStatusCallback_ind_t, [360](#)
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, [362](#)
- storageIndex
 - FMSImageIdElement, [34](#)
- storageType
 - pack_sms_SLQSDDeleteSMS_t, [218](#)
 - pack_sms_SLQSGetSMS_t, [219](#)
 - pack_sms_SLQSGetSMSList_t, [220](#)
 - pack_sms_SLQSModifySMSStatus_t, [221](#)
 - sMSMTMessage, [259](#)
 - unpack_sms_SLQSWmsMemoryFullCallback_ind_t, [376](#)
- String
 - unpack_dms_GetDeviceHardwareRev_t, [285](#)
 - unpack_dms_GetDeviceMfr_t, [285](#)
 - unpack_dms_GetFSN_t, [288](#)
 - unpack_dms_UIMGetICCID_t, [302](#)
- stringSize
 - unpack_dms_GetDeviceHardwareRev_t, [285](#)
 - unpack_dms_GetDeviceMfr_t, [285](#)
 - unpack_dms_UIMGetICCID_t, [302](#)
- subnetMask
 - unpack_qos_IPv4Addr_t, [349](#)
- SubnetMaskV4
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- SupUSBComps
 - unpack_dms_GetUSBComp_t, [291](#)
- svc
 - pack_qmi_t, [213](#)

- sw1
 - uim_cardResult, [268](#)
- sw2
 - uim_cardResult, [268](#)
- SwiDataTypes.h, [566](#)
 - BOOL, [567](#)
 - BYTE, [567](#)
 - CHAR, [567](#)
 - FLOAT, [567](#)
 - INT32, [567](#)
 - INT8, [567](#)
 - LPCSTR, [567](#)
 - SHORT, [567](#)
 - SWI_API, [567](#)
 - ULONG, [567](#)
 - ULONGLONG, [567](#)
 - UNUSEDPARAM, [567](#)
 - USHORT, [567](#)
 - WORD, [567](#)
- swiloc.h, [567](#)
 - pack_swiloc_SwiLocGetAutoStart, [568](#)
 - pack_swiloc_SwiLocSetAutoStart, [568](#)
 - unpack_swiloc_SwiLocGetAutoStart, [568](#)
 - unpack_swiloc_SwiLocSetAutoStart, [569](#)
- swioma.h, [569](#)
 - pack_swioma_SLQSOMADMAAlertCallback, [570](#)
 - pack_swioma_SLQSOMADMCancelSession, [571](#)
 - pack_swioma_SLQSOMADMGetSessionInfo, [571](#)
 - pack_swioma_SLQSOMADMGetSettings, [572](#)
 - pack_swioma_SLQSOMADMSendSelection, [573](#)
 - pack_swioma_SLQSOMADMSetSettings, [573](#)
 - pack_swioma_SLQSOMADMStartSession, [574](#)
 - unpack_swioma_SLQSOMADMAAlertCallback, [575](#)
 - unpack_swioma_SLQSOMADMAAlertCallback_ind, [575](#)
 - unpack_swioma_SLQSOMADMCancelSession, [576](#)
 - unpack_swioma_SLQSOMADMGetSessionInfo, [576](#)
 - unpack_swioma_SLQSOMADMGetSettings, [577](#)
 - unpack_swioma_SLQSOMADMSendSelection, [577](#)
 - unpack_swioma_SLQSOMADMSetSettings, [578](#)
 - unpack_swioma_SLQSOMADMStartSession, [578](#)
- sysInfoCDMA
 - nas_CDMA SysInfo, [93](#)
- sysInfoGSM
 - nas_GSM SysInfo, [113](#)
- sysInfoHDR
 - nas_HDR SysInfo, [118](#)
- sysInfoLTE
 - nas_LTE SysInfo, [135](#)
- sysInfoWCDMA
 - nas_WCDMA SysInfo, [170](#)
- system
 - loc_SV, [80](#)
- SystemID
 - unpack_nas_SLQSGetServingSystem_t, [327](#)
- systemID
 - nas_CDMA SysInfo, [93](#)
- systemMode
 - nas_CommInfo, [97](#)
- TCPDstPort
 - unpack_qos_swiQosFilter_t, [365](#)
- TCPSrcPort
 - unpack_qos_swiQosFilter_t, [365](#)
- TDSCDMAECIOThreshListLen
 - nas_TDSCDMAECIOThresh, [157](#)
- TDSCDMARSCPTThreshListLen
 - nas_TDSCDMARSCPTThresh, [157](#)
- TDSCDMARSSIThreshListLen
 - nas_TDSCDMARSSIThresh, [158](#)
- TDSCDMASINRThreshListLen
 - nas_TDSCDMASINRThresh, [158](#)
- TRMessageTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [374](#)
- TXChan
 - nas_LTEInfo, [124](#)
- tXDroppedCount
 - unpack_wds_GetPacketStatus_t, [397](#)
- tXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, [397](#)
- tXOkBytesCount
 - unpack_wds_GetPacketStatus_t, [397](#)
- tXPacketErrors
 - unpack_wds_GetPacketStatus_t, [397](#)
- tXPacketOverflows
 - unpack_wds_GetPacketStatus_t, [397](#)
- tXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, [397](#)
- Tables, [19](#)
- tac
 - nas_LTEInfoIntrafreq, [127](#)
 - nas_LTE SysInfo, [135](#)
- tacValid
 - nas_LTE SysInfo, [136](#)
- TdsBandCapability
 - unpack_dms_SLQSGetBandCapability_t, [298](#)
- tdscdmaSigInfoExt, [262](#)
 - ecio, [262](#)
 - rsdp, [263](#)
 - rsi, [263](#)
 - sinr, [263](#)
- techName
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [406](#)
- Technology
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- TechnologyPref
 - pack_nas_SetNetworkPreference_t, [196](#)
- temperature
 - nas_CommInfo, [97](#)
- threshGsmHigh
 - nas_lteGsmCellInfo, [121](#)
- threshGsmLow
 - nas_lteGsmCellInfo, [121](#)

- threshServingLow
 - nas_LTEInfoIntraFreq, 127
- threshXHigh
 - nas_infoInterFreq, 120
- threshXLow
 - nas_infoInterFreq, 120
- threshXhigh
 - nas_lteWcdmaCellInfo, 137
- threshXlow
 - nas_lteWcdmaCellInfo, 137
- Time
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 382
- time
 - NASTimeInfoTlv, 181
- TimeLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 382
- timeTlv
 - NASQmiCbkNasSwiOTAMessageInd, 178
- timeout
 - pack_qmi_t, 213
- timestamp
 - unpack_dms_GetNetworkTime_t, 289
- timingAdvance
 - nas_GERANInfo, 106
- TlvPresent
 - dms_ActivationStatusTlv, 28
 - dms_OperatingModeTlv, 29
 - eTWSPLMNInfoTlv, 32
 - messageModeTlv, 82
 - nas_RejectReasonTlv, 144
 - nas_RFInfoTlv, 145
 - nas_SignalStrengthTlv, 150
 - nas_SLQSSignalStrengthsTlv, 152
 - NASBandPreferenceTlv, 171
 - NASEmergencyModeTlv, 171
 - NASGWAcqOrderPrefTlv, 172
 - NASLTEBandPreferenceTlv, 172
 - NASLteNasReleaseInfoTlv, 172
 - NASModePreferenceTlv, 172
 - NASNetSelPreferenceTlv, 173
 - NASOTAMessageTlv, 173
 - NASPhyCaAggPcellInfo, 174
 - NASPhyCaAggScellIDBw, 175
 - NASPhyCaAggScellIndex, 175
 - NASPhyCaAggScellIndType, 176
 - NASPhyCaAggScellInfo, 177
 - NASPRLPreferenceTlv, 177
 - NASRoamPreferenceTlv, 179
 - NASServDomainPrefTlv, 179
 - NASTimeInfoTlv, 181
 - newMTMessageTlv, 181
 - sMSCAddressTlv, 257
 - sMSEtwMessageTlv, 258
 - sMSOnIMSTlv, 260
 - transferRouteMessageTlv, 263
- Tlvresult
 - pack_dms_GetCustFeaturesV2_t, 182
 - pack_dms_SetCustFeaturesV2_t, 183
 - pack_dms_SetPower_t, 184
 - pack_dms_SetUSBComp_t, 184
 - pack_dms_UIMGetICCID_t, 185
 - pack_fms_GetImagesPreference_t, 186
 - pack_fms_GetStoredImages_t, 186
 - pack_fms_SetImagesPreference_t, 187
 - pack_loc_Delete_Assist_Data_t, 188
 - pack_loc_EventRegister_t, 190
 - pack_loc_SetExtPowerState_t, 190
 - pack_loc_SetOperationMode_t, 191
 - pack_loc_Start_t, 193
 - pack_loc_Stop_t, 193
 - pack_nas_SetNetworkPreference_t, 196
 - pack_uim_ChangePin_t, 227
 - pack_uim_ReadTransparent_t, 229
 - pack_uim_SetPinProtection_t, 230
 - pack_uim_UnblockPin_t, 232
 - pack_uim_VerifyPin_t, 234
 - unpack_dms_GetBandCapability_t, 282
 - unpack_dms_GetCrashAction_t, 282
 - unpack_dms_GetCustFeature_t, 283
 - unpack_dms_GetCustFeaturesV2_t, 283
 - unpack_dms_GetDeviceCap_t, 284
 - unpack_dms_GetDeviceHardwareRev_t, 285
 - unpack_dms_GetDeviceMfr_t, 285
 - unpack_dms_GetDeviceSerialNumbers_t, 286
 - unpack_dms_GetFirmwareInfo_t, 287
 - unpack_dms_GetFirmwareRevision_t, 287
 - unpack_dms_GetFirmwareRevisions_t, 288
 - unpack_dms_GetFSN_t, 288
 - unpack_dms_GetIMSI_t, 288
 - unpack_dms_GetModelID_t, 289
 - unpack_dms_GetNetworkTime_t, 289
 - unpack_dms_GetPower_t, 290
 - unpack_dms_GetPRLVersion_t, 290
 - unpack_dms_GetUSBComp_t, 291
 - unpack_dms_GetVoiceNumber_t, 291
 - unpack_dms_SetCustFeature_t, 292
 - unpack_dms_SetCustFeaturesV2_t, 292
 - unpack_dms_SetEventReport_ind_t, 293
 - unpack_dms_SetEventReport_t, 293
 - unpack_dms_SetFirmwarePreference_t, 293
 - unpack_dms_SetPower_t, 293
 - unpack_dms_SetUSBComp_t, 294
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 295
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 296
 - unpack_dms_SLQSDmsSwiIndicationRegister_t, 296
 - unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 298
 - unpack_dms_SLQSSwiGetDyingGaspCfg_t, 300
 - unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 300
 - unpack_dms_SLQSSwiSetDyingGaspCfg_t, 302
 - unpack_dms_UIMGetICCID_t, 302

