

LinuxQMISDK

Lite-0.7.1

Generated by Doxygen 1.8.6

Wed Aug 24 2016 10:20:15

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	17
5.1	File List	17
6	Module Documentation	19
6.1	QMI pack/unpack (pack)	19
6.1.1	Detailed Description	19
7	Namespace Documentation	21
7.1	Tables Namespace Reference	21
7.1.1	Detailed Description	21
8	Data Structure Documentation	23
8.1	appStats Struct Reference	23
8.1.1	Detailed Description	23
8.1.2	Field Documentation	25
8.1.2.1	aidLength	25
8.1.2.2	aidVal	25

8.1.2.3	appState	25
8.1.2.4	appType	25
8.1.2.5	persoFeature	25
8.1.2.6	persoRetries	25
8.1.2.7	persoState	25
8.1.2.8	persoUnblockRetries	26
8.1.2.9	pin1Retries	26
8.1.2.10	pin1State	26
8.1.2.11	pin2Retries	26
8.1.2.12	pin2State	26
8.1.2.13	puk1Retries	26
8.1.2.14	puk2Retries	26
8.1.2.15	univPin	26
8.2	CarrierImage_t Struct Reference	26
8.2.1	Detailed Description	26
8.2.2	Field Documentation	27
8.2.2.1	m_FwBuildId	27
8.2.2.2	m_FwImageld	27
8.2.2.3	m_nCarrierId	27
8.2.2.4	m_nFolderId	27
8.2.2.5	m_nStorage	27
8.2.2.6	m_PriBuildId	27
8.2.2.7	m_PrImageld	27
8.3	cdmaSSInfo Struct Reference	27
8.3.1	Detailed Description	27
8.3.2	Field Documentation	27
8.3.2.1	ecio	27
8.3.2.2	rssI	27
8.4	connectionStatus Struct Reference	27
8.4.1	Detailed Description	27
8.4.2	Field Documentation	28
8.4.2.1	MDMCallDuration	28
8.4.2.2	MDMConnStatus	28
8.5	currNetworkInfo Struct Reference	28
8.5.1	Detailed Description	28
8.5.2	Field Documentation	28
8.5.2.1	NetworkType	28
8.5.2.2	RATMask	28
8.5.2.3	SOMask	28
8.6	dms_ActivationStatusTlv Struct Reference	28

8.6.1	Detailed Description	28
8.6.2	Field Documentation	29
8.6.2.1	activationStatus	29
8.6.2.2	TlvPresent	29
8.7	dms_OperatingModeTlv Struct Reference	29
8.7.1	Detailed Description	29
8.7.2	Field Documentation	29
8.7.2.1	operatingMode	29
8.7.2.2	TlvPresent	30
8.8	DMScustSettingInfo Struct Reference	30
8.8.1	Detailed Description	30
8.8.2	Field Documentation	30
8.8.2.1	cust_attr	30
8.8.2.2	cust_id	30
8.8.2.3	cust_value	30
8.8.2.4	id_length	30
8.8.2.5	value_length	30
8.9	DMScustSettingList Struct Reference	30
8.9.1	Detailed Description	31
8.9.2	Field Documentation	31
8.9.2.1	custSetting	31
8.9.2.2	list_type	31
8.9.2.3	num_instances	31
8.10	DMSgetCustomFeatureV2 Struct Reference	31
8.10.1	Detailed Description	31
8.10.2	Field Documentation	32
8.10.2.1	pCustSettingInfo	32
8.10.2.2	pCustSettingList	32
8.10.2.3	pGetCustomInput	32
8.11	DMSgetCustomInput Struct Reference	32
8.11.1	Detailed Description	32
8.11.2	Field Documentation	32
8.11.2.1	cust_id	32
8.11.2.2	list_type	32
8.12	dunchannelRate Struct Reference	32
8.12.1	Detailed Description	32
8.12.2	Field Documentation	33
8.12.2.1	CurrChanRxRate	33
8.12.2.2	CurrChanTxRate	33
8.12.2.3	MaxChanRxRate	33

8.12.2.4	MaxChanTxRate	33
8.13	eTWSPLMNInfoTlv Struct Reference	33
8.13.1	Detailed Description	33
8.13.2	Field Documentation	33
8.13.2.1	ETWSPLMNInfo	33
8.13.2.2	TlvPresent	33
8.14	FMSImageElement Struct Reference	33
8.14.1	Detailed Description	33
8.14.2	Field Documentation	34
8.14.2.1	buildId	34
8.14.2.2	buildIdLength	34
8.14.2.3	imageId	34
8.14.2.4	imageType	34
8.15	FMSImageIdElement Struct Reference	34
8.15.1	Detailed Description	34
8.15.2	Field Documentation	35
8.15.2.1	buildID	35
8.15.2.2	buildIDLength	35
8.15.2.3	failureCount	35
8.15.2.4	imageID	35
8.15.2.5	storageIndex	35
8.16	FMSImageIDEntries Struct Reference	35
8.16.1	Detailed Description	35
8.16.2	Field Documentation	35
8.16.2.1	executingImage	35
8.16.2.2	imageIDElement	35
8.16.2.3	imageIDSize	36
8.16.2.4	imageType	36
8.16.2.5	maxImages	36
8.17	FMSImageList Struct Reference	36
8.17.1	Detailed Description	36
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	37
8.18.2.1	listEntries	37
8.18.2.2	listSize	37
8.19	hdrSSInfo Struct Reference	37

8.19.1 Detailed Description	37
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	38
8.21.1 Detailed Description	38
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	39
8.22.2.1 delayClass	39
8.22.2.2 meanThroughputClass	39
8.22.2.3 peakThroughputClass	39
8.22.2.4 precedenceClass	39
8.22.2.5 reliabilityClass	39
8.23 LibpackProfile3GPP Struct Reference	39
8.23.1 Detailed Description	40
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	43
8.23.2.5 pAPNnameSize	43
8.23.2.6 pAuthenticationPref	43
8.23.2.7 pGPRSMinimumQoS	43
8.23.2.8 pGPRSRequestedQos	43
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 pPDPTtype	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	44
8.23.2.33 pUMTSMinQoS	44
8.23.2.34 pUMTSMinQoSSigInd	44
8.23.2.35 pUMTSReqQoS	44
8.23.2.36 pUMTSReqQoSSigInd	44
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	49
8.24.2.1 pAllowLinger	49
8.24.2.2 pAPNClass3GPP2	49
8.24.2.3 pAPNEnabled3GPP2	49
8.24.2.4 pApnString	49
8.24.2.5 pApnStringSize	49
8.24.2.6 pAppPriority	49
8.24.2.7 pAppType	49
8.24.2.8 pAuthPassword	49
8.24.2.9 pAuthPasswordSize	49
8.24.2.10 pAuthProtocol	49
8.24.2.11 pAuthRetryCount	49

8.24.2.12 pAuthTimeout	49
8.24.2.13 pDataMode	49
8.24.2.14 pDataRate	49
8.24.2.15 plpcpAckTimeout	49
8.24.2.16 plpcpCreqRetryCount	49
8.24.2.17 plsPcscfAddressNedded	49
8.24.2.18 pLcpAckTimeout	49
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	50
8.24.2.24 pPppSessCloseTimerDO	50
8.24.2.25 pPrimaryV4DnsAddress	50
8.24.2.26 pPriV6DnsAddress	50
8.24.2.27 pRATType	50
8.24.2.28 pSecondaryV4DnsAddress	50
8.24.2.29 pSecV6DnsAddress	50
8.24.2.30 pUserId	50
8.24.2.31 pUserIdSize	50
8.25 LibPackprofile_3GPP Struct Reference	50
8.25.1 Detailed Description	51
8.25.2 Field Documentation	54
8.25.2.1 pAddrAllocPref	54
8.25.2.2 pAPNClass	54
8.25.2.3 pAPNDisabledFlag	54
8.25.2.4 pAPNName	54
8.25.2.5 pAPNnameSize	55
8.25.2.6 pAuthenticationPref	55
8.25.2.7 pGPRSMinimumQoS	55
8.25.2.8 pGPRSRequestedQos	55
8.25.2.9 plmCnFlag	55
8.25.2.10 pIPv4AddrPref	55
8.25.2.11 pIPv6AddPref	55
8.25.2.12 pPassword	55
8.25.2.13 pPasswordSize	55
8.25.2.14 pPcscfAddrUsingDhcp	55
8.25.2.15 pPcscfAddrUsingPCO	55
8.25.2.16 pPDNInactivTimeout	55
8.25.2.17 pPdpAccessConFlag	55

8.25.2.18 pPdpContext	55
8.25.2.19 pPdpDataCompType	55
8.25.2.20 pPdpHdrCompType	55
8.25.2.21 pPDPtype	55
8.25.2.22 pPriDNSIPv4AddPref	55
8.25.2.23 pPriDNSIPv6addpref	55
8.25.2.24 pPrimaryID	55
8.25.2.25 pProfilename	55
8.25.2.26 pProfilenameSize	55
8.25.2.27 pQosClassID	55
8.25.2.28 pSecDNSIPv4AddPref	55
8.25.2.29 pSecDNSIPv6addpref	55
8.25.2.30 pSecondaryFlag	55
8.25.2.31 pTFTID1Params	55
8.25.2.32 pTFTID2Params	55
8.25.2.33 pUMTSMinQoS	56
8.25.2.34 pUMTSMinQoSSigInd	56
8.25.2.35 pUMTSReqQoS	56
8.25.2.36 pUMTSReqQoSSigInd	56
8.25.2.37 pUsername	56
8.25.2.38 pUsernameSize	56
8.26 LibPackprofile_3GPP2 Struct Reference	56
8.26.1 Detailed Description	56
8.26.2 Field Documentation	60
8.26.2.1 pAllowLinger	60
8.26.2.2 pAPNClass3GPP2	60
8.26.2.3 pAPNEnabled3GPP2	60
8.26.2.4 pApnString	60
8.26.2.5 pApnStringSize	60
8.26.2.6 pAppPriority	60
8.26.2.7 pAppType	60
8.26.2.8 pAuthPassword	60
8.26.2.9 pAuthPassword_tSize	60
8.26.2.10 pAuthProtocol	60
8.26.2.11 pAuthRetryCount	60
8.26.2.12 pAuthTimeout	60
8.26.2.13 pDataMode	60
8.26.2.14 pDataRate	60
8.26.2.15 plpcpAckTimeout	60
8.26.2.16 plpcpCreqRetryCount	60