- unpack_fms_GetImagesPreference_t, 303
- unpack_fms_GetStoredImages_t, 303
- unpack_fms_SetImagesPreference_t, 304
- unpack_loc_Delete_Assist_Data_t, 305
- unpack_loc_EngineState_Ind_t, 305
- unpack_loc_EventRegister_t, 306
- unpack_loc_PositionRpt_Ind_t, 312
- unpack_loc_SetExtPowerState_t, 312
- unpack_loc_SetOperationMode_t, 313
- unpack_loc_Start_t, 313
- unpack_loc_Stop_t, 314
- unpack_nas_GetNetworkPreference_t, 317
- unpack_nas_SetNetworkPreference_t, 323
- unpack_nas_SetServingSystemCallback_ind_t, 324
- unpack_nas_SlqsGetLTECphyCAInfo_t, 324
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 339
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 340
- unpack_uim_ChangePin_t, 385
- unpack_uim_GetCardStatus_t, 386
- unpack_uim_ReadTransparent_t, 387
- unpack_uim_SetPinProtection_t, 388
- unpack_uim_UnblockPin_t, 390
- unpack_uim_VerifyPin_t, 392
- unpack_wds_SLQSCreateProfile_t, 399
- unpack_wds_SLQSGetProfileSettings_t, 403
- unpack_wds_SLQSSetIPFamilyPreference_t, 406
- TokenBucket
 - unpack_qos_swiQosFlow_t, 370
- tokenRate
 - unpack_qos_tokenBucket_t, 371
- tosMask
 - LibPackTFTIDParams, 66
- total_rx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- total_rx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- total_tx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- total_tx_bytes_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- total_tx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 358
- TrackAreaCode
 - unpack_nas_SLQSGetServingSystem_t, 328
- TrafficClass
 - unpack_qos_swiQosFlow_t, 370
- trafficClass
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- trafficPriority
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- TranDstPort
 - unpack_qos_swiQosFilter_t, 365
- TranSrcPort
 - unpack_qos_swiQosFilter_t, 365
- transactionID
 - sMSTransferRouteMTMessage, 262
- transferDelay
 - LibPackUMTSQoS, 69
 - wds_UMTSMInQoS, 422
- TransferRouteMTMessageInfo
 - transferRouteMessageTlv, 263
- transferRouteMessageTlv, 263
 - TlvPresent, 263
 - TransferRouteMTMessageInfo, 263
- transferStatInd, 263
 - StatsMask, 263
 - StatsPeriod, 263
- transferStats
 - pack_wds_SLQSSetWdsEventCallback_t, 247
- trueSrvStatus
 - nas_GSMSrvStatusInfo, 110
- tx_bytes
 - unpack_QosFlowStat_t, 372
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- tx_bytes_drp
 - unpack_QosFlowStat_t, 372
- tx_pkt
 - unpack_QosFlowStat_t, 372
- tx_pkt_drp
 - unpack_QosFlowStat_t, 372
- tx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- txOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, 403
- TxQFilter
 - unpack_qos_QosFlowInfo_t, 353
- TxQFlowGranted
 - unpack_qos_QosFlowInfo_t, 353
- type
 - pack_wds_GetDefaultProfileNum_t, 235
 - pack_wds_SetDefaultProfileNum_t, 239
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 295
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 296
 - unpack_qmi_t, 348
- u16PRLVersion
 - unpack_dms_GetPRLVersion_t, 290
- u8PRLPreference
 - unpack_dms_GetPRLVersion_t, 290
- UDPDstPort
 - unpack_qos_swiQosFilter_t, 365
- UDPSrcPort
 - unpack_qos_swiQosFilter_t, 365
- ULONG
 - SwiDataTypes.h, 567
- ULONGLONG
 - SwiDataTypes.h, 567

- UMTSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, 405
- UMTSInstInfo
 - nas_UMTSInfo, 160
- UMTSLTENbrCell
 - nas_WCDMAInfoLTENeighborCell, 166
- UMTSReqQoS
 - LibPackUMTSReqQoSSigInd, 70
- UNIQUE_ID_LEN
 - dms.h, 436
- UNUSEDPARAM
 - common.h, 429
 - SwiDataTypes.h, 567
- uPhyCardStatus
 - slot_t, 253
- uPhySlotStatus
 - slot_t, 253
- USBComp
 - pack_dms_SetUSBComp_t, 184
 - unpack_dms_GetUSBComp_t, 291
- USHORT
 - SwiDataTypes.h, 567
- uarfcn
 - nas_lteWcdmaCellInfo, 137
 - nas_UMTSInfo, 160
- uelIdle
 - nas_LTEInfoInterfreq, 124
 - nas_LTEInfoIntrafreq, 127
 - nas_LTEInfoNeighboringGSM, 127
 - nas_LTEInfoNeighboringWCDMA, 128
- uim.h, 579
 - MAX_ICCID_LENGTH, 580
 - MAX_NO_OF_SLOTS, 580
 - MAX_SLOTS_STATUS, 580
 - pack_uim_ChangePin, 581
 - pack_uim_GetCardStatus, 581
 - pack_uim_ReadTransparent, 581
 - pack_uim_SLQSUIEventRegister, 582
 - pack_uim_SLQSUIGetSlotsStatus, 582
 - pack_uim_SLQSUISwitchSlot, 582
 - pack_uim_SetPinProtection, 581
 - pack_uim_UnblockPin, 584
 - pack_uim_VerifyPin, 584
 - unpack_uim_ChangePin, 584
 - unpack_uim_GetCardStatus, 585
 - unpack_uim_ReadTransparent, 585
 - unpack_uim_SLQSUIEventRegister, 586
 - unpack_uim_SLQSUIGetSlotsStatus, 586
 - unpack_uim_SLQSUISetStatusChangeCall-
Back_ind, 587
 - unpack_uim_SLQSUISwitchSlot, 587
 - unpack_uim_SetPinProtection, 585
 - unpack_uim_SetUimSlotStatusChangeCallback_-
ind, 586
 - unpack_uim_UnblockPin, 587
 - unpack_uim_VerifyPin, 588
- uim_UIMSessionInformation, 278
 - aid, 279
 - aidLength, 279
 - sessionType, 279
- uim_appStatus, 264
 - aidLength, 267
 - aidVal, 267
 - appState, 267
 - appType, 267
 - persoFeature, 267
 - persoRetries, 267
 - persoState, 267
 - persoUnblockRetries, 267
 - pin1Retries, 267
 - pin1State, 267
 - pin2Retries, 267
 - pin2State, 267
 - puk1Retries, 267
 - puk2Retries, 267
 - univPin, 267
- uim_cardResult, 267
 - sw1, 268
 - sw2, 268
- uim_cardStatus, 268
 - index1xPri, 269
 - index1xSec, 269
 - indexGwPri, 269
 - indexGwSec, 269
 - numSlot, 269
 - SlotInfo, 269
- uim_changeUIMPIN, 269
 - oldPINLen, 270
 - oldPINVal, 270
 - pinID, 270
 - pinLen, 270
 - pinVal, 270
- uim_encryptedPIN1, 270
 - pin1Len, 271
 - pin1Val, 271
- uim_fileInfo, 271
 - fileID, 271
 - path, 271
 - pathLen, 271
- uim_hotSwapStatus, 272
 - hotSwap, 272
 - hotSwapLength, 272
- uim_readResult, 272
 - content, 273
 - contentLen, 273
- uim_readTransparentInfo, 273
 - length, 273
 - offset, 273
- uim_remainingRetries, 273
 - unblockLeft, 274
 - verifyLeft, 274
- uim_sessionInformation, 274
 - aid, 275
 - aidLength, 275
 - sessionType, 275
- uim_setPINProtection, 275