8.26.2.17	plsPcscfAddressNedded	60
8.26.2.18	pLcpAckTimeout	60
8.26.2.19	pLcpCreqRetryCount	61
8.26.2.20	pNegoDnsSrvrPref	61
8.26.2.21	pPDNInactivTimeout3GPP2	61
8.26.2.22	pPdnType	61
8.26.2.23	pPppSessCloseTimer1x	61
8.26.2.24	pPppSessCloseTimerDO	61
8.26.2.25	pPrimaryV4DnsAddress	61
8.26.2.26	pPriV6DnsAddress	61
8.26.2.27	pRATType	61
8.26.2.28	pSecondaryV4DnsAddress	61
8.26.2.29	pSecV6DnsAddress	61
8.26.2.30	pUserId	61
8.26.2.31	pUserIdSize	61
8.27	LibPackQosClassID Struct Reference	61
8.27.1	Detailed Description	61
8.27.2	Field Documentation	62
8.27.2.1	gDIBitRate	62
8.27.2.2	gUIBitRate	62
8.27.2.3	maxDIBitRate	62
8.27.2.4	maxUIBitRate	62
8.27.2.5	QCI	62
8.28	LibPackTFTIDParams Struct Reference	62
8.28.1	Detailed Description	62
8.28.2	Field Documentation	63
8.28.2.1	destPortRangeEnd	63
8.28.2.2	destPortRangeStart	63
8.28.2.3	eValid	63
8.28.2.4	filterId	63
8.28.2.5	flowLabel	63
8.28.2.6	IPSECSPi	63
8.28.2.7	ipVersion	64
8.28.2.8	nextHeader	64
8.28.2.9	pSourceIP	64
8.28.2.10	sourceIPMask	64
8.28.2.11	srcPortRangeEnd	64
8.28.2.12	srcPortRangeStart	64
8.28.2.13	tosMask	64
8.29	LibPackUMTSQoS Struct Reference	64

8.29.1 Detailed Description	64
8.29.2 Field Documentation	66
8.29.2.1 deliveryErrSDU	66
8.29.2.2 grntDownlinkBitrate	66
8.29.2.3 grntUplinkBitrate	66
8.29.2.4 maxDownlinkBitrate	66
8.29.2.5 maxSDUSize	66
8.29.2.6 maxUplinkBitrate	66
8.29.2.7 qosDeliveryOrder	66
8.29.2.8 resBerRatio	66
8.29.2.9 sduErrorRatio	66
8.29.2.10 trafficClass	66
8.29.2.11 trafficPriority	66
8.29.2.12 transferDelay	66
8.30 LibPackUMTSReqQoSsigInd Struct Reference	66
8.30.1 Detailed Description	66
8.30.2 Field Documentation	67
8.30.2.1 SigInd	67
8.30.2.2 UMTSReqQoS	67
8.31 loc_BdsSV Struct Reference	67
8.31.1 Detailed Description	67
8.31.2 Field Documentation	67
8.31.2.1 id	67
8.31.2.2 mask	67
8.32 loc_BdsSVInfo Struct Reference	67
8.32.1 Detailed Description	68
8.32.2 Field Documentation	68
8.32.2.1 len	68
8.32.2.2 pSV	68
8.33 loc_CellDb Struct Reference	68
8.33.1 Detailed Description	68
8.33.2 Field Documentation	68
8.33.2.1 mask	68
8.34 loc_ClkInfo Struct Reference	69
8.34.1 Detailed Description	69
8.34.2 Field Documentation	70
8.34.2.1 mask	70
8.35 loc_GnssData Struct Reference	70
8.35.1 Detailed Description	70
8.35.2 Field Documentation	71

8.35.2.1	mask	71
8.36	loc_gpsTime Struct Reference	71
8.36.1	Detailed Description	72
8.36.2	Field Documentation	72
8.36.2.1	gpsTimeOfWeekMs	72
8.36.2.2	gpsWeek	72
8.37	loc_LocApplicationInfo Struct Reference	72
8.37.1	Detailed Description	72
8.37.2	Field Documentation	73
8.37.2.1	appNameLength	73
8.37.2.2	appProviderLength	73
8.37.2.3	appVersionLength	73
8.37.2.4	appVersionValid	73
8.37.2.5	pAppName	73
8.37.2.6	pAppProvider	73
8.37.2.7	pAppVersion	73
8.38	loc_precisionDilution Struct Reference	73
8.38.1	Detailed Description	73
8.38.2	Field Documentation	74
8.38.2.1	HDOP	74
8.38.2.2	PDOP	74
8.38.2.3	VDOP	74
8.39	loc_sensorDataUsage Struct Reference	74
8.39.1	Detailed Description	74
8.39.2	Field Documentation	74
8.39.2.1	aidingIndicatorMask	75
8.39.2.2	usageMask	75
8.40	loc_SV Struct Reference	75
8.40.1	Detailed Description	75
8.40.2	Field Documentation	75
8.40.2.1	id	75
8.40.2.2	mask	75
8.40.2.3	system	75
8.41	loc_SVInfo Struct Reference	75
8.41.1	Detailed Description	76
8.41.2	Field Documentation	76
8.41.2.1	len	76
8.41.2.2	pSV	76
8.42	loc_svUsedforFix Struct Reference	76
8.42.1	Detailed Description	76

8.42.2	Field Documentation	77
8.42.2.1	gnssSvUsedList	77
8.42.2.2	gnssSvUsedList_len	77
8.43	IteSSInfo Struct Reference	77
8.43.1	Detailed Description	77
8.43.2	Field Documentation	77
8.43.2.1	rsrp	77
8.43.2.2	rsrq	77
8.43.2.3	rssr	77
8.43.2.4	snr	77
8.44	messageModeTlv Struct Reference	77
8.44.1	Detailed Description	77
8.44.2	Field Documentation	78
8.44.2.1	MessageModelInfo	78
8.44.2.2	TlvPresent	78
8.45	nas_acqOrderPref Struct Reference	78
8.45.1	Detailed Description	78
8.45.2	Field Documentation	78
8.45.2.1	acqOrdeLen	78
8.45.2.2	pAcqOrder	78
8.46	nas_AddCDMASysInfo Struct Reference	78
8.46.1	Detailed Description	78
8.46.2	Field Documentation	79
8.46.2.1	geoSysIdx	79
8.46.2.2	regPrd	79
8.47	nas_AddSysInfo Struct Reference	79
8.47.1	Detailed Description	79
8.47.2	Field Documentation	79
8.47.2.1	cellBroadcastCap	79
8.47.2.2	geoSysIdx	79
8.48	nas_CallBarringSysInfo Struct Reference	79
8.48.1	Detailed Description	80
8.48.2	Field Documentation	80
8.48.2.1	csBarStatus	80
8.48.2.2	psBarStatus	80
8.49	nas_callBarStatus Struct Reference	80
8.49.1	Detailed Description	80
8.49.2	Field Documentation	81
8.49.2.1	csBarStatus	81
8.49.2.2	psBarStatus	81

8.50 nas_CDMAECIOThresh Struct Reference	81
8.50.1 Detailed Description	81
8.50.2 Field Documentation	81
8.50.2.1 CDMAECIOThreshListLen	81
8.50.2.2 pCDMAECIOThreshList	82
8.51 nas_CDMAInfo Struct Reference	82
8.51.1 Detailed Description	82
8.51.2 Field Documentation	82
8.51.2.1 baseId	82
8.51.2.2 baseLat	82
8.51.2.3 baseLong	82
8.51.2.4 nid	83
8.51.2.5 refpn	83
8.51.2.6 sid	83
8.52 nas_CDMARSSIThresh Struct Reference	83
8.52.1 Detailed Description	83
8.52.2 Field Documentation	83
8.52.2.1 CDMARSSIThreshListLen	83
8.52.2.2 pCDMARSSIThreshList	83
8.53 nas_CDMASysInfo Struct Reference	83
8.53.1 Detailed Description	84
8.53.2 Field Documentation	86
8.53.2.1 baseId	86
8.53.2.2 baseLat	86
8.53.2.3 baseLong	86
8.53.2.4 bsInfoValid	86
8.53.2.5 bsPRev	86
8.53.2.6 bsPRevValid	86
8.53.2.7 ccsSupported	86
8.53.2.8 ccsSupportedValid	86
8.53.2.9 cdmaSysIdValid	86
8.53.2.10 isSysPrIMatch	86
8.53.2.11 isSysPrIMatchValid	86
8.53.2.12 MCC	86
8.53.2.13 MNC	86
8.53.2.14 networkId	86
8.53.2.15 networkIdValid	87
8.53.2.16 packetZone	87
8.53.2.17 packetZoneValid	87
8.53.2.18 pRevInUse	87

8.53.2.19 pRevInUseValid	87
8.53.2.20 sysInfoCDMA	87
8.53.2.21 systemID	87
8.54 nas_CDMASysInfoExt Struct Reference	87
8.54.1 Detailed Description	87
8.54.2 Field Documentation	87
8.54.2.1 imsi_11_12	87
8.54.2.2 MCC	87
8.55 nas_cellParams Struct Reference	87
8.55.1 Detailed Description	88
8.55.2 Field Documentation	88
8.55.2.1 pci	88
8.55.2.2 rsrp	88
8.55.2.3 rsrq	88
8.55.2.4 rssi	88
8.55.2.5 srxlev	88
8.56 nas_CommInfo Struct Reference	88
8.56.1 Detailed Description	88
8.56.2 Field Documentation	90
8.56.2.1 imsRegState	90
8.56.2.2 modemMode	90
8.56.2.3 psState	90
8.56.2.4 systemMode	90
8.56.2.5 temperature	90
8.57 nas_CSGID Struct Reference	90
8.57.1 Detailed Description	90
8.57.2 Field Documentation	90
8.57.2.1 id	91
8.57.2.2 mcc	91
8.57.2.3 mnc	91
8.57.2.4 mncPcsDigits	91
8.57.2.5 rat	91
8.58 nas_currentPLMN Struct Reference	91
8.58.1 Detailed Description	91
8.58.2 Field Documentation	91
8.58.2.1 MCC	91
8.58.2.2 MNC	91
8.58.2.3 netDescr	92
8.58.2.4 netDescrLength	92
8.59 nas_dataSrvCapabilities Struct Reference	92

8.59.1 Detailed Description	92
8.59.2 Field Documentation	92
8.59.2.1 dataCapabilities	92
8.59.2.2 dataCapabilitiesLen	92
8.60 nas_detailSvcInfo Struct Reference	92
8.60.1 Detailed Description	93
8.60.2 Field Documentation	94
8.60.2.1 hdrHybrid	94
8.60.2.2 hdrSrvStatus	94
8.60.2.3 isSysForbidden	94
8.60.2.4 srvCapability	94
8.60.2.5 srvStatus	94
8.61 nas_ecioListElement Struct Reference	94
8.61.1 Detailed Description	94
8.61.2 Field Documentation	94
8.61.2.1 ecio	94
8.61.2.2 radiolf	94
8.62 nas_errorRateListElement Struct Reference	95
8.62.1 Detailed Description	95
8.62.2 Field Documentation	95
8.62.2.1 errorRate	95
8.62.2.2 radiolf	95
8.63 nas_GERANInfo Struct Reference	95
8.63.1 Detailed Description	96
8.63.2 Field Documentation	97
8.63.2.1 arfcn	97
8.63.2.2 bsic	97
8.63.2.3 cellID	97
8.63.2.4 insNmrCellInfo	97
8.63.2.5 lac	97
8.63.2.6 nmrlnst	97
8.63.2.7 plmn	97
8.63.2.8 rxLev	97
8.63.2.9 timingAdvance	97
8.64 nas_geranInstInfo Struct Reference	97
8.64.1 Detailed Description	98
8.64.2 Field Documentation	98
8.64.2.1 geranArfcn	98
8.64.2.2 geranBsicBcc	98
8.64.2.3 geranBsicNcc	98