- pinID, [276](#)
- pinLength, [276](#)
- pinOperation, [276](#)
- pinValue, [276](#)
- uim_slotInfo, [276](#)
 - AppStatus, [278](#)
 - cardState, [278](#)
 - errorState, [278](#)
 - numApp, [278](#)
 - upinRetries, [278](#)
 - upinState, [278](#)
 - upukRetries, [278](#)
- uim_unblockUIMPIN, [279](#)
 - newPINLen, [280](#)
 - newPINVal, [280](#)
 - pinID, [280](#)
 - pukLen, [280](#)
 - pukVal, [280](#)
- uim_verifyUIMPIN, [280](#)
 - pinID, [281](#)
 - pinLen, [281](#)
 - pinVal, [281](#)
- uimSlotStatus
 - slots_t, [256](#)
- ulMask
 - rmTrasferStaticsReq, [252](#)
- ulPhysicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, [231](#)
- umtsEcio
 - nas_UMTSinstInfo, [162](#)
- umtsInst
 - nas_UMTSInfo, [160](#)
- umtsLTENbrCellLen
 - nas_WCDMAInfoLTENeighborCell, [166](#)
- umtsPsc
 - nas_UMTSinstInfo, [162](#)
- umtsRscp
 - nas_UMTSinstInfo, [162](#)
- umtsUarfcn
 - nas_UMTSinstInfo, [162](#)
- UnPackGetProfileSettingOut, [411](#)
 - curProfile, [411](#)
 - pExtErrCode, [411](#)
- unblockLeft
 - uim_remainingRetries, [274](#)
- uniqueID
 - image_info_t, [37](#)
- univPin
 - appStats, [25](#)
 - uim_appStatus, [267](#)
- unpack_QosFlowStat_t, [371](#)
 - bearerId, [372](#)
 - tx_bytes, [372](#)
 - tx_bytes_drp, [372](#)
 - tx_pkt, [372](#)
 - tx_pkt_drp, [372](#)
- unpack_dms_GetActivationState
 - dms.h, [449](#)
- unpack_dms_GetActivationState_t, [281](#)
 - state, [281](#)
- unpack_dms_GetBandCapability
 - dms.h, [449](#)
- unpack_dms_GetBandCapability_t, [281](#)
 - BandCapability, [282](#)
 - Tlvresult, [282](#)
- unpack_dms_GetCrashAction
 - dms.h, [450](#)
- unpack_dms_GetCrashAction_t, [282](#)
 - DevCrashState, [282](#)
 - Tlvresult, [282](#)
- unpack_dms_GetCustFeature
 - dms.h, [450](#)
- unpack_dms_GetCustFeature_t, [282](#)
 - DHCPRelayEnabled, [282](#)
 - DisableIMSI, [282](#)
 - GPSPMP, [282](#)
 - GPSSel, [282](#)
 - GpsEnable, [282](#)
 - IPFamSupport, [282](#)
 - IsVoiceEnabled, [282](#)
 - RMAutoConnect, [283](#)
 - SMSSupport, [283](#)
 - Tlvresult, [283](#)
- unpack_dms_GetCustFeaturesV2
 - dms.h, [450](#)
- unpack_dms_GetCustFeaturesV2_t, [283](#)
 - GetCustomFeatureV2, [283](#)
 - Tlvresult, [283](#)
- unpack_dms_GetDeviceCap
 - dms.h, [450](#)
- unpack_dms_GetDeviceCap_t, [283](#)
 - DataServiceCapability, [283](#)
 - MaxRXChannelRate, [283](#)
 - MaxTXChannelRate, [283](#)
 - Radiolfaces, [283](#)
 - RadiolfacesSize, [283](#)
 - SimCapability, [284](#)
 - Tlvresult, [284](#)
- unpack_dms_GetDeviceCapabilities
 - dms.h, [451](#)
- unpack_dms_GetDeviceCapabilities_t, [284](#)
 - dataServiceCaCapability, [284](#)
 - maxRxChannelRate, [284](#)
 - maxTxChannelRate, [284](#)
 - Radiolfaces, [284](#)
 - radiolfacesSize, [284](#)
 - simCapability, [284](#)
- unpack_dms_GetDeviceHardwareRev
 - dms.h, [451](#)
- unpack_dms_GetDeviceHardwareRev_t, [284](#)
 - String, [285](#)
 - stringSize, [285](#)
 - Tlvresult, [285](#)
- unpack_dms_GetDeviceMfr
 - dms.h, [451](#)
- unpack_dms_GetDeviceMfr_t, [285](#)

- String, [285](#)
- stringSize, [285](#)
- Tlvresult, [285](#)
- unpack_dms_GetDeviceSerialNumbers
 - dms.h, [452](#)
- unpack_dms_GetDeviceSerialNumbers_t, [285](#)
 - ESNString, [285](#)
 - esnSize, [285](#)
 - IMEIString, [285](#)
 - imeiSize, [285](#)
 - imeiSvnSize, [285](#)
 - lmeiSvnString, [285](#)
 - MEIDString, [286](#)
 - meidSize, [285](#)
 - Tlvresult, [286](#)
- unpack_dms_GetFSN
 - dms.h, [453](#)
- unpack_dms_GetFSN_t, [288](#)
 - String, [288](#)
 - Tlvresult, [288](#)
- unpack_dms_GetFirmwareInfo
 - dms.h, [452](#)
- unpack_dms_GetFirmwareInfo_t, [286](#)
 - appversion_str, [286](#)
 - bootversion_str, [286](#)
 - carrier_str, [286](#)
 - cur_carr_name, [286](#)
 - cur_carr_rev, [286](#)
 - modelid_str, [286](#)
 - packageid_str, [286](#)
 - priversion_str, [287](#)
 - sku_str, [287](#)
 - Tlvresult, [287](#)
- unpack_dms_GetFirmwareRevision
 - dms.h, [452](#)
- unpack_dms_GetFirmwareRevision_t, [287](#)
 - AMSSString, [287](#)
 - amssSize, [287](#)
 - Tlvresult, [287](#)
- unpack_dms_GetFirmwareRevisions
 - dms.h, [453](#)
- unpack_dms_GetFirmwareRevisions_t, [287](#)
 - AMSSString, [287](#)
 - amssSize, [287](#)
 - bootSize, [287](#)
 - BootString, [287](#)
 - PRISString, [288](#)
 - priSize, [288](#)
 - Tlvresult, [288](#)
- unpack_dms_GetHardwareRevision
 - dms.h, [453](#)
- unpack_dms_GetHardwareRevision_t, [288](#)
 - hwVer, [288](#)
- unpack_dms_GetIMSI
 - dms.h, [454](#)
- unpack_dms_GetIMSI_t, [288](#)
 - imsi, [288](#)
 - Tlvresult, [288](#)
- unpack_dms_GetModelID
 - dms.h, [454](#)
- unpack_dms_GetModelID_t, [289](#)
 - modelid, [289](#)
 - Tlvresult, [289](#)
- unpack_dms_GetNetworkTime
 - dms.h, [454](#)
- unpack_dms_GetNetworkTime_t, [289](#)
 - source, [289](#)
 - timestamp, [289](#)
 - Tlvresult, [289](#)
- unpack_dms_GetPRLVersion
 - dms.h, [455](#)
- unpack_dms_GetPRLVersion_t, [290](#)
 - Tlvresult, [290](#)
 - u16PRLVersion, [290](#)
 - u8PRLPreference, [290](#)
- unpack_dms_GetPower
 - dms.h, [455](#)
- unpack_dms_GetPower_t, [289](#)
 - HardwareControlledMode, [290](#)
 - OfflineReason, [290](#)
 - OperationMode, [290](#)
 - Tlvresult, [290](#)
- unpack_dms_GetSerialNumbers
 - dms.h, [455](#)
- unpack_dms_GetSerialNumbers_t, [290](#)
 - esn, [291](#)
 - imei_no, [291](#)
 - imeisv_svn, [291](#)
 - meid, [291](#)
- unpack_dms_GetUSBComp
 - dms.h, [456](#)
- unpack_dms_GetUSBComp_t, [291](#)
 - NumSupUSBComps, [291](#)
 - SupUSBComps, [291](#)
 - Tlvresult, [291](#)
 - USBComp, [291](#)
- unpack_dms_GetVoiceNumber
 - dms.h, [456](#)
- unpack_dms_GetVoiceNumber_t, [291](#)
 - MIN, [291](#)
 - minSize, [291](#)
 - Tlvresult, [291](#)
 - VoiceNumber, [292](#)
 - voiceNumberSize, [292](#)
- unpack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [459](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind
 - dms.h, [459](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, [294](#)
 - source, [295](#)
 - Tlvresult, [295](#)
 - type, [295](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_t, [295](#)
 - source, [296](#)
 - Tlvresult, [296](#)
 - type, [296](#)

- unpack_dms_SLQSDmsSwiIndicationRegister
dms.h, 459
- unpack_dms_SLQSDmsSwiIndicationRegister_t, 296
Tlvresult, 296
- unpack_dms_SLQSGetBandCapability
dms.h, 460
- unpack_dms_SLQSGetBandCapability_t, 296
bandCapability, 298
LteBandCapability, 298
TdsBandCapability, 298
- unpack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 460
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 298
Tlvresult, 298
- unpack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 460
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, 298
pGetDyingGaspCfg, 300
Tlvresult, 300
- unpack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 461
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 300
pGetDyingGaspStatistics, 300
Tlvresult, 300
- unpack_dms_SLQSSwiGetFirmwareCurr
dms.h, 461
- unpack_dms_SLQSSwiGetFirmwareCurr_t, 300
carrier, 301
fwvers, 301
numEntries, 301
pCurrImgInfo, 301
pkgver, 301
priver, 301
- unpack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 461
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, 301
Tlvresult, 302
- unpack_dms_SetCustFeature
dms.h, 456
- unpack_dms_SetCustFeature_t, 292
Tlvresult, 292
- unpack_dms_SetCustFeaturesV2
dms.h, 457
- unpack_dms_SetCustFeaturesV2_t, 292
Tlvresult, 292
- unpack_dms_SetEventReport
dms.h, 457
- unpack_dms_SetEventReport_ind
dms.h, 457
- unpack_dms_SetEventReport_ind_t, 292
ActivationStatusTlv, 293
OperatingModeTlv, 293
Tlvresult, 293
- unpack_dms_SetEventReport_t, 293
Tlvresult, 293
- unpack_dms_SetFirmwarePreference
dms.h, 458
- unpack_dms_SetFirmwarePreference_t, 293
Tlvresult, 293
- unpack_dms_SetPower
dms.h, 458
- unpack_dms_SetPower_t, 293
Tlvresult, 293
- unpack_dms_SetUSBComp
dms.h, 458
- unpack_dms_SetUSBComp_t, 294
Tlvresult, 294
- unpack_dms_UIMGetICCID
dms.h, 462
- unpack_dms_UIMGetICCID_t, 302
String, 302
stringSize, 302
Tlvresult, 302
- unpack_fms_GetImagesPreference
fms.h, 465
- unpack_fms_GetImagesPreference_t, 302
ImageListSize, 303
pImageList, 303
Tlvresult, 303
- unpack_fms_GetStoredImages
fms.h, 465
- unpack_fms_GetStoredImages_t, 303
imageList, 303
imagelistSize, 303
Tlvresult, 303
- unpack_fms_SetImagesPreference
fms.h, 465
- unpack_fms_SetImagesPreference_t, 304
ImageTypes, 304
ImageTypesSize, 304
Tlvresult, 304
- unpack_loc_Delete_Assist_Data_t, 304
Tlvresult, 305
- unpack_loc_DeleteAssistData
loc.h, 472
- unpack_loc_EngineState_Ind
loc.h, 472
- unpack_loc_EngineState_Ind_t, 305
engineState, 305
Tlvresult, 305
- unpack_loc_EventRegister
loc.h, 472
- unpack_loc_EventRegister_t, 305
Tlvresult, 306
- unpack_loc_PositionRpt_Ind
loc.h, 473
- unpack_loc_PositionRpt_Ind_t, 306
pAltitudeAssumed, 311
pAltitudeWrtEllipsoid, 311
pAltitudeWrtMeanSeaLevel, 311
pFixId, 311
pGpsTime, 311
pHeading, 311
pHeadingUnc, 311
pHorConfidence, 311

- pHorReliability, [311](#)
- pHorUncCircular, [311](#)
- pHorUncEllipseOrientAzimuth, [311](#)
- pHorUncEllipseSemiMajor, [311](#)
- pHorUncEllipseSemiMinor, [311](#)
- pLatitude, [311](#)
- pLeapSeconds, [311](#)
- pLongitude, [311](#)
- pMagneticDeviation, [311](#)
- pPrecisionDilution, [311](#)
- pSensorDataUsage, [312](#)
- pSpeedHorizontal, [312](#)
- pSpeedUnc, [312](#)
- pSpeedVertical, [312](#)
- pSvUsedforFix, [312](#)
- pTechnologyMask, [312](#)
- pTimeSrc, [312](#)
- pTimeUnc, [312](#)
- pTimestampUtc, [312](#)
- pVertConfidence, [312](#)
- pVertReliability, [312](#)
- pVertUnc, [312](#)
- sessionId, [312](#)
- sessionStatus, [312](#)
- Tlvresult, [312](#)
- unpack_loc_SetExtPowerState
 - loc.h, [473](#)
- unpack_loc_SetExtPowerState_t, [312](#)
 - Tlvresult, [312](#)
- unpack_loc_SetOperationMode
 - loc.h, [473](#)
- unpack_loc_SetOperationMode_t, [312](#)
 - Tlvresult, [313](#)
- unpack_loc_Start
 - loc.h, [474](#)
- unpack_loc_Start_t, [313](#)
 - Tlvresult, [313](#)
- unpack_loc_Stop
 - loc.h, [474](#)
- unpack_loc_Stop_t, [313](#)
 - Tlvresult, [314](#)
- unpack_nas_GetACCOLC
 - nas.h, [491](#)
- unpack_nas_GetANAAAAAuthenticationStatus
 - nas.h, [491](#)
- unpack_nas_GetCDMANetworkParameters
 - nas.h, [491](#)
- unpack_nas_GetCDMANetworkParameters_t, [314](#)
 - Application, [314](#)
 - Broadcast, [314](#)
 - CustomSCP, [314](#)
 - ForceRev0, [314](#)
 - Protocol, [314](#)
 - RegForeignNID, [314](#)
 - RegForeignSID, [315](#)
 - RegHomeSID, [315](#)
 - Roaming, [315](#)
 - SCI, [315](#)
 - SCM, [315](#)
- unpack_nas_GetHomeNetwork
 - nas.h, [492](#)
- unpack_nas_GetHomeNetwork_t, [315](#)
 - mcc, [315](#)
 - mnc, [315](#)
 - name, [315](#)
 - nid, [315](#)
 - sid, [315](#)
- unpack_nas_GetNetworkPreference
 - nas.h, [492](#)
- unpack_nas_GetNetworkPreference_t, [315](#)
 - ActiveTechPref, [317](#)
 - Duration, [317](#)
 - PersistentTechPref, [317](#)
 - Tlvresult, [317](#)
- unpack_nas_GetRFInfo
 - nas.h, [492](#)
- unpack_nas_GetRFInfo_t, [317](#)
 - instancesSize, [318](#)
 - RFBandInfoElements, [318](#)
- unpack_nas_GetServingNetwork
 - nas.h, [492](#)
- unpack_nas_GetServingNetwork_t, [318](#)
 - CSDomain, [319](#)
 - DataCaps, [319](#)
 - DataCapsLen, [319](#)
 - MCC, [319](#)
 - MNC, [319](#)
 - Name, [319](#)
 - nameSize, [319](#)
 - PSDomain, [319](#)
 - RAN, [319](#)
 - Radiofaces, [319](#)
 - RadiofacesSize, [319](#)
 - RegistrationState, [319](#)
 - Roaming, [319](#)
- unpack_nas_GetServingNetworkCapabilities
 - nas.h, [493](#)
- unpack_nas_GetServingNetworkCapabilities_t, [319](#)
 - DataCaps, [319](#)
 - DataCapsLen, [319](#)
- unpack_nas_GetSignalStrengths
 - nas.h, [493](#)
- unpack_nas_GetSignalStrengths_t, [320](#)
 - len, [320](#)
 - radio, [320](#)
 - rsi, [320](#)
- unpack_nas_PerformNetworkScan
 - nas.h, [493](#)
- unpack_nas_PerformNetworkScan_t, [320](#)
 - p3GppNetworkInfoInstances, [320](#)
 - p3GppNetworkInstanceSize, [320](#)
 - pPDSInstance, [320](#)
 - pPDSInstanceSize, [321](#)
 - pPRATInstance, [321](#)
 - pPRATInstanceSize, [321](#)
 - pScanResult, [321](#)