8.64.2.4	geranRssi	98
8.65	nas_gsmCellInfo Struct Reference	98
8.65.1	Detailed Description	98
8.65.2	Field Documentation	99
8.65.2.1	arfcn	99
8.65.2.2	band1900	99
8.65.2.3	bsicld	99
8.65.2.4	cellIdValid	99
8.65.2.5	rssI	99
8.65.2.6	srxlev	99
8.66	nas_GSMRSSIthresh Struct Reference	99
8.66.1	Detailed Description	99
8.66.2	Field Documentation	100
8.66.2.1	GSMRSSIthreshListLen	100
8.66.2.2	pGSMRSSIthreshList	100
8.67	nas_GSMSrvStatusInfo Struct Reference	100
8.67.1	Detailed Description	100
8.67.2	Field Documentation	101
8.67.2.1	isPrefDataPath	101
8.67.2.2	srvStatus	101
8.67.2.3	trueSrvStatus	101
8.68	nas_GSMSysInfo Struct Reference	101
8.68.1	Detailed Description	101
8.68.2	Field Documentation	103
8.68.2.1	cellId	103
8.68.2.2	cellIdValid	103
8.68.2.3	dtmSupp	103
8.68.2.4	dtmSuppValid	103
8.68.2.5	egprsSupp	103
8.68.2.6	egprsSuppValid	103
8.68.2.7	lac	103
8.68.2.8	lacValid	103
8.68.2.9	MCC	103
8.68.2.10	MNC	103
8.68.2.11	networkIdValid	103
8.68.2.12	regRejectInfoValid	103
8.68.2.13	rejCause	104
8.68.2.14	rejectSrvDomain	104
8.68.2.15	sysInfoGSM	104
8.69	nas_HDRECIOthresh Struct Reference	104

8.69.1 Detailed Description	104
8.69.2 Field Documentation	104
8.69.2.1 HDRECIOThreshListLen	104
8.69.2.2 pHRECIOThreshList	104
8.70 nas_HDRIOThresh Struct Reference	104
8.70.1 Detailed Description	104
8.70.2 Field Documentation	105
8.70.2.1 HDRIOThreshListLen	105
8.70.2.2 pHDRIOThreshList	105
8.71 nas_HDRRSSIThresh Struct Reference	105
8.71.1 Detailed Description	105
8.71.2 Field Documentation	105
8.71.2.1 HDRRSSIThreshListLen	105
8.71.2.2 pHDRRSSIThreshList	105
8.72 nas_HDRSINRThreshold Struct Reference	105
8.72.1 Detailed Description	106
8.72.2 Field Documentation	106
8.72.2.1 HDRSINRThreshListLen	106
8.72.2.2 pHDRSINRThreshList	106
8.73 nas_HDRSysInfo Struct Reference	106
8.73.1 Detailed Description	106
8.73.2 Field Documentation	108
8.73.2.1 hdrActiveProt	108
8.73.2.2 hdrActiveProtValid	108
8.73.2.3 hdrPersonality	108
8.73.2.4 hdrPersonalityValid	108
8.73.2.5 is856SysId	108
8.73.2.6 is856SysIdValid	108
8.73.2.7 isSysPrIMatch	108
8.73.2.8 isSysPrIMatchValid	108
8.73.2.9 sysInfoHDR	108
8.74 nas_infoInterFreq Struct Reference	108
8.74.1 Detailed Description	108
8.74.2 Field Documentation	109
8.74.2.1 cell_resel_priority	109
8.74.2.2 cellInterFreqParams	109
8.74.2.3 cells_len	109
8.74.2.4 earfcn	109
8.74.2.5 threshXHigh	109
8.74.2.6 threshXLow	109

8.75	nas_lteGsmCellInfo Struct Reference	109
8.75.1	Detailed Description	109
8.75.2	Field Documentation	110
8.75.2.1	cellReselPriority	110
8.75.2.2	cells_len	110
8.75.2.3	GsmCellInfo	110
8.75.2.4	nccPermitted	110
8.75.2.5	threshGsmHigh	110
8.75.2.6	threshGsmLow	110
8.76	nas_LTEInfo Struct Reference	110
8.76.1	Detailed Description	111
8.76.2	Field Documentation	112
8.76.2.1	band	112
8.76.2.2	bandwidth	112
8.76.2.3	emmConnState	112
8.76.2.4	emmState	112
8.76.2.5	emmSubState	112
8.76.2.6	RXChan	112
8.76.2.7	TXChan	112
8.77	nas_LTEInfoInterfreq Struct Reference	113
8.77.1	Detailed Description	113
8.77.2	Field Documentation	113
8.77.2.1	freqsLen	113
8.77.2.2	InfoInterfreq	113
8.77.2.3	ueInIdle	113
8.78	nas_LTEInfoIntrafreq Struct Reference	113
8.78.1	Detailed Description	114
8.78.2	Field Documentation	115
8.78.2.1	CellParams	115
8.78.2.2	cellReselPriority	115
8.78.2.3	cellsLen	115
8.78.2.4	earfcn	115
8.78.2.5	globalCellId	115
8.78.2.6	plmn	115
8.78.2.7	servingCellId	115
8.78.2.8	sIntraSearch	115
8.78.2.9	sNonIntraSearch	115
8.78.2.10	tac	115
8.78.2.11	threshServingLow	115
8.78.2.12	ueInIdle	115

8.79	nas_LTEInfoNeighboringGSM Struct Reference	115
8.79.1	Detailed Description	116
8.79.2	Field Documentation	116
8.79.2.1	freqsLen	116
8.79.2.2	LteGsmCellInfo	116
8.79.2.3	ueInIdle	116
8.80	nas_LTEInfoNeighboringWCDMA Struct Reference	116
8.80.1	Detailed Description	116
8.80.2	Field Documentation	117
8.80.2.1	freqsLen	117
8.80.2.2	LTEWCDMACellInfo	117
8.80.2.3	ueInIdle	117
8.81	nas_lteRsrpInformation Struct Reference	117
8.81.1	Detailed Description	117
8.81.2	Field Documentation	117
8.81.2.1	rsrplevel	117
8.82	nas_LTERSRPThresh Struct Reference	117
8.82.1	Detailed Description	117
8.82.2	Field Documentation	117
8.82.2.1	LTERSRPThreshListLen	118
8.82.2.2	pLTERSRPThreshList	118
8.83	nas_LTERSRQThresh Struct Reference	118
8.83.1	Detailed Description	118
8.83.2	Field Documentation	118
8.83.2.1	LTERSRQThreshListLen	118
8.83.2.2	pLTERSRQThreshList	118
8.84	nas_LTERSSIThresh Struct Reference	118
8.84.1	Detailed Description	118
8.84.2	Field Documentation	119
8.84.2.1	LTERSSIThreshListLen	119
8.84.2.2	pLTERSSIThreshList	119
8.85	nas_LTESigRptConfig Struct Reference	119
8.85.1	Detailed Description	119
8.85.2	Field Documentation	119
8.85.2.1	avgPeriod	119
8.85.2.2	rptRate	119
8.86	nas_lteSnrInformation Struct Reference	119
8.86.1	Detailed Description	120
8.86.2	Field Documentation	120
8.86.2.1	snrlevel	120

8.87 nas_LTESNRThreshold Struct Reference	120
8.87.1 Detailed Description	120
8.87.2 Field Documentation	120
8.87.2.1 LTESNRThreshListLen	120
8.87.2.2 pLTESNRThreshList	120
8.88 nas_LTESysInfo Struct Reference	120
8.88.1 Detailed Description	121
8.88.2 Field Documentation	122
8.88.2.1 cellId	122
8.88.2.2 cellIdValid	122
8.88.2.3 lac	122
8.88.2.4 lacValid	122
8.88.2.5 MCC	123
8.88.2.6 MNC	123
8.88.2.7 networkIdValid	123
8.88.2.8 regRejectInfoValid	123
8.88.2.9 rejCause	123
8.88.2.10 rejectSrvDomain	123
8.88.2.11 sysInfoLTE	123
8.88.2.12 tac	123
8.88.2.13 tacValid	123
8.89 nas_lteWcdmaCellInfo Struct Reference	123
8.89.1 Detailed Description	123
8.89.2 Field Documentation	124
8.89.2.1 cellReselPriority	124
8.89.2.2 cellsLen	124
8.89.2.3 threshXhigh	124
8.89.2.4 threshXlow	124
8.89.2.5 uarfcn	124
8.89.2.6 WCDMACellInfo	124
8.90 nas_MNRInfo Struct Reference	124
8.90.1 Detailed Description	124
8.90.2 Field Documentation	124
8.90.2.1 mcc	125
8.90.2.2 mnc	125
8.90.2.3 rat	125
8.91 nas_netSelectionPref Struct Reference	125
8.91.1 Detailed Description	125
8.91.2 Field Documentation	125
8.91.2.1 mcc	125

8.91.2.2	mnc	125
8.91.2.3	netReg	125
8.92	nas_nmrCellInfo Struct Reference	125
8.92.1	Detailed Description	126
8.92.2	Field Documentation	126
8.92.2.1	nmrArfcn	126
8.92.2.2	nmrBsic	126
8.92.2.3	nmrCellID	126
8.92.2.4	nmrLac	126
8.92.2.5	nmrPlmn	126
8.92.2.6	nmrRxLev	126
8.93	nas_PhyCaAggPcellInfo Struct Reference	127
8.93.1	Detailed Description	127
8.93.2	Field Documentation	127
8.93.2.1	dl_bw_value	127
8.93.2.2	freq	127
8.93.2.3	iLTEbandValue	127
8.93.2.4	pci	127
8.93.2.5	TlvPresent	127
8.94	nas_PhyCaAggScellDIBw Struct Reference	127
8.94.1	Detailed Description	128
8.94.2	Field Documentation	128
8.94.2.1	dl_bw_value	128
8.94.2.2	TlvPresent	128
8.95	nas_PhyCaAggScellIndex Struct Reference	128
8.95.1	Detailed Description	128
8.95.2	Field Documentation	128
8.95.2.1	scell_idx	128
8.95.2.2	TlvPresent	128
8.96	nas_PhyCaAggScellIndType Struct Reference	128
8.96.1	Detailed Description	129
8.96.2	Field Documentation	129
8.96.2.1	freq	129
8.96.2.2	pci	129
8.96.2.3	scell_state	129
8.96.2.4	TlvPresent	129
8.97	nas_PhyCaAggScellInfo Struct Reference	129
8.97.1	Detailed Description	129
8.97.2	Field Documentation	132
8.97.2.1	dl_bw_value	132

8.97.2.2	freq	132
8.97.2.3	iLTEbandValue	132
8.97.2.4	pci	132
8.97.2.5	scell_state	132
8.97.2.6	TlvPresent	132
8.98	nas_qaQmi3Gpp2TimeZone Struct Reference	132
8.98.1	Detailed Description	132
8.98.2	Field Documentation	132
8.98.2.1	daylightSavings	132
8.98.2.2	leapSeconds	132
8.98.2.3	localTimeOffset	133
8.99	nas_QmiNas3GppNetworkInfo Struct Reference	133
8.99.1	Detailed Description	133
8.99.2	Field Documentation	133
8.99.2.1	Desription	133
8.99.2.2	Forbidden	133
8.99.2.3	InUse	133
8.99.2.4	MCC	133
8.99.2.5	MNC	133
8.99.2.6	Preferred	133
8.99.2.7	Roaming	133
8.100	nas_QmiNas3GppNetworkRAT Struct Reference	133
8.100.1	Detailed Description	133
8.100.2	Field Documentation	134
8.100.2.1	MCC	134
8.100.2.2	MNC	134
8.100.2.3	RAT	134
8.101	nas_QmisNasPcsDigit Struct Reference	134
8.101.1	Detailed Description	134
8.101.2	Field Documentation	134
8.101.2.1	includes_pcs_digit	135
8.101.2.2	MCC	135
8.101.2.3	MNC	135
8.102	nas_RejectReasonTlv Struct Reference	135
8.102.1	Detailed Description	135
8.102.2	Field Documentation	135
8.102.2.1	rejectCause	135
8.102.2.2	serviceDomain	135
8.102.2.3	TlvPresent	135
8.103	nas_RFInfoTlv Struct Reference	135