- unpack_nas_SLQSGetPLMNName
 - nas.h, [495](#)
- unpack_nas_SLQSGetPLMNName_t, [324](#)
 - longName, [325](#)
 - longNameCI, [325](#)
 - longNameEn, [325](#)
 - longNameLen, [325](#)
 - longNameSB, [325](#)
 - shortName, [325](#)
 - shortNameCI, [325](#)
 - shortNameEn, [325](#)
 - shortNameLen, [325](#)
 - shortNameSB, [325](#)
 - spn, [325](#)
 - spnEncoding, [325](#)
 - spnLength, [325](#)
- unpack_nas_SLQSGetServingSystem
 - nas.h, [496](#)
- unpack_nas_SLQSGetServingSystem_t, [325](#)
 - BasestationID, [327](#)
 - BasestationLatitude, [327](#)
 - BasestationLongitude, [327](#)
 - CDMASystemInfoExt, [327](#)
 - CallBarStatus, [327](#)
 - CellID, [327](#)
 - ConcSvcInfo, [327](#)
 - CurrentPLMN, [327](#)
 - DTMInd, [327](#)
 - DataSrvCapabilities, [327](#)
 - DefaultRoamInd, [327](#)
 - DetailedSvcInfo, [327](#)
 - Gpp2TimeZone, [327](#)
 - GppNetworkDSTAdjustment, [327](#)
 - GppTimeZone, [327](#)
 - HdrPersonality, [327](#)
 - Lac, [327](#)
 - NetworkID, [327](#)
 - PRLInd, [327](#)
 - RoamIndicatorVal, [327](#)
 - RoamingIndicatorList, [327](#)
 - ServingSystem, [327](#)
 - SystemID, [327](#)
 - TrackAreaCode, [328](#)
- unpack_nas_SLQSGetSignalStrength
 - nas.h, [496](#)
- unpack_nas_SLQSGetSignalStrength_t, [328](#)
 - ecioList, [328](#)
 - ecioListLen, [328](#)
 - errorRateList, [328](#)
 - errorRateListLen, [328](#)
 - lo, [328](#)
 - ltersrp, [328](#)
 - ltesn, [328](#)
 - rsrqInfo, [329](#)
 - rxSignalStrengthList, [329](#)
 - rxSignalStrengthListLen, [329](#)
 - signalStrengthReqMask, [329](#)
 - sinr, [329](#)
- unpack_nas_SLQSGetSysInfo
 - nas.h, [496](#)
- unpack_nas_SLQSGetSysInfo_t, [329](#)
 - pAddCDMASysInfo, [331](#)
 - pAddGSMSSysInfo, [331](#)
 - pAddHDRSysInfo, [331](#)
 - pAddLTESysInfo, [331](#)
 - pCDMASrvStatusInfo, [331](#)
 - pCDMASysInfo, [331](#)
 - pGSMCallBarringSysInfo, [331](#)
 - pGSMCipherDomainSysInfo, [331](#)
 - pGSMSSrvStatusInfo, [331](#)
 - pGSMSSysInfo, [331](#)
 - pHDRSrvStatusInfo, [331](#)
 - pHDRSysInfo, [331](#)
 - pLTESrvStatusInfo, [331](#)
 - pLTESysInfo, [331](#)
 - pLTEVoiceSupportSysInfo, [331](#)
 - pWCDMASysInfo, [331](#)
- unpack_nas_SLQSGetSysSelectionPref
 - nas.h, [497](#)
- unpack_nas_SLQSGetSysSelectionPref_t, [331](#)
 - pBandPref, [335](#)
 - pEmerMode, [335](#)
 - pGWAcqOrderPref, [335](#)
 - pLTEBandPref, [335](#)
 - pModePref, [335](#)
 - pNetSelPref, [335](#)
 - pPRLPref, [335](#)
 - pRoamPref, [335](#)
 - pSrvDomainPref, [335](#)
- unpack_nas_SLQSIInitiateNetworkRegistration
 - nas.h, [497](#)
- unpack_nas_SLQSNasConfigSigInfo2
 - nas.h, [497](#)
- unpack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [498](#)
- unpack_nas_SLQSNasGetCellLocationInfo_t, [335](#)
 - pCDMAInfo, [336](#)
 - pLTEInfoInterfreq, [336](#)
 - pLTEInfoIntrafreq, [337](#)
 - pUMTSInfo, [337](#)
- unpack_nas_SLQSNasGetSigInfo
 - nas.h, [498](#)
- unpack_nas_SLQSNasGetSigInfo_t, [337](#)
 - CDMASSInfo, [337](#)
 - GSMSSInfo, [337](#)
 - HDRSSInfo, [337](#)
 - LTESInfo, [337](#)
- unpack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [498](#)
- unpack_nas_SLQSNasSigInfoCallback
 - nas.h, [499](#)
- unpack_nas_SLQSNasSigInfoCallback_t, [337](#)
 - pCDMASigInfo, [338](#)
 - pGSMSSigInfo, [338](#)
 - pHDRSigInfo, [338](#)
 - pLTESigInfo, [338](#)

- pRscp, [338](#)
- unpack_nas_SLQSNasSwtModemStatus
 - nas.h, [499](#)
- unpack_nas_SLQSNasSwtModemStatus_t, [338](#)
 - commonInfo, [339](#)
 - pLTEInfo, [339](#)
- unpack_nas_SLQSNasSwtOTAMessageCallback
 - nas.h, [499](#)
- unpack_nas_SLQSNasSwtOTAMessageCallback_ind
 - nas.h, [500](#)
- unpack_nas_SLQSNasSwtOTAMessageCallback_ind_t, [339](#)
 - Info, [339](#)
 - Tlvresult, [339](#)
- unpack_nas_SLQSNasSysInfoCallback
 - nas.h, [500](#)
- unpack_nas_SLQSSetBandPreference
 - nas.h, [500](#)
- unpack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [501](#)
- unpack_nas_SLQSSetSysSelectionPref
 - nas.h, [501](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
 - nas.h, [501](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, [339](#)
 - Info, [340](#)
 - Tlvresult, [340](#)
- unpack_nas_SLQSSwtGetLteCQI
 - nas.h, [501](#)
- unpack_nas_SLQSSwtGetLteCQI_t, [340](#)
 - ValidityCW0, [341](#)
 - ValidityCW1, [341](#)
- unpack_nas_SLQSSysInfoCallback_t, [341](#)
 - pAddCDMASysInfo, [343](#)
 - pAddGSMSysInfo, [343](#)
 - pAddHDRSysInfo, [343](#)
 - pAddLTESysInfo, [343](#)
 - pCDMASrvStatusInfo, [343](#)
 - pCDMASysInfo, [343](#)
 - pGSMCallBarringSysInfo, [343](#)
 - pGSMCipherDomainSysInfo, [343](#)
 - pGSMSrvStatusInfo, [343](#)
 - pGSMSysInfo, [344](#)
 - pHRSrvStatusInfo, [344](#)
 - pHRSysInfo, [344](#)
 - pLTESrvStatusInfo, [344](#)
 - pLTESysInfo, [344](#)
 - pLTEVoiceSupportSysInfo, [344](#)
 - pSysInfoNoChange, [344](#)
 - pWCDMASysInfo, [344](#)
- unpack_nas_SetACCOLC
 - nas.h, [494](#)
- unpack_nas_SetDataCapabilitiesCallback_ind
 - nas.h, [494](#)
- unpack_nas_SetDataCapabilitiesCallback_ind_t, [321](#)
 - dataCaps, [321](#)
 - dataCapsSize, [321](#)
- unpack_nas_SetEventReportInd
 - nas.h, [494](#)
- unpack_nas_SetEventReportInd_t, [321](#)
 - RFTlv, [321](#)
 - RRTlv, [321](#)
 - SLQSSSTlv, [322](#)
 - SSTlv, [322](#)
- unpack_nas_SetLURejectCallback
 - nas.h, [494](#)
- unpack_nas_SetNetworkPreference
 - nas.h, [495](#)
- unpack_nas_SetNetworkPreference_t, [322](#)
 - Tlvresult, [323](#)
- unpack_nas_SetRFInfoCallback
 - nas.h, [495](#)
- unpack_nas_SetRoamingIndicatorCallback_ind
 - nas.h, [495](#)
- unpack_nas_SetRoamingIndicatorCallback_ind_t, [323](#)
 - roaming, [323](#)
- unpack_nas_SetServingSystemCallback_ind
 - nas.h, [495](#)
- unpack_nas_SetServingSystemCallback_ind_t, [323](#)
 - SSInfo, [324](#)
 - Tlvresult, [324](#)
- unpack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [495](#)
- unpack_nas_SlqsGetLTECphyCAInfo_t, [324](#)
 - LTECphyCAInfo, [324](#)
 - Tlvresult, [324](#)
- unpack_omaDmConfigTlv_t, [344](#)
 - alertmsg, [345](#)
 - alertmsglength, [345](#)
 - state, [345](#)
 - userInputReq, [345](#)
 - userInputTimeout, [345](#)
- unpack_omaDmFotaTlv_t, [345](#)
 - description, [347](#)
 - descriptionlength, [347](#)
 - fwloadsize, [347](#)
 - fwloadComplete, [347](#)
 - namelength, [347](#)
 - package_name, [347](#)
 - sessionType, [347](#)
 - severity, [347](#)
 - state, [347](#)
 - updateCompleteStatus, [347](#)
 - userInputReq, [347](#)
 - userInputTimeout, [347](#)
 - version, [347](#)
 - versionlength, [347](#)
- unpack_omaDmNotificationsTlv_t, [347](#)
 - notification, [348](#)
 - sessionStatus, [348](#)
- unpack_qmi_t, [348](#)
 - msgid, [348](#)
 - type, [348](#)
 - xid, [348](#)
- unpack_qos_IPv4Addr_t, [349](#)