8.103.1 Detailed Description	135
8.103.2 Field Documentation	135
8.103.2.1 activeBandClass	135
8.103.2.2 activeChannel	136
8.103.2.3 radiolInterface	136
8.103.2.4 radiolInterfaceSize	136
8.103.2.5 TlvPresent	136
8.104nas_roamIndList Struct Reference	136
8.104.1 Detailed Description	136
8.104.2 Field Documentation	136
8.104.2.1 numInstances	136
8.104.2.2 radiolInterface	136
8.104.2.3 roamIndicator	137
8.105nas_rsrqInformation Struct Reference	137
8.105.1 Detailed Description	137
8.105.2 Field Documentation	137
8.105.2.1 radiolf	137
8.105.2.2 rsrq	137
8.106nas_rxSignalStrengthListElement Struct Reference	137
8.106.1 Detailed Description	137
8.106.2 Field Documentation	138
8.106.2.1 radiolf	138
8.106.2.2 rxSignalStrength	138
8.107nas_servSystem Struct Reference	138
8.107.1 Detailed Description	138
8.107.2 Field Documentation	139
8.107.2.1 csAttachState	139
8.107.2.2 numRadiolInterfaces	139
8.107.2.3 psAttachState	139
8.107.2.4 radiolInterface	139
8.107.2.5 regState	139
8.107.2.6 selNetwork	139
8.108nas_SignalStrengthTlv Struct Reference	140
8.108.1 Detailed Description	140
8.108.2 Field Documentation	140
8.108.2.1 radiolInterface	140
8.108.2.2 signalStrength	140
8.108.2.3 TlvPresent	140
8.109nas_SLQSSignalStrengthsIndReq Struct Reference	140
8.109.1 Detailed Description	140

8.109.2 Field Documentation	141
8.109.2.1 ecioDelta	141
8.109.2.2 ecioThresholdList	141
8.109.2.3 ecioThresholdListLen	141
8.109.2.4 ioDelta	141
8.109.2.5 lteRsrpDelta	141
8.109.2.6 lteSnrDelta	141
8.109.2.7 rsrqDelta	141
8.109.2.8 rxSignalStrengthDelta	141
8.109.2.9 sinrDelta	141
8.109.2.10 sinrThresholdList	141
8.109.2.11 sinrThresholdListLen	141
8.110 nas_SLQSSignalStrengthsInformation Struct Reference	141
8.110.1 Detailed Description	141
8.110.2 Field Documentation	142
8.110.2.1 ecioInfo	142
8.110.2.2 errorRateInfo	142
8.110.2.3 io	142
8.110.2.4 lteRsrpinfo	142
8.110.2.5 lteSnrinfo	142
8.110.2.6 rsrqInfo	142
8.110.2.7 rxSignalStrengthInfo	142
8.110.2.8 sinr	142
8.111 nas_SLQSSignalStrengthsTlv Struct Reference	142
8.111.1 Detailed Description	142
8.111.2 Field Documentation	142
8.111.2.1 sSLQSSignalStrengthsInfo	142
8.111.2.2 TlvPresent	142
8.112 nas_SrvStatusInfo Struct Reference	142
8.112.1 Detailed Description	143
8.112.2 Field Documentation	143
8.112.2.1 isPrefDataPath	143
8.112.2.2 srvStatus	143
8.113 nas_sysInfoCommon Struct Reference	143
8.113.1 Detailed Description	143
8.113.2 Field Documentation	145
8.113.2.1 isSysForbidden	145
8.113.2.2 isSysForbiddenValid	145
8.113.2.3 roamStatus	145
8.113.2.4 roamStatusValid	145

8.113.2.5	srvCapability	145
8.113.2.6	srvCapabilityValid	145
8.113.2.7	srvDomain	145
8.113.2.8	srvDomainValid	145
8.114	nas_TDSCDMAECIOThresh Struct Reference	146
8.114.1	Detailed Description	146
8.114.2	Field Documentation	146
8.114.2.1	pTDSCDMAECIOThreshList	146
8.114.2.2	TDSCDMAECIOThreshListLen	146
8.115	nas_TDSCDMARSCPThresh Struct Reference	146
8.115.1	Detailed Description	146
8.115.2	Field Documentation	146
8.115.2.1	pTDSCDMARSCPThreshList	146
8.115.2.2	TDSCDMARSCPThreshListLen	147
8.116	nas_TDSCDMARSSIThresh Struct Reference	147
8.116.1	Detailed Description	147
8.116.2	Field Documentation	147
8.116.2.1	pTDSCDMARSSIThreshList	147
8.116.2.2	TDSCDMARSSIThreshListLen	147
8.117	nas_TDSCDMASINRThresh Struct Reference	147
8.117.1	Detailed Description	147
8.117.2	Field Documentation	147
8.117.2.1	pTDSCDMASINRThreshList	148
8.117.2.2	TDSCDMASINRThreshListLen	148
8.118	nas_timeInfo Struct Reference	148
8.118.1	Detailed Description	148
8.118.2	Field Documentation	149
8.118.2.1	day	149
8.118.2.2	dayLtSavingAdj	149
8.118.2.3	dayOfWeek	149
8.118.2.4	hour	149
8.118.2.5	minute	149
8.118.2.6	month	149
8.118.2.7	radioInterface	149
8.118.2.8	second	149
8.118.2.9	timeZone	149
8.118.2.10	tlvPresent	149
8.118.2.11	year	149
8.119	nas_UMTSInfo Struct Reference	149
8.119.1	Detailed Description	150

8.119.2 Field Documentation	151
8.119.2.1 cellID	151
8.119.2.2 ecio	151
8.119.2.3 geranInst	151
8.119.2.4 GeranInstInfo	151
8.119.2.5 lac	151
8.119.2.6 plmn	151
8.119.2.7 psc	151
8.119.2.8 rscp	151
8.119.2.9 uarfcn	151
8.119.2.10 umtsInst	151
8.119.2.11 UMTSInstInfo	151
8.120 nas_UMTSinstInfo Struct Reference	151
8.120.1 Detailed Description	151
8.120.2 Field Documentation	152
8.120.2.1 umtsEcio	152
8.120.2.2 umtsPsc	152
8.120.2.3 umtsRscp	152
8.120.2.4 umtsUarfcn	152
8.121 nas_umtsLTENbrCell Struct Reference	152
8.121.1 Detailed Description	152
8.121.2 Field Documentation	153
8.121.2.1 cellIsTDD	153
8.121.2.2 earfcn	153
8.121.2.3 pci	153
8.121.2.4 rsrp	153
8.121.2.5 rsrq	153
8.121.2.6 srxlev	153
8.122 nas_UniversalTime Struct Reference	153
8.122.1 Detailed Description	153
8.122.2 Field Documentation	154
8.122.2.1 day	154
8.122.2.2 dayOfWeek	154
8.122.2.3 hour	154
8.122.2.4 minute	154
8.122.2.5 month	154
8.122.2.6 second	154
8.122.2.7 year	154
8.123 nas_wcdmaCellInfo Struct Reference	154
8.123.1 Detailed Description	154

8.123.2 Field Documentation	155
8.123.2.1 cpich_ecno	155
8.123.2.2 cpich_rscp	155
8.123.2.3 psc	155
8.123.2.4 srxlev	155
8.124nas_WCDMAECIOThresh Struct Reference	155
8.124.1 Detailed Description	155
8.124.2 Field Documentation	155
8.124.2.1 pWCDMAECIOThreshList	155
8.124.2.2 WCDMAECIOThreshListLen	156
8.125nas_WCDMAInfoLTENeighborCell Struct Reference	156
8.125.1 Detailed Description	156
8.125.2 Field Documentation	156
8.125.2.1 UMTSLTENbrCell	156
8.125.2.2 umtsLTENbrCellLen	156
8.125.2.3 wcdmaRRCState	156
8.126nas_WCDMARSSIThresh Struct Reference	156
8.126.1 Detailed Description	157
8.126.2 Field Documentation	157
8.126.2.1 pWCDMARSSIThreshList	157
8.126.2.2 WCDMARSSIThreshListLen	157
8.127nas_WCDMASysInfo Struct Reference	157
8.127.1 Detailed Description	157
8.127.2 Field Documentation	160
8.127.2.1 cellId	160
8.127.2.2 cellIdValid	160
8.127.2.3 hsCallStatus	160
8.127.2.4 hsCallStatusValid	160
8.127.2.5 hsInd	160
8.127.2.6 hsIndValid	160
8.127.2.7 lac	160
8.127.2.8 lacValid	160
8.127.2.9 MCC	160
8.127.2.10MNC	160
8.127.2.11networkIdValid	160
8.127.2.12psc	160
8.127.2.13pscValid	160
8.127.2.14regRejectInfoValid	160
8.127.2.15rejCause	160
8.127.2.16rejectSrvDomain	160

8.127.2.17sysInfoWCDMA	160
8.128NASBandPreferenceTlv Struct Reference	160
8.128.1 Field Documentation	160
8.128.1.1 band_pref	160
8.128.1.2 TlvPresent	161
8.129NASEmergencyModeTlv Struct Reference	161
8.129.1 Field Documentation	161
8.129.1.1 EmerMode	161
8.129.1.2 TlvPresent	161
8.130NasGetLTECphyCaInfo Struct Reference	161
8.130.1 Field Documentation	161
8.130.1.1 PhyCaAggPcellInfo	161
8.130.1.2 PhyCaAggScellDIBw	161
8.130.1.3 PhyCaAggScellIndex	161
8.130.1.4 PhyCaAggScellIndType	161
8.130.1.5 PhyCaAggScellInfo	161
8.131NASGWAcqOrderPrefTlv Struct Reference	161
8.131.1 Field Documentation	161
8.131.1.1 GWAcqOrderPref	161
8.131.1.2 TlvPresent	161
8.132NASLTEBandPreferenceTlv Struct Reference	162
8.132.1 Field Documentation	162
8.132.1.1 LTEBandPref	162
8.132.1.2 TlvPresent	162
8.133NASLteNasReleaseInfoTlv Struct Reference	162
8.133.1 Field Documentation	162
8.133.1.1 nas_major	162
8.133.1.2 nas_minor	162
8.133.1.3 nas_release	162
8.133.1.4 TlvPresent	162
8.134NASModePreferenceTlv Struct Reference	162
8.134.1 Field Documentation	162
8.134.1.1 ModePref	162
8.134.1.2 TlvPresent	162
8.135NASNetSelPreferenceTlv Struct Reference	162
8.135.1 Field Documentation	163
8.135.1.1 NetSelPref	163
8.135.1.2 TlvPresent	163
8.136NASOTAMessageTlv Struct Reference	163
8.136.1 Field Documentation	163

8.136.1.1 data_buf	163
8.136.1.2 data_len	163
8.136.1.3 message_type	163
8.136.1.4 TlvPresent	163
8.137NASPhyCaAggPcellInfo Struct Reference	163
8.137.1 Detailed Description	163
8.137.2 Field Documentation	164
8.137.2.1 dl_bw_value	164
8.137.2.2 freq	164
8.137.2.3 iLTEbandValue	164
8.137.2.4 pci	164
8.137.2.5 TlvPresent	164
8.138NASPhyCaAggScellDIBw Struct Reference	164
8.138.1 Detailed Description	164
8.138.2 Field Documentation	164
8.138.2.1 dl_bw_value	164
8.138.2.2 TlvPresent	164
8.139NASPhyCaAggScellIndex Struct Reference	164
8.139.1 Detailed Description	165
8.139.2 Field Documentation	165
8.139.2.1 scell_idx	165
8.139.2.2 TlvPresent	165
8.140NASPhyCaAggScellIndType Struct Reference	165
8.140.1 Detailed Description	165
8.140.2 Field Documentation	165
8.140.2.1 freq	166
8.140.2.2 pci	166
8.140.2.3 scell_state	166
8.140.2.4 TlvPresent	166
8.141NASPhyCaAggScellInfo Struct Reference	166
8.141.1 Detailed Description	166
8.141.2 Field Documentation	166
8.141.2.1 dl_bw_value	166
8.141.2.2 freq	167
8.141.2.3 iLTEbandValue	167
8.141.2.4 pci	167
8.141.2.5 scell_state	167
8.141.2.6 TlvPresent	167
8.142NASPRLPreferenceTlv Struct Reference	167
8.142.1 Field Documentation	167

8.142.1.1 PRLPref	167
8.142.1.2 TlvPresent	167
8.143NASQmiCbkNasSwtOTAMessageInd Struct Reference	167
8.143.1 Field Documentation	167
8.143.1.1 nasRelInfoTlv	167
8.143.1.2 otaMsgTlv	167
8.143.1.3 timeTlv	167
8.144NASQmiCbkNasSystemSelPrefInd Struct Reference	167
8.144.1 Field Documentation	168
8.144.1.1 BPTlv	168
8.144.1.2 EMTlv	168
8.144.1.3 GWAOPTlv	168
8.144.1.4 LBPTlv	168
8.144.1.5 MPTlv	168
8.144.1.6 NSPTlv	168
8.144.1.7 PRLPTlv	168
8.144.1.8 RPTlv	168
8.144.1.9 SDPTlv	168
8.145NASRoamPreferenceTlv Struct Reference	168
8.145.1 Field Documentation	168
8.145.1.1 RoamPref	168
8.145.1.2 TlvPresent	168
8.146NASServDomainPrefTlv Struct Reference	168
8.146.1 Field Documentation	168
8.146.1.1 SrvDomainPref	168
8.146.1.2 TlvPresent	168
8.147NASServingSystemInfo Struct Reference	168
8.147.1 Detailed Description	169
8.147.2 Field Documentation	170
8.147.2.1 csAttachState	170
8.147.2.2 hdrPersonality	170
8.147.2.3 psAttachState	170
8.147.2.4 radiolInterfaceList	170
8.147.2.5 radiolInterfaceNo	170
8.147.2.6 registrationState	170
8.147.2.7 selectedNetwork	170
8.148NASTimeInfoTlv Struct Reference	170
8.148.1 Field Documentation	170
8.148.1.1 time	170
8.148.1.2 TlvPresent	170

8.149newMTMessageTlv Struct Reference	170
8.149.1 Detailed Description	170
8.149.2 Field Documentation	170
8.149.2.1 MTMessageInfo	171
8.149.2.2 TlvPresent	171
8.150pack_dms_GetCustFeaturesV2_t Struct Reference	171
8.150.1 Detailed Description	171
8.150.2 Field Documentation	171
8.150.2.1 cust_id	171
8.150.2.2 list_type	171
8.150.2.3 Tlvresult	171
8.151pack_dms_SetCrashAction_t Struct Reference	171
8.151.1 Detailed Description	171
8.151.2 Field Documentation	172
8.151.2.1 crashAction	172
8.152pack_dms_SetCustFeature_t Struct Reference	172
8.152.1 Field Documentation	172
8.152.1.1 DHCPRelayEnabled	172
8.152.1.2 DisableIMSI	172
8.152.1.3 GpsEnable	172
8.152.1.4 GPSPLM	172
8.152.1.5 GPSSel	172
8.152.1.6 IPFamSupport	172
8.152.1.7 IsVoiceEnabled	172
8.152.1.8 RMAutoConnect	172
8.152.1.9 SMSSupport	172
8.153pack_dms_SetCustFeaturesV2_t Struct Reference	172
8.153.1 Detailed Description	173
8.153.2 Field Documentation	173
8.153.2.1 cust_id	173
8.153.2.2 cust_value	173
8.153.2.3 Tlvresult	173
8.153.2.4 value_length	173
8.154pack_dms_SetEventReport_t Struct Reference	173
8.154.1 Field Documentation	173
8.154.1.1 mode	173
8.155pack_dms_SetPower_t Struct Reference	173
8.155.1 Field Documentation	173
8.155.1.1 mode	174
8.155.1.2 Tlvresult	174

8.156pack_dms_SetUSBComp_t Struct Reference	174
8.156.1 Field Documentation	174
8.156.1.1 Tlvresult	174
8.156.1.2 USBComp	174
8.157pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	174
8.157.1 Detailed Description	174
8.157.2 Field Documentation	174
8.157.2.1 resetInfoInd	174
8.158pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	174
8.158.1 Detailed Description	174
8.158.2 Field Documentation	175
8.158.2.1 pDestSMSContent	175
8.158.2.2 pDestSMSNum	175
8.159pack_dms_UIMGetICCID_t Struct Reference	175
8.159.1 Detailed Description	175
8.159.2 Field Documentation	175
8.159.2.1 Tlvresult	175
8.160pack_fms_GetImagesPreference_t Struct Reference	175
8.160.1 Detailed Description	175
8.160.2 Field Documentation	175
8.160.2.1 Tlvresult	175
8.161pack_fms_GetStoredImages_t Struct Reference	176
8.161.1 Detailed Description	176
8.161.2 Field Documentation	176
8.161.2.1 Tlvresult	176
8.162pack_fms_SetImagesPreference_t Struct Reference	176
8.162.1 Detailed Description	176
8.162.2 Field Documentation	177
8.162.2.1 bForceDownload	177
8.162.2.2 imageListSize	177
8.162.2.3 modemindex	177
8.162.2.4 pImageList	177
8.162.2.5 Tlvresult	177
8.163pack_loc_Delete_Assist_Data_t Struct Reference	177
8.163.1 Detailed Description	177
8.163.2 Field Documentation	177
8.163.2.1 pBdsSVInfo	177
8.163.2.2 pCellDb	177
8.163.2.3 pCikInfo	178
8.163.2.4 pGnssData	178

8.163.2.5 pSVInfo	178
8.163.2.6 Tlvresult	178
8.164pack_loc_EventRegister_t Struct Reference	178
8.164.1 Detailed Description	178
8.164.2 Field Documentation	180
8.164.2.1 eventRegister	180
8.164.2.2 Tlvresult	180
8.165pack_loc_SetExtPowerState_t Struct Reference	180
8.165.1 Detailed Description	180
8.165.2 Field Documentation	180
8.165.2.1 extPowerState	180
8.165.2.2 Tlvresult	180
8.166pack_loc_SetOperationMode_t Struct Reference	180
8.166.1 Detailed Description	181
8.166.2 Field Documentation	181
8.166.2.1 mode	181
8.166.2.2 Tlvresult	181
8.167pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	181
8.167.1 Detailed Description	181
8.167.2 Field Documentation	181
8.167.2.1 Tlvresult	181
8.167.2.2 xid	181
8.168pack_loc_Start_t Struct Reference	181
8.168.1 Detailed Description	182
8.168.2 Field Documentation	183
8.168.2.1 pApplicationInfo	183
8.168.2.2 pConfigAltitudeAssumed	183
8.168.2.3 pHorizontalAccuracyLvl	183
8.168.2.4 pIntermediateReportState	183
8.168.2.5 pMinIntervalTime	183
8.168.2.6 pRecurrenceType	183
8.168.2.7 SessionId	183
8.168.2.8 Tlvresult	183
8.169pack_loc_Stop_t Struct Reference	183
8.169.1 Detailed Description	183
8.169.2 Field Documentation	183
8.169.2.1 SessionId	183
8.169.2.2 Tlvresult	183
8.170pack_nas_SetACCOLC_t Struct Reference	183
8.170.1 Detailed Description	184

8.170.2 Field Documentation	184
8.170.2.1 accolc	184
8.170.2.2 spc	184
8.171 pack_nas_SetNetworkPreference_t Struct Reference	184
8.171.1 Detailed Description	184
8.171.2 Field Documentation	184
8.171.2.1 Duration	185
8.171.2.2 TechnologyPref	185
8.171.2.3 Tlvresult	185
8.172 pack_nas_SLQSGetPLMNName_t Struct Reference	185
8.172.1 Detailed Description	185
8.172.2 Field Documentation	185
8.172.2.1 mcc	185
8.172.2.2 mnc	185
8.172.2.3 pMncPcsStatus	185
8.173 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference	185
8.173.1 Detailed Description	186
8.173.2 Field Documentation	186
8.173.2.1 pChangeDuration	186
8.173.2.2 pMncPcsDigitStatus	186
8.173.2.3 pMNRInfo	186
8.173.2.4 regAction	186
8.174 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference	186
8.174.1 Detailed Description	187
8.174.2 Field Documentation	189
8.174.2.1 pCDMAECIODelta	190
8.174.2.2 pCDMAECIOThresh	190
8.174.2.3 pCDMARSSIDelta	190
8.174.2.4 pCDMARSSIThresh	190
8.174.2.5 pGSMRSSIDelta	190
8.174.2.6 pGSMRSSIThresh	190
8.174.2.7 pHDRECIODelta	190
8.174.2.8 pHDRECIOThresh	190
8.174.2.9 pHDRIODelta	190
8.174.2.10 pHDRIOThresh	190
8.174.2.11 pHDRRSSIDelta	190
8.174.2.12 pHDRRSSIThresh	190
8.174.2.13 pHDRSINRDelta	190
8.174.2.14 pHDRSINRThresh	190
8.174.2.15 pLTERSRPDelta	190