- addr, [349](#)
- subnetMask, [349](#)
- unpack_qos_IPv6Addr_t, [349](#)
 - addr, [349](#)
 - prefixLen, [349](#)
- unpack_qos_IPv6TrafCls_t, [349](#)
 - mask, [350](#)
 - val, [350](#)
- unpack_qos_Port_t, [350](#)
 - port, [351](#)
 - range, [351](#)
- unpack_qos_QosFlowInfo_t, [351](#)
 - BearerID, [352](#)
 - is_RxQFlowGranted_Available, [352](#)
 - is_TxQFlowGranted_Available, [352](#)
 - NumRxFilters, [352](#)
 - NumTxFilters, [352](#)
 - QFlowState, [352](#)
 - RxQFilter, [352](#)
 - RxQFlowGranted, [353](#)
 - TxQFilter, [353](#)
 - TxQFlowGranted, [353](#)
- unpack_qos_QosFlowInfoState_t, [353](#)
 - id, [353](#)
 - isNewFlow, [353](#)
 - state, [353](#)
- unpack_qos_SLQSQosGetNetworkStatus
 - qos.h, [550](#)
- unpack_qos_SLQSQosGetNetworkStatus_t, [353](#)
 - NWQoSStatus, [355](#)
- unpack_qos_SLQSQosSwiReadApnExtraParams
 - qos.h, [552](#)
- unpack_qos_SLQSQosSwiReadApnExtraParams_t, [355](#)
 - ambr_dl, [356](#)
 - ambr_dl_ext, [356](#)
 - ambr_dl_ext2, [356](#)
 - ambr_ul, [356](#)
 - ambr_ul_ext, [356](#)
 - ambr_ul_ext2, [356](#)
 - apnId, [356](#)
- unpack_qos_SLQSQosSwiReadDataStats
 - qos.h, [552](#)
- unpack_qos_SLQSQosSwiReadDataStats_t, [356](#)
 - apnId, [357](#)
 - numQosFlow, [358](#)
 - qosFlow, [358](#)
 - total_rx_bytes, [358](#)
 - total_rx_pkt, [358](#)
 - total_tx_bytes, [358](#)
 - total_tx_pkt, [358](#)
- unpack_qos_SLQSSetQosEventCallback
 - qos.h, [553](#)
- unpack_qos_SLQSSetQosEventCallback_ind
 - qos.h, [553](#)
- unpack_qos_SLQSSetQosEventCallback_ind_t, [358](#)
 - NumFlows, [358](#)
 - QosFlowInfo, [358](#)
- unpack_qos_SLQSSetQosNWStatusCallback_ind
 - qos.h, [554](#)
- unpack_qos_SLQSSetQosNWStatusCallback_ind_t, [358](#)
 - status, [360](#)
- unpack_qos_SLQSSetQosPriEventCallback_ind
 - qos.h, [554](#)
- unpack_qos_SLQSSetQosPriEventCallback_ind_t, [360](#)
 - event, [360](#)
- unpack_qos_SLQSSetQosStatusCallback_ind
 - qos.h, [556](#)
- unpack_qos_SLQSSetQosStatusCallback_ind_t, [360](#)
 - event, [362](#)
 - id, [362](#)
 - reason, [362](#)
 - status, [362](#)
- unpack_qos_Tos_t, [371](#)
 - mask, [371](#)
 - val, [371](#)
- unpack_qos_dataRate_t, [348](#)
 - dataRateMax, [348](#)
 - guaranteedRate, [348](#)
- unpack_qos_pktErrRate_t, [350](#)
 - exponent, [350](#)
 - multiplier, [350](#)
- unpack_qos_swiQosFilter_t, [362](#)
 - EspSpi, [364](#)
 - IPv4DstAddr, [364](#)
 - IPv4SrcAddr, [364](#)
 - IPv4Tos, [364](#)
 - IPv6DstAddr, [364](#)
 - IPv6Label, [364](#)
 - IPv6SrcAddr, [364](#)
 - IPv6TrafCls, [364](#)
 - Id, [364](#)
 - index, [364](#)
 - is_EspSpi_Available, [364](#)
 - is_IPv4DstAddr_Available, [364](#)
 - is_IPv4SrcAddr_Available, [365](#)
 - is_IPv4Tos_Available, [365](#)
 - is_IPv6DstAddr_Available, [365](#)
 - is_IPv6Label_Available, [365](#)
 - is_IPv6SrcAddr_Available, [365](#)
 - is_IPv6TrafCls_Available, [365](#)
 - is_Id_Available, [364](#)
 - is_NxtHdrProto_Available, [365](#)
 - is_Precedence_Available, [365](#)
 - is_TCPDstPort_Available, [365](#)
 - is_TCPSrcPort_Available, [365](#)
 - is_TranDstPort_Available, [365](#)
 - is_TranSrcPort_Available, [365](#)
 - is_UDPDstPort_Available, [365](#)
 - is_UDPSrcPort_Available, [365](#)
 - NxtHdrProto, [365](#)
 - Precedence, [365](#)
 - TCPDstPort, [365](#)
 - TCPSrcPort, [365](#)
 - TranDstPort, [365](#)

- TranSrcPort, [365](#)
- UDPDstPort, [365](#)
- UDPSrcPort, [365](#)
- version, [365](#)
- unpack_qos_swiQosFlow_t, [365](#)
 - DataRate, [369](#)
 - index, [369](#)
 - is_DataRate_Available, [369](#)
 - is_Jitter_Available, [369](#)
 - is_Latency_Available, [369](#)
 - is_LteQci_Available, [369](#)
 - is_MaxAllowedPktSz_Available, [369](#)
 - is_MinPolicedPktSz_Available, [369](#)
 - is_PktErrRate_Available, [369](#)
 - is_ProfileId3GPP2_Available, [369](#)
 - is-TokenBucket_Available, [369](#)
 - is_TrafficClass_Available, [369](#)
 - is_val_3GPP2Pri_Available, [369](#)
 - is_val_3GPPImCn_Available, [369](#)
 - is_val_3GPPSigInd_Available, [369](#)
 - Jitter, [370](#)
 - Latency, [370](#)
 - LteQci, [370](#)
 - MaxAllowedPktSz, [370](#)
 - MinPolicedPktSz, [370](#)
 - PktErrRate, [370](#)
 - ProfileId3GPP2, [370](#)
 - TokenBucket, [370](#)
 - TrafficClass, [370](#)
 - val_3GPP2Pri, [370](#)
 - val_3GPPImCn, [370](#)
 - val_3GPPResResidualBER, [370](#)
 - val_3GPPSigInd, [370](#)
 - val_3GPPTraHdlPri, [370](#)
- unpack_qos_tokenBucket_t, [370](#)
 - bucketSz, [370](#)
 - peakRate, [370](#)
 - tokenRate, [371](#)
- unpack_result_code_only
 - common.h, [431](#)
- unpack_sms_SLQSDDeleteSMS
 - sms.h, [564](#)
- unpack_sms_SLQSDDeleteSMS_t, [374](#)
- unpack_sms_SLQSGetSMS
 - sms.h, [565](#)
- unpack_sms_SLQSGetSMS_t, [374](#)
 - message, [375](#)
 - messageFormat, [375](#)
 - messageSize, [375](#)
 - messageTag, [375](#)
- unpack_sms_SLQSGetSMSList
 - sms.h, [565](#)
- unpack_sms_SLQSGetSMSList_t, [375](#)
 - messageList, [376](#)
 - messageListSize, [376](#)
- unpack_sms_SLQSModifySMSStatus
 - sms.h, [565](#)
- unpack_sms_SLQSModifySMSStatus_t, [376](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind
 - sms.h, [566](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, [376](#)
 - messageMode, [376](#)
 - storageType, [376](#)
- unpack_sms_SendSMS
 - sms.h, [563](#)
- unpack_sms_SendSMS_t, [372](#)
 - messageFailureCode, [372](#)
 - messageID, [372](#)
- unpack_sms_SetNewSMSCallback
 - sms.h, [564](#)
- unpack_sms_SetNewSMSCallback_ind
 - sms.h, [564](#)
- unpack_sms_SetNewSMSCallback_ind_t, [373](#)
 - ETWSTlv, [374](#)
 - IMSTlv, [374](#)
 - MMTlv, [374](#)
 - NewMMTlv, [374](#)
 - SMSCTlv, [374](#)
 - TRMessageTlv, [374](#)
- unpack_sms_SetNewSMSCallback_t, [374](#)
- unpack_swiloc_SwiLocGetAutoStart
 - swiloc.h, [568](#)
- unpack_swiloc_SwiLocGetAutoStart_t, [376](#)
 - fix_rate, [378](#)
 - fix_rate_reported, [378](#)
 - fix_type, [378](#)
 - fix_type_reported, [378](#)
 - function, [378](#)
 - function_reported, [378](#)
 - max_dist, [378](#)
 - max_dist_reported, [378](#)
 - max_time, [378](#)
 - max_time_reported, [378](#)
- unpack_swiloc_SwiLocSetAutoStart
 - swiloc.h, [569](#)
- unpack_swioma_SLQSOMADMAAlertCallback
 - swioma.h, [575](#)
- unpack_swioma_SLQSOMADMAAlertCallback_ind
 - swioma.h, [575](#)
- unpack_swioma_SLQSOMADMAAlertCallback_ind_t, [378](#)
 - eventType, [379](#)
- unpack_swioma_SLQSOMADMCancelSession
 - swioma.h, [576](#)
- unpack_swioma_SLQSOMADMGetSessionInfo
 - swioma.h, [576](#)
- unpack_swioma_SLQSOMADMGetSessionInfo_t, [379](#)
 - Date, [381](#)
 - DateLength, [382](#)
 - PkgDescLength, [382](#)
 - PkgDescription, [382](#)
 - PkgName, [382](#)
 - PkgNameLength, [382](#)
 - RetryCount, [382](#)
 - SessionState, [382](#)