8.174.2.16	pLTERSRPThresh	190
8.174.2.17	pLTERSRQDelta	190
8.174.2.18	pLTERSRQThresh	190
8.174.2.19	pLTERSSIDelta	190
8.174.2.20	pLTERSSIThresh	190
8.174.2.21	pLTESigRptConfig	190
8.174.2.22	pLTESNRDelta	190
8.174.2.23	pLTESNRThresh	190
8.174.2.24	pTDSCDMAECIODelta	190
8.174.2.25	pTDSCDMAECIOThresh	190
8.174.2.26	pTDSCDMARSCPDelta	190
8.174.2.27	pTDSCDMARSCPThresh	190
8.174.2.28	pTDSCDMARSSIDelta	190
8.174.2.29	pTDSCDMARSSIThresh	191
8.174.2.30	pTDSCDMASINRDelta	191
8.174.2.31	pTDSCDMASINRThresh	191
8.174.2.32	pWCDMAECIODelta	191
8.174.2.33	pWCDMAECIOThresh	191
8.174.2.34	pWCDMARSSIDelta	191
8.174.2.35	pWCDMARSSIThresh	191
8.175	pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference	191
8.175.1	Detailed Description	191
8.175.2	Field Documentation	193
8.175.2.1	pDDTMInd	193
8.175.2.2	pDualStandByPrefInd	193
8.175.2.3	pErrorRateInd	193
8.175.2.4	pHDRNewUATIAssInd	193
8.175.2.5	pHDRSessionCloseInd	193
8.175.2.6	pLTECphyCa	193
8.175.2.7	pManagedRoamingInd	193
8.175.2.8	pNetworkTimeInd	193
8.175.2.9	pServingSystemInd	193
8.175.2.10	pSignalStrengthInd	193
8.175.2.11	pSubscriptionInfoInd	193
8.175.2.12	pSysInfoInd	194
8.175.2.13	pSystemSelectionInd	194
8.176	pack_nas_SLQSNasSwiOTAMessageCallback_t Struct Reference	194
8.176.1	Detailed Description	194
8.176.2	Field Documentation	194
8.176.2.1	gsmUmtsDI	195

8.176.2.2 gsmUmtsUI	195
8.176.2.3 lteEmmDI	195
8.176.2.4 lteEmmUI	195
8.176.2.5 lteEsmDI	195
8.176.2.6 lteEsmUI	195
8.176.2.7 pRankIndicatorInd	195
8.177 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference	195
8.177.1 Detailed Description	195
8.177.2 Field Documentation	195
8.177.2.1 bEnable	195
8.177.2.2 pSigIndReq	195
8.178 pack_nas_SLQSSetSysSelectionPref_t Struct Reference	195
8.178.1 Detailed Description	196
8.178.2 Field Documentation	199
8.178.2.1 pAcqOrderPref	199
8.178.2.2 pBandPref	199
8.178.2.3 pChgDuration	199
8.178.2.4 pCSGID	199
8.178.2.5 pEmerMode	199
8.178.2.6 pGWAcqOrderPref	199
8.178.2.7 pLTEBandPref	200
8.178.2.8 pMNCIncPCSDigStat	200
8.178.2.9 pModePref	200
8.178.2.10 pNetSelPref	200
8.178.2.11 pPRLPref	200
8.178.2.12 pRAT	200
8.178.2.13 pRoamPref	200
8.178.2.14 pSrvDomainPref	200
8.178.2.15 pSrvRegRestriction	200
8.178.2.16 pTdsdmaBandPref	200
8.179 pack_qmi_t Struct Reference	200
8.179.1 Detailed Description	200
8.179.2 Field Documentation	200
8.179.2.1 msgid	200
8.179.2.2 svc	200
8.179.2.3 timeout	200
8.179.2.4 xid	200
8.180 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference	200
8.180.1 Detailed Description	201
8.180.2 Field Documentation	201

8.180.2.1 apnId	201
8.181pack_qos_SLQSQosSwiReadDataStats_t Struct Reference	201
8.181.1 Detailed Description	201
8.181.2 Field Documentation	201
8.181.2.1 apnId	201
8.182pack_qos_SLQSSetQosEventCallback_t Struct Reference	201
8.182.1 Detailed Description	201
8.182.2 Field Documentation	202
8.182.2.1 enable	202
8.183pack_sms_SendSMS_t Struct Reference	202
8.183.1 Detailed Description	202
8.183.2 Field Documentation	202
8.183.2.1 messageFormat	202
8.183.2.2 messageSize	202
8.183.2.3 pLinktimer	202
8.183.2.4 pMessage	202
8.184pack_sms_SetNewSMSCallback_t Struct Reference	202
8.184.1 Detailed Description	203
8.184.2 Field Documentation	203
8.184.2.1 status	203
8.185pack_sms_SLQSDeleteSMS_t Struct Reference	203
8.185.1 Detailed Description	203
8.185.2 Field Documentation	203
8.185.2.1 pMessageIndex	204
8.185.2.2 pMessageMode	204
8.185.2.3 pMessageTag	204
8.185.2.4 storageType	204
8.186pack_sms_SLQSGetSMS_t Struct Reference	204
8.186.1 Detailed Description	204
8.186.2 Field Documentation	204
8.186.2.1 messageIndex	204
8.186.2.2 pMessageMode	204
8.186.2.3 storageType	204
8.187pack_sms_SLQSGetSMSList_t Struct Reference	204
8.187.1 Detailed Description	204
8.187.2 Field Documentation	205
8.187.2.1 pMessageMode	205
8.187.2.2 pRequestedTag	205
8.187.2.3 storageType	205
8.188pack_sms_SLQSModifySMSStatus_t Struct Reference	205

8.188.1 Detailed Description	205
8.188.2 Field Documentation	206
8.188.2.1 messageIndex	206
8.188.2.2 messageTag	206
8.188.2.3 pMessageMode	206
8.188.2.4 storageType	206
8.189pack_swiloc_SwiLocSetAutoStart_t Struct Reference	206
8.189.1 Detailed Description	206
8.189.2 Field Documentation	207
8.189.2.1 fix_rate	207
8.189.2.2 fix_type	207
8.189.2.3 function	207
8.189.2.4 max_dist	207
8.189.2.5 max_time	207
8.189.2.6 set_fix_rate	207
8.189.2.7 set_fix_type	207
8.189.2.8 set_function	207
8.189.2.9 set_max_dist	207
8.189.2.10set_max_time	207
8.190pack_swioama_SLQSOMADMCancelSession_t Struct Reference	207
8.190.1 Detailed Description	208
8.190.2 Field Documentation	208
8.190.2.1 sessionType	208
8.191pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference	208
8.191.1 Detailed Description	208
8.191.2 Field Documentation	208
8.191.2.1 SessionType	208
8.192pack_swioama_SLQSOMADMSendSelection_t Struct Reference	208
8.192.1 Detailed Description	209
8.192.2 Field Documentation	209
8.192.2.1 pDeferTime	209
8.192.2.2 pRejectReason	209
8.192.2.3 selection	209
8.193pack_swioama_SLQSOMADMSetSettings_t Struct Reference	209
8.193.1 Detailed Description	209
8.193.2 Field Documentation	210
8.193.2.1 FOTAdownload	210
8.193.2.2 FOTAUpdate	210
8.193.2.3 pAutosdm	210
8.193.2.4 pFwAutoCheck	210

8.194pack_swima_SLQSOMADMStartSession_t Struct Reference	210
8.194.1 Detailed Description	210
8.194.2 Field Documentation	210
8.194.2.1 sessionType	210
8.195pack_uim_ChangePin_t Struct Reference	211
8.195.1 Detailed Description	211
8.195.2 Field Documentation	211
8.195.2.1 changePIN	211
8.195.2.2 EncryptedPIN1	211
8.195.2.3 pIndicationToken	211
8.195.2.4 pKeyReferenceID	211
8.195.2.5 sessionInfo	211
8.195.2.6 Tlvresult	212
8.196pack_uim_ReadTransparent_t Struct Reference	212
8.196.1 Detailed Description	212
8.196.2 Field Documentation	212
8.196.2.1 fileIndex	212
8.196.2.2 pEncryptData	212
8.196.2.3 pIndicationToken	212
8.196.2.4 readTransparent	213
8.196.2.5 sessionInfo	213
8.196.2.6 Tlvresult	213
8.197pack_uim_SetPinProtection_t Struct Reference	213
8.197.1 Detailed Description	213
8.197.2 Field Documentation	213
8.197.2.1 EncryptedPIN1	213
8.197.2.2 pIndicationToken	213
8.197.2.3 pinProtection	214
8.197.2.4 pKeyReferenceID	214
8.197.2.5 sessionInfo	214
8.197.2.6 Tlvresult	214
8.198pack_uim_SLQSUIEventRegister_t Struct Reference	214
8.198.1 Detailed Description	214
8.198.2 Field Documentation	214
8.198.2.1 eventMask	214
8.199pack_uim_SLQSUISSwitchSlot_t Struct Reference	214
8.199.1 Detailed Description	214
8.199.2 Field Documentation	215
8.199.2.1 bLogicalSlot	215
8.199.2.2 ulPhysicalSlot	215

8.200pack_uim_UnblockPin_t Struct Reference	215
8.200.1 Detailed Description	215
8.200.2 Field Documentation	216
8.200.2.1 EncryptedPIN1	216
8.200.2.2 pIndicationToken	216
8.200.2.3 pinProtection	216
8.200.2.4 pKeyReferenceID	216
8.200.2.5 sessionInfo	216
8.200.2.6 Tlvresult	216
8.201pack_uim_VerifyPin_t Struct Reference	216
8.201.1 Detailed Description	216
8.201.2 Field Documentation	217
8.201.2.1 pEncryptedPIN1	217
8.201.2.2 pIndicationToken	217
8.201.2.3 pKeyReferenceID	217
8.201.2.4 sessionInfo	217
8.201.2.5 Tlvresult	217
8.201.2.6 verifyPIN	217
8.202pack_wds_GetDefaultProfile_t Struct Reference	217
8.202.1 Detailed Description	217
8.202.2 Field Documentation	217
8.202.2.1 profiletype	217
8.203pack_wds_GetDefaultProfileNum_t Struct Reference	217
8.203.1 Detailed Description	218
8.203.2 Field Documentation	218
8.203.2.1 family	218
8.203.2.2 type	218
8.204pack_wds_GetDormancyState_t Struct Reference	218
8.205pack_wds_GetLastMobileIPError_t Struct Reference	218
8.206pack_wds_GetMobileIP_t Struct Reference	218
8.207pack_wds_GetMobileIPProfile_t Struct Reference	218
8.207.1 Detailed Description	218
8.207.2 Field Documentation	218
8.207.2.1 index	218
8.208pack_wds_GetPacketStatus_t Struct Reference	218
8.208.1 Detailed Description	219
8.208.2 Field Documentation	219
8.208.2.1 statmask	219
8.209pack_wds_GetSessionDuration_t Struct Reference	219
8.210pack_wds_RMSetTransferStatistics_t Struct Reference	219

8.210.1 Detailed Description	219
8.210.2 Field Documentation	219
8.210.2.1 RmTrasferStaticsReq	219
8.211 pack_wds_SetDefaultProfile_t Struct Reference	219
8.211.1 Detailed Description	219
8.211.2 Field Documentation	220
8.211.2.1 authentication	220
8.211.2.2 ipAddress	220
8.211.2.3 pApnname	220
8.211.2.4 pdpType	220
8.211.2.5 pName	220
8.211.2.6 pPassword	220
8.211.2.7 primaryDNS	220
8.211.2.8 profileType	220
8.211.2.9 pUsername	220
8.211.2.10secondaryDNS	220
8.212 pack_wds_SetDefaultProfileNum_t Struct Reference	220
8.212.1 Field Documentation	220
8.212.1.1 family	220
8.212.1.2 index	220
8.212.1.3 type	220
8.213 pack_wds_SetMobileIPProfile_t Struct Reference	220
8.213.1 Detailed Description	221
8.213.2 Field Documentation	221
8.213.2.1 index	221
8.213.2.2 pAAASPI	221
8.213.2.3 pAddress	221
8.213.2.4 pEnabled	221
8.213.2.5 pHASPI	221
8.213.2.6 pMNAAA	221
8.213.2.7 pMNHA	221
8.213.2.8 pNAI	221
8.213.2.9 pPrimaryHA	221
8.213.2.10pRevTunneling	221
8.213.2.11pSecondaryHA	222
8.213.2.12spc	222
8.214 pack_wds_SLQSCreateProfile_t Struct Reference	222
8.214.1 Detailed Description	222
8.214.2 Field Documentation	222
8.214.2.1 pCurProfile	222

8.214.2.2 pProfileId	222
8.214.2.3 pProfileType	222
8.215pack_wds_SLQSDeleteProfile_t Struct Reference	222
8.215.1 Detailed Description	222
8.215.2 Field Documentation	223
8.215.2.1 profileIndex	223
8.215.2.2 profileType	223
8.216pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	223
8.217pack_wds_SLQSGetDataBearerTechnology_t Struct Reference	223
8.218pack_wds_SLQSGetDUNCallInfo_t Struct Reference	223
8.218.1 Detailed Description	223
8.218.2 Field Documentation	223
8.218.2.1 Mask	223
8.218.2.2 pReportChannelRate	223
8.218.2.3 pReportConnStatus	223
8.218.2.4 pReportDataBearerTech	223
8.218.2.5 pReportDormStatus	223
8.218.2.6 pTransferStatInd	224
8.219pack_wds_SLQSGetProfileSettings_t Struct Reference	224
8.219.1 Detailed Description	224
8.219.2 Field Documentation	224
8.219.2.1 ProfileId	224
8.219.2.2 ProfileType	224
8.220pack_wds_SLQSGetRuntimeSettings_t Struct Reference	224
8.220.1 Detailed Description	224
8.220.2 Field Documentation	225
8.220.2.1 pReqSettings	225
8.221pack_wds_SLQSModifyProfile_t Struct Reference	225
8.221.1 Detailed Description	225
8.221.2 Field Documentation	226
8.221.2.1 curProfile	226
8.221.2.2 pProfileId	226
8.221.2.3 pProfileType	226
8.222pack_wds_SLQSSet3GPPConfigItem_t Struct Reference	226
8.222.1 Detailed Description	226
8.222.2 Field Documentation	227
8.222.2.1 _3gppRelease	227
8.222.2.2 defaultPDNEnabled	227
8.222.2.3 LTEAttachProfileList	227
8.222.2.4 LTEAttachProfileListLen	227

8.222.2.5 profileList	227
8.223pack_wds_SLQSSetIPFamilyPreference_t Struct Reference	227
8.223.1 Detailed Description	227
8.223.2 Field Documentation	227
8.223.2.1 IPFamilyPreference	227
8.224pack_wds_SLQSSetWdsEventCallback_t Struct Reference	227
8.224.1 Detailed Description	228
8.224.2 Field Documentation	228
8.224.2.1 currentDataBearer	228
8.224.2.2 dataBearer	228
8.224.2.3 dataSystemStatus	228
8.224.2.4 dormancyStatus	228
8.224.2.5 interval	228
8.224.2.6 mobileIP	228
8.224.2.7 transferStats	228
8.225pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	228
8.225.1 Detailed Description	228
8.225.2 Field Documentation	228
8.225.2.1 pProfileId	228
8.226pack_wds_SLQSSStartDataSession_t Struct Reference	228
8.226.1 Detailed Description	229
8.226.2 Field Documentation	229
8.226.2.1 pAuth	229
8.226.2.2 pPass	229
8.226.2.3 pprofileid3gpp	229
8.226.2.4 pprofileid3gpp2	229
8.226.2.5 pTech	230
8.226.2.6 pUser	230
8.227pack_wds_SLQSSStopDataSession_t Struct Reference	230
8.227.1 Detailed Description	230
8.227.2 Field Documentation	230
8.227.2.1 psid	230
8.228pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	230
8.228.1 Detailed Description	230
8.228.2 Field Documentation	230
8.228.2.1 contextId	230
8.228.2.2 contextType	230
8.229PackCreateProfileOut Struct Reference	230
8.229.1 Field Documentation	230
8.229.1.1 ExtErrorCode	230

8.229.1.2 ProfileIndex	231
8.229.1.3 ProfileType	231
8.230packetDyingGaspCfg Struct Reference	231
8.230.1 Detailed Description	231
8.230.2 Field Documentation	231
8.230.2.1 pDestSMSContent	231
8.230.2.2 pDestSMSNum	231
8.231packetDyingGaspStatistics Struct Reference	231
8.231.1 Detailed Description	231
8.231.2 Field Documentation	231
8.231.2.1 pSMSAttemptedFlag	231
8.231.2.2 pTimeStamp	232
8.232qmiSmsMessageList Struct Reference	232
8.232.1 Detailed Description	232
8.232.2 Field Documentation	232
8.232.2.1 messageIndex	232
8.232.2.2 messageTag	232
8.233qmiWDSDataBearerTechnology Struct Reference	232
8.233.1 Detailed Description	232
8.233.2 Field Documentation	232
8.233.2.1 currentNetwork	232
8.233.2.2 ratMask	232
8.233.2.3 soMask	232
8.234RFBandInfoElements Struct Reference	232
8.234.1 Detailed Description	233
8.234.2 Field Documentation	233
8.234.2.1 activeBandClass	233
8.234.2.2 activeChannel	233
8.234.2.3 radiolInterface	233
8.235rmTrasferStaticsReq Struct Reference	233
8.235.1 Detailed Description	233
8.235.2 Field Documentation	233
8.235.2.1 bResetStatistics	233
8.235.2.2 ulMask	233
8.236slot_t Struct Reference	233
8.236.1 Detailed Description	234
8.236.2 Field Documentation	234
8.236.2.1 bICCID	234
8.236.2.2 bICCIDLength	234
8.236.2.3 bLogicalSlot	234

8.236.2.4 uPhyCardStatus	234
8.236.2.5 uPhySlotStatus	234
8.237slotInf Struct Reference	234
8.237.1 Detailed Description	235
8.237.2 Field Documentation	236
8.237.2.1 AppStatus	236
8.237.2.2 cardState	236
8.237.2.3 errorState	236
8.237.2.4 numApp	236
8.237.2.5 upinRetries	236
8.237.2.6 upinState	236
8.237.2.7 upukRetries	236
8.238slots_t Struct Reference	236
8.238.1 Field Documentation	236
8.238.1.1 uimSlotStatus	236
8.239sMSCAddress Struct Reference	236
8.239.1 Detailed Description	236
8.239.2 Field Documentation	236
8.239.2.1 data	237
8.239.2.2 length	237
8.240sMSCAddressTlv Struct Reference	237
8.240.1 Detailed Description	237
8.240.2 Field Documentation	237
8.240.2.1 SMSCInfo	237
8.240.2.2 TlvPresent	237
8.241sMSEtwsMessage Struct Reference	237
8.241.1 Detailed Description	237
8.241.2 Field Documentation	237
8.241.2.1 data	238
8.241.2.2 length	238
8.241.2.3 notificationType	238
8.242sMSEtwsMessageTlv Struct Reference	238
8.242.1 Detailed Description	238
8.242.2 Field Documentation	238
8.242.2.1 EtwsMessageInfo	238
8.242.2.2 TlvPresent	238
8.243sMSEtwsPlmn Struct Reference	238
8.243.1 Detailed Description	238
8.243.2 Field Documentation	238
8.243.2.1 mobileCountryCode	238

8.243.2.2 mobileNetworkCode	239
8.244sMSMessageMode Struct Reference	239
8.244.1 Detailed Description	239
8.244.2 Field Documentation	239
8.244.2.1 messageMode	239
8.245sMSMTMessage Struct Reference	239
8.245.1 Detailed Description	239
8.245.2 Field Documentation	239
8.245.2.1 messageIndex	239
8.245.2.2 storageType	239
8.246sMSOnIMS Struct Reference	239
8.246.1 Detailed Description	239
8.246.2 Field Documentation	239
8.246.2.1 smsOnIMS	240
8.247sMSOnIMSTlv Struct Reference	240
8.247.1 Detailed Description	240
8.247.2 Field Documentation	240
8.247.2.1 IMSInfo	240
8.247.2.2 TlvPresent	240
8.248sMSTransferRouteMTMessage Struct Reference	240
8.248.1 Detailed Description	240
8.248.2 Field Documentation	241
8.248.2.1 ackIndicator	241
8.248.2.2 data	241
8.248.2.3 format	241
8.248.2.4 length	241
8.248.2.5 transactionID	241
8.249tdscdmaSigInfoExt Struct Reference	241
8.249.1 Detailed Description	241
8.249.2 Field Documentation	241
8.249.2.1 ecio	241
8.249.2.2 rscp	241
8.249.2.3 rssi	241
8.249.2.4 sinr	241
8.250transferRouteMessageTlv Struct Reference	241
8.250.1 Detailed Description	242
8.250.2 Field Documentation	242
8.250.2.1 TlvPresent	242
8.250.2.2 TransferRouteMTMessageInfo	242
8.251transferStatInd Struct Reference	242

8.251.1 Detailed Description	242
8.251.2 Field Documentation	242
8.251.2.1 StatsMask	242
8.251.2.2 StatsPeriod	242
8.252uim_appStatus Struct Reference	242
8.252.1 Detailed Description	243
8.252.2 Field Documentation	245
8.252.2.1 aidLength	245
8.252.2.2 aidVal	245
8.252.2.3 appState	245
8.252.2.4 appType	245
8.252.2.5 persoFeature	245
8.252.2.6 persoRetries	245
8.252.2.7 persoState	245
8.252.2.8 persoUnblockRetries	245
8.252.2.9 pin1Retries	245
8.252.2.10pin1State	245
8.252.2.11pin2Retries	245
8.252.2.12pin2State	245
8.252.2.13puk1Retries	245
8.252.2.14puk2Retries	245
8.252.2.15univPin	245
8.253uim_cardResult Struct Reference	245
8.253.1 Detailed Description	246
8.253.2 Field Documentation	246
8.253.2.1 sw1	246
8.253.2.2 sw2	246
8.254uim_cardStatus Struct Reference	246
8.254.1 Detailed Description	246
8.254.2 Field Documentation	247
8.254.2.1 index1xPri	247
8.254.2.2 index1xSec	247
8.254.2.3 indexGwPri	247
8.254.2.4 indexGwSec	247
8.254.2.5 numSlot	247
8.254.2.6 SlotInfo	247
8.255uim_changeUIMPIN Struct Reference	247
8.255.1 Detailed Description	247
8.255.2 Field Documentation	248
8.255.2.1 oldPINLen	248