- SessionType, [382](#)
- Severity, [382](#)
- Source, [382](#)
- SourceLength, [382](#)
- Status, [382](#)
- Time, [382](#)
- TimeLength, [382](#)
- UpdateCompleteStatus, [382](#)
- unpack_swisma_SLQSOMADMGetSettings
 - swisma.h, [577](#)
- unpack_swisma_SLQSOMADMGetSettings_t, [382](#)
 - Autosdm, [384](#)
 - FOTAdownload, [384](#)
 - FwAutoCheck, [384](#)
- unpack_swisma_SLQSOMADMSelectSelection
 - swisma.h, [577](#)
- unpack_swisma_SLQSOMADMSetSettings
 - swisma.h, [578](#)
- unpack_swisma_SLQSOMADMStartSession
 - swisma.h, [578](#)
- unpack_swisma_SLQSOMADMStartSession_t, [384](#)
 - FwAvailability, [384](#)
- unpack_uim_ChangePin
 - uim.h, [584](#)
- unpack_uim_ChangePin_t, [384](#)
 - pEncryptedPIN1, [385](#)
 - pIndicationToken, [385](#)
 - pRemainingRetries, [385](#)
 - Tlvresult, [385](#)
- unpack_uim_GetCardStatus
 - uim.h, [585](#)
- unpack_uim_GetCardStatus_t, [385](#)
 - pCardStatus, [386](#)
 - pHotSwapStatus, [386](#)
 - Tlvresult, [386](#)
- unpack_uim_ReadTransparent
 - uim.h, [585](#)
- unpack_uim_ReadTransparent_t, [386](#)
 - pCardResult, [387](#)
 - pEncryptedData, [387](#)
 - pIndicationToken, [387](#)
 - pReadResult, [387](#)
 - Tlvresult, [387](#)
- unpack_uim_SLQSUIMEventRegister
 - uim.h, [586](#)
- unpack_uim_SLQSUIMEventRegister_t, [388](#)
 - eventMask, [389](#)
- unpack_uim_SLQSUIMGetSlotsStatus
 - uim.h, [586](#)
- unpack_uim_SLQSUIMGetSlotsStatus_t, [389](#)
 - pNumberOfPhySlot, [389](#)
 - pUimSlotsStatus, [389](#)
- unpack_uim_SLQSUIMSetStatusChangeCallBack_ind
 - uim.h, [587](#)
- unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, [389](#)
- unpack_uim_SLQSUIMSwitchSlot
 - uim.h, [587](#)
- unpack_uim_SetPinProtection
 - uim.h, [585](#)
- unpack_uim_SetPinProtection_t, [387](#)
 - pEncryptedPIN1, [388](#)
 - pIndicationToken, [388](#)
 - pRemainingRetries, [388](#)
 - Tlvresult, [388](#)
- unpack_uim_SetUimSlotStatusChangeCallback_ind
 - uim.h, [586](#)
- unpack_uim_SetUimSlotStatusChangeCallback_ind_t, [388](#)
 - bNumberOfPhySlots, [388](#)
 - slotsstatusChange, [388](#)
- unpack_uim_UnblockPin
 - uim.h, [587](#)
- unpack_uim_UnblockPin_t, [390](#)
 - pEncryptedPIN1, [390](#)
 - pIndicationToken, [390](#)
 - pRemainingRetries, [390](#)
 - Tlvresult, [390](#)
- unpack_uim_VerifyPin
 - uim.h, [588](#)
- unpack_uim_VerifyPin_t, [390](#)
 - pEncryptedPIN1, [392](#)
 - pIndicationToken, [392](#)
 - pRemainingRetries, [392](#)
 - Tlvresult, [392](#)
- unpack_wds_GetConnectionRate
 - wds.h, [606](#)
- unpack_wds_GetConnectionRate_t, [392](#)
 - currentChannelRXRate, [393](#)
 - currentChannelTXRate, [393](#)
 - maxChannelRXRate, [393](#)
 - maxChannelTXRate, [393](#)
- unpack_wds_GetDefaultProfile
 - wds.h, [606](#)
- unpack_wds_GetDefaultProfile_t, [393](#)
 - apnname, [393](#)
 - apnsize, [393](#)
 - auth, [393](#)
 - ipaddr, [394](#)
 - ipaddrv6, [394](#)
 - name, [394](#)
 - namesize, [394](#)
 - pdptype, [394](#)
 - pridns, [394](#)
 - pridnsv6, [394](#)
 - secdns, [394](#)
 - secdnsv6, [394](#)
 - username, [394](#)
 - usersize, [394](#)
- unpack_wds_GetDefaultProfileNum
 - wds.h, [607](#)
- unpack_wds_GetDefaultProfileNum_t, [394](#)
 - index, [394](#)
- unpack_wds_GetDormancyState
 - wds.h, [607](#)
- unpack_wds_GetDormancyState_t, [394](#)

- dormancyState, 395
- unpack_wds_GetLastMobileIPError
 - wds.h, 607
- unpack_wds_GetLastMobileIPError_t, 395
 - error, 395
- unpack_wds_GetMobileIP
 - wds.h, 608
- unpack_wds_GetMobileIP_t, 395
 - mipMode, 395
- unpack_wds_GetMobileIPProfile
 - wds.h, 608
- unpack_wds_GetMobileIPProfile_t, 395
 - AAASPI, 396
 - AAASState, 396
 - address, 396
 - enabled, 396
 - HASPI, 396
 - HASState, 396
 - NAI, 396
 - naiSize, 396
 - primaryHA, 396
 - revTunneling, 396
 - secondaryHA, 396
- unpack_wds_GetPacketStatus
 - wds.h, 608
- unpack_wds_GetPacketStatus_t, 396
 - rXDroppedCount, 397
 - rXOKBytesLastCall, 397
 - rXOKBytesCount, 397
 - rXPacketErrors, 397
 - rXPacketOverflows, 397
 - rXPacketSuccesses, 397
 - tXDroppedCount, 397
 - tXOKBytesLastCall, 397
 - tXOKBytesCount, 397
 - tXPacketErrors, 397
 - tXPacketOverflows, 397
 - tXPacketSuccesses, 397
- unpack_wds_GetSessionDuration
 - wds.h, 609
- unpack_wds_GetSessionDuration_t, 398
 - callDuration, 398
- unpack_wds_GetSessionState
 - wds.h, 609
- unpack_wds_GetSessionState_t, 398
 - connectionStatus, 398
- unpack_wds_RMSetTransferStatistics
 - wds.h, 609
- unpack_wds_RMSetTransferStatistics_t, 398
- unpack_wds_SLQSCreateProfile
 - wds.h, 611
- unpack_wds_SLQSCreateProfile_t, 398
 - pCreateProfileOut, 399
 - pProfileID, 399
 - Tlvresult, 399
- unpack_wds_SLQSDeleteProfile
 - wds.h, 611
- unpack_wds_SLQSDeleteProfile_t, 399
 - extendedErrorCode, 399
- unpack_wds_SLQSGet3GPPConfigItem
 - wds.h, 611
- unpack_wds_SLQSGet3GPPConfigItem_t, 399
 - _3gppRelease, 400
 - defaultPDNEnabled, 400
 - profileList, 400
- unpack_wds_SLQSGetCurrDataSystemStat
 - wds.h, 612
- unpack_wds_SLQSGetCurrDataSystemStat_t, 401
 - currNetworkInfo, 401
 - networkInfoLen, 401
 - prefNetwork, 401
- unpack_wds_SLQSGetDUNCallInfo
 - wds.h, 612
- unpack_wds_SLQSGetDUNCallInfo_t, 402
 - callEndReason, 402
 - channelRate, 402
 - connectionStatus, 402
 - dataBearerTech, 402
 - dormancyStatus, 402
 - lastCallDataBearerTech, 402
 - mdmCallDurationActive, 402
 - rxOKBytesCount, 403
 - txOKBytesCount, 403
- unpack_wds_SLQSGetDataBearerTechnology
 - wds.h, 612
- unpack_wds_SLQSGetDataBearerTechnology_t, 401
 - curDataBearerTechnology, 401
 - dataBearerMask, 401
 - lastCallDataBearerTechnology, 401
- unpack_wds_SLQSGetProfileSettings
 - wds.h, 613
- unpack_wds_SLQSGetProfileSettings_t, 403
 - pProfileSettings, 403
 - ProfileType, 403
 - Tlvresult, 403
- unpack_wds_SLQSGetRuntimeSettings
 - wds.h, 613
- unpack_wds_SLQSGetRuntimeSettings_t, 403
 - APNName, 404
 - Authentication, 404
 - DomainList, 404
 - GPRSGrantedQoS, 404
 - GWAddressV4, 404
 - IMCNflag, 404
 - IPFamilyPreference, 404
 - IPv6AddrInfo, 404
 - IPv6GWAddrInfo, 404
 - IPv4, 404
 - Mtu, 404
 - PDPTType, 405
 - PrimaryDNSV4, 405
 - PrimaryDNSV6, 405
 - ProfileID, 405
 - ProfileName, 405
 - SecondaryDNSV4, 405
 - SecondaryDNSV6, 405

- ServerAddrList, [405](#)
- SubnetMaskV4, [405](#)
- Technology, [405](#)
- UMTSGrantedQoS, [405](#)
- Username, [405](#)
- unpack_wds_SLQSModifyProfile
 - wds.h, [613](#)
- unpack_wds_SLQSModifyProfile_t, [405](#)
 - pExtErrorCode, [405](#)
- unpack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [615](#)
- unpack_wds_SLQSSetDHCPv4ClientConfig_t, [408](#)
 - pHwConfig, [409](#)
- unpack_wds_SLQSSet3GPPConfigItem
 - wds.h, [614](#)
- unpack_wds_SLQSSetIPFamilyPreference
 - wds.h, [614](#)
- unpack_wds_SLQSSetIPFamilyPreference_t, [405](#)
 - Tlvresult, [406](#)
- unpack_wds_SLQSSetPacketSrvStatusCallback
 - wds.h, [614](#)
- unpack_wds_SLQSSetPacketSrvStatusCallback_t, [406](#)
 - bearerID, [406](#)
 - conn_status, [406](#)
 - ipFamily, [406](#)
 - reconfigReqd, [406](#)
 - sessionEndReason, [406](#)
 - techName, [406](#)
 - verboseSessnEndReason, [407](#)
 - verboseSessnEndReasonType, [407](#)
- unpack_wds_SLQSSetWdsEventCallback
 - wds.h, [615](#)
- unpack_wds_SLQSSetWdsEventCallback_ind
 - wds.h, [615](#)
- unpack_wds_SLQSSetWdsEventCallback_ind_t, [407](#)
 - currDBTechAvail, [408](#)
 - currNWInfo, [408](#)
 - dBTechAvail, [408](#)
 - dBTechnology, [408](#)
 - dataSysStatAvail, [408](#)
 - dormancyStatAvail, [408](#)
 - dormancyStatus, [408](#)
 - mipStatus, [408](#)
 - mipstatAvail, [408](#)
 - netInfoLen, [408](#)
 - prefNetwork, [408](#)
 - ratMask, [408](#)
 - rx_bytes, [408](#)
 - rx_pkts, [408](#)
 - soMask, [408](#)
 - tx_bytes, [408](#)
 - tx_pkts, [408](#)
 - xferStatAvail, [408](#)
- unpack_wds_SLQSStartDataSession
 - wds.h, [615](#)
- unpack_wds_SLQSStartDataSession_t, [409](#)
 - pFailureReason, [409](#)
 - pVerboseFailReasonType, [409](#)
 - pVerboseFailureReason, [409](#)
 - psid, [409](#)
- unpack_wds_SLQSStopDataSession
 - wds.h, [617](#)
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, [617](#)
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [409](#)
 - apnName, [410](#)
 - bearerId, [410](#)
 - contextId, [410](#)
 - ipv4Address, [410](#)
 - ipv4GWAddress, [410](#)
 - ipv6Address, [410](#)
 - ipv6GWAddress, [411](#)
- unpack_wds_SetDefaultProfile
 - wds.h, [610](#)
- unpack_wds_SetDefaultProfileNum
 - wds.h, [610](#)
- unpack_wds_SetMobileIPProfile
 - wds.h, [610](#)
- unpack_wds_SetMobileIPProfile_t, [398](#)
- UnpackQmiProfileInfo
 - wds.h, [592](#)
- unpackWdsProfileParam, [411](#)
 - SlqsProfile3GPP, [411](#)
 - SlqsProfile3GPP2, [411](#)
- UpdateCompleteStatus
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [382](#)
- updateCompleteStatus
 - unpack_omaDmFotaTlv_t, [347](#)
- upinRetries
 - slotInf, [256](#)
 - uim_slotInfo, [278](#)
- upinState
 - slotInf, [256](#)
 - uim_slotInfo, [278](#)
- upukRetries
 - slotInf, [256](#)
 - uim_slotInfo, [278](#)
- usageMask
 - loc_sensorDataUsage, [79](#)
- userInputReq
 - unpack_omaDmConfigTlv_t, [345](#)
 - unpack_omaDmFotaTlv_t, [347](#)
- userInputTimeout
 - unpack_omaDmConfigTlv_t, [345](#)
 - unpack_omaDmFotaTlv_t, [347](#)
- Username
 - unpack_wds_SLQSGetRuntimeSettings_t, [405](#)
- username
 - unpack_wds_GetDefaultProfile_t, [394](#)
- usersize
 - unpack_wds_GetDefaultProfile_t, [394](#)
- VDOP
 - loc_precisionDilution, [78](#)
- val

- unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 617
- unpack_wds_SetDefaultProfile, 610
- unpack_wds_SetDefaultProfileNum, 610
- unpack_wds_SetMobileIPProfile, 610
- UnpackQmiProfileInfo, 592
- wds_Domain, 413
 - domainLen, 414
 - domainName, 414
- wds_DomainNameList, 414
 - domain, 414
 - numInstances, 414
- wds_GPRSQoS, 414
 - delayClass, 416
 - meanThroughputClass, 416
 - peakThroughputClass, 416
 - precedenceClass, 416
 - reliabilityClass, 416
- wds_IPV6AddressInfo, 416
 - IPAddressV6, 417
 - IPV6PrefixLen, 417
- wds_IPV6GWAddressInfo, 417
 - gwAddressV6, 417
 - gwV6PrefixLen, 417
- wds_PCSCFFQDNAddress, 417
 - fqdnAddr, 418
 - fqdnLen, 418
- wds_PCSCFFQDNAddressList, 418
 - numInstances, 418
 - pcsfFQDNAddress, 418
- wds_PCSCFIPv4ServerAddressList, 418
 - numInstances, 419
 - pcscfIPv4Addr, 419
- wds_ProfileIdentifier, 419
 - profileIndex, 419
 - profileType, 419
- wds_UMTSMInQoS, 420
 - deliveryErrSDU, 422
 - grntDownlinkBitrate, 422
 - grntUplinkBitrate, 422
 - maxDownlinkBitrate, 422
 - maxSDUSize, 422
 - maxUplinkBitrate, 422
 - qosDeliveryOrder, 422
 - resBerRatio, 422
 - sduErrorRatio, 422
 - trafficClass, 422
 - trafficPriority, 422
 - transferDelay, 422
- wds_currNetworkInfo, 411
 - NetworkType, 413
 - RATMask, 413
 - SOMask, 413
- wds_profileInfo, 419
 - SlqsProfile3GPP, 419
 - SlqsProfile3GPP2, 419
- wdsDhcpv4HwConfig, 423
 - chaddr, 423
 - chaddrLen, 423
 - hwType, 423
- wdsDhcpv4Option, 423
 - optCode, 424
 - optVal, 424
 - optValLen, 424
- wdsDhcpv4OptionList, 424
 - numOpt, 424
 - pOptList, 424
- wdsDhcpv4ProfileId, 424
 - profileId, 424
 - profileType, 425
- xferStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 408
- xid
 - pack_qmi_t, 214
 - unpack_qmi_t, 348