8.255.2.2 oldPINVal	248
8.255.2.3 pinID	248
8.255.2.4 pinLen	248
8.255.2.5 pinVal	248
8.256uim_encryptedPIN1 Struct Reference	248
8.256.1 Detailed Description	248
8.256.2 Field Documentation	249
8.256.2.1 pin1Len	249
8.256.2.2 pin1Val	249
8.257uim_fileInfo Struct Reference	249
8.257.1 Detailed Description	249
8.257.2 Field Documentation	249
8.257.2.1 fileID	249
8.257.2.2 path	249
8.257.2.3 pathLen	249
8.258uim_hotSwapStatus Struct Reference	249
8.258.1 Detailed Description	250
8.258.2 Field Documentation	250
8.258.2.1 hotSwap	250
8.258.2.2 hotSwapLength	250
8.259uim_readResult Struct Reference	250
8.259.1 Detailed Description	250
8.259.2 Field Documentation	250
8.259.2.1 content	250
8.259.2.2 contentLen	251
8.260uim_readTransparentInfo Struct Reference	251
8.260.1 Detailed Description	251
8.260.2 Field Documentation	251
8.260.2.1 length	251
8.260.2.2 offset	251
8.261uim_remainingRetries Struct Reference	251
8.261.1 Detailed Description	251
8.261.2 Field Documentation	252
8.261.2.1 unblockLeft	252
8.261.2.2 verifyLeft	252
8.262uim_sessionInformation Struct Reference	252
8.262.1 Detailed Description	252
8.262.2 Field Documentation	252
8.262.2.1 aid	252
8.262.2.2 aidLength	252

8.262.2.3 sessionType	253
8.263uim_setPINProtection Struct Reference	253
8.263.1 Detailed Description	253
8.263.2 Field Documentation	253
8.263.2.1 pinID	253
8.263.2.2 pinLength	253
8.263.2.3 pinOperation	253
8.263.2.4 pinValue	253
8.264uim_slotInfo Struct Reference	253
8.264.1 Detailed Description	254
8.264.2 Field Documentation	255
8.264.2.1 AppStatus	255
8.264.2.2 cardState	255
8.264.2.3 errorState	255
8.264.2.4 numApp	255
8.264.2.5 upinRetries	255
8.264.2.6 upinState	255
8.264.2.7 upukRetries	255
8.265uim_UIMSessionInformation Struct Reference	255
8.265.1 Detailed Description	255
8.265.2 Field Documentation	256
8.265.2.1 aid	256
8.265.2.2 aidLength	256
8.265.2.3 sessionType	256
8.266uim_unblockUIMPIN Struct Reference	256
8.266.1 Detailed Description	256
8.266.2 Field Documentation	257
8.266.2.1 newPINLen	257
8.266.2.2 newPINVal	257
8.266.2.3 pinID	257
8.266.2.4 pukLen	257
8.266.2.5 pukVal	257
8.267uim_verifyUIMPIN Struct Reference	257
8.267.1 Detailed Description	257
8.267.2 Field Documentation	257
8.267.2.1 pinID	257
8.267.2.2 pinLen	257
8.267.2.3 pinVal	257
8.268unpack_dms_GetActivationState_t Struct Reference	257
8.268.1 Detailed Description	258

8.268.2 Field Documentation	258
8.268.2.1 state	258
8.269unpack_dms_GetBandCapability_t Struct Reference	258
8.269.1 Field Documentation	258
8.269.1.1 BandCapability	258
8.269.1.2 Tlvresult	258
8.270unpack_dms_GetCrashAction_t Struct Reference	258
8.270.1 Field Documentation	258
8.270.1.1 DevCrashState	258
8.270.1.2 Tlvresult	258
8.271unpack_dms_GetCustFeature_t Struct Reference	259
8.271.1 Field Documentation	259
8.271.1.1 DHCPRelayEnabled	259
8.271.1.2 DisableIMSI	259
8.271.1.3 GpsEnable	259
8.271.1.4 GPSLPM	259
8.271.1.5 GPSSel	259
8.271.1.6 IPFamSupport	259
8.271.1.7 IsVoiceEnabled	259
8.271.1.8 RMAutoConnect	259
8.271.1.9 SMSSupport	259
8.271.1.10Tlvresult	259
8.272unpack_dms_GetCustFeaturesV2_t Struct Reference	259
8.272.1 Detailed Description	259
8.272.2 Field Documentation	260
8.272.2.1 GetCustomFeatureV2	260
8.272.2.2 Tlvresult	260
8.273unpack_dms_GetDeviceCap_t Struct Reference	260
8.273.1 Field Documentation	260
8.273.1.1 DataServiceCapability	260
8.273.1.2 MaxRXChannelRate	260
8.273.1.3 MaxTXChannelRate	260
8.273.1.4 Radiofaces	260
8.273.1.5 RadiofacesSize	260
8.273.1.6 SimCapability	260
8.273.1.7 Tlvresult	260
8.274unpack_dms_GetDeviceCapabilities_t Struct Reference	260
8.274.1 Detailed Description	260
8.274.2 Field Documentation	261
8.274.2.1 dataServiceCaCapability	261

8.274.2.2 maxRxChannelRate	261
8.274.2.3 maxTxChannelRate	261
8.274.2.4 Radiolfaces	261
8.274.2.5 radiolfacesSize	261
8.274.2.6 simCapability	261
8.275unpack_dms_GetDeviceHardwareRev_t Struct Reference	261
8.275.1 Field Documentation	261
8.275.1.1 String	261
8.275.1.2 stringSize	261
8.275.1.3 Tlvresult	261
8.276unpack_dms_GetDeviceMfr_t Struct Reference	261
8.276.1 Field Documentation	261
8.276.1.1 String	261
8.276.1.2 stringSize	261
8.276.1.3 Tlvresult	261
8.277unpack_dms_GetDeviceSerialNumbers_t Struct Reference	262
8.277.1 Field Documentation	262
8.277.1.1 esnSize	262
8.277.1.2 ESNString	262
8.277.1.3 imeiSize	262
8.277.1.4 IMEIString	262
8.277.1.5 imeiSvnSize	262
8.277.1.6 lmeiSvnString	262
8.277.1.7 meidSize	262
8.277.1.8 MEIDString	262
8.277.1.9 Tlvresult	262
8.278unpack_dms_GetFirmwareInfo_t Struct Reference	262
8.278.1 Detailed Description	262
8.278.2 Field Documentation	263
8.278.2.1 appversion_str	263
8.278.2.2 bootversion_str	263
8.278.2.3 carrier_str	263
8.278.2.4 cur_carr_name	263
8.278.2.5 cur_carr_rev	263
8.278.2.6 modelid_str	263
8.278.2.7 packageid_str	263
8.278.2.8 priversion_str	263
8.278.2.9 sku_str	263
8.278.2.10Tlvresult	263
8.279unpack_dms_GetFirmwareRevision_t Struct Reference	263

8.279.1 Field Documentation	263
8.279.1.1 amssSize	263
8.279.1.2 AMSSString	263
8.279.1.3 PRIString	264
8.279.1.4 Tlvresult	264
8.280unpack_dms_GetFirmwareRevisions_t Struct Reference	264
8.280.1 Detailed Description	264
8.280.2 Field Documentation	264
8.280.2.1 amssSize	264
8.280.2.2 AMSSString	264
8.280.2.3 bootSize	264
8.280.2.4 BootString	264
8.280.2.5 priSize	264
8.280.2.6 PRIString	264
8.280.2.7 Tlvresult	264
8.281unpack_dms_GetFSN_t Struct Reference	264
8.281.1 Field Documentation	264
8.281.1.1 String	264
8.281.1.2 Tlvresult	264
8.282unpack_dms_GetHardwareRevision_t Struct Reference	265
8.282.1 Detailed Description	265
8.282.2 Field Documentation	265
8.282.2.1 hwVer	265
8.283unpack_dms_GetIMSI_t Struct Reference	265
8.283.1 Field Documentation	265
8.283.1.1 imsi	265
8.283.1.2 Tlvresult	265
8.284unpack_dms_GetModelID_t Struct Reference	265
8.284.1 Detailed Description	265
8.284.2 Field Documentation	265
8.284.2.1 modelid	265
8.284.2.2 Tlvresult	265
8.285unpack_dms_GetNetworkTime_t Struct Reference	265
8.285.1 Detailed Description	266
8.285.2 Field Documentation	266
8.285.2.1 source	266
8.285.2.2 timestamp	266
8.285.2.3 Tlvresult	266
8.286unpack_dms_GetPower_t Struct Reference	266
8.286.1 Detailed Description	266

8.286.2 Field Documentation	267
8.286.2.1 HardwareControlledMode	267
8.286.2.2 OfflineReason	267
8.286.2.3 OperationMode	267
8.286.2.4 Tlvresult	267
8.287unpack_dms_GetPRLVersion_t Struct Reference	267
8.287.1 Field Documentation	267
8.287.1.1 Tlvresult	267
8.287.1.2 u16PRLVersion	267
8.287.1.3 u8PRLPreference	267
8.288unpack_dms_GetSerialNumbers_t Struct Reference	267
8.288.1 Detailed Description	267
8.288.2 Field Documentation	267
8.288.2.1 esn	267
8.288.2.2 imei_no	267
8.288.2.3 imeisv_svn	267
8.288.2.4 meid	268
8.289unpack_dms_GetUSBComp_t Struct Reference	268
8.289.1 Field Documentation	268
8.289.1.1 NumSupUSBComps	268
8.289.1.2 SupUSBComps	268
8.289.1.3 Tlvresult	268
8.289.1.4 USBComp	268
8.290unpack_dms_GetVoiceNumber_t Struct Reference	268
8.290.1 Field Documentation	268
8.290.1.1 MIN	268
8.290.1.2 minSize	268
8.290.1.3 Tlvresult	268
8.290.1.4 VoiceNumber	268
8.290.1.5 voiceNumberSize	268
8.291unpack_dms_SetCrashAction_t Struct Reference	268
8.291.1 Detailed Description	268
8.291.2 Field Documentation	269
8.291.2.1 notused	269
8.292unpack_dms_SetCustFeature_t Struct Reference	269
8.292.1 Field Documentation	269
8.292.1.1 Tlvresult	269
8.293unpack_dms_SetCustFeaturesV2_t Struct Reference	269
8.293.1 Detailed Description	269
8.293.2 Field Documentation	269

8.293.2.1 Tlvresult	269
8.294unpack_dms_SetEventReport_ind_t Struct Reference	269
8.294.1 Detailed Description	269
8.294.2 Field Documentation	270
8.294.2.1 ActivationStatusTlv	270
8.294.2.2 OperatingModeTlv	270
8.294.2.3 Tlvresult	270
8.295unpack_dms_SetEventReport_t Struct Reference	270
8.295.1 Field Documentation	270
8.295.1.1 Tlvresult	270
8.296unpack_dms_SetFirmwarePreference_t Struct Reference	270
8.296.1 Field Documentation	270
8.296.1.1 Tlvresult	270
8.297unpack_dms_SetPower_t Struct Reference	270
8.297.1 Field Documentation	270
8.297.1.1 Tlvresult	270
8.298unpack_dms_SetUSBComp_t Struct Reference	270
8.298.1 Field Documentation	271
8.298.1.1 Tlvresult	271
8.299unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference	271
8.299.1 Detailed Description	271
8.299.2 Field Documentation	271
8.299.2.1 source	271
8.299.2.2 Tlvresult	272
8.299.2.3 type	272
8.300unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference	272
8.300.1 Detailed Description	272
8.300.2 Field Documentation	272
8.300.2.1 source	272
8.300.2.2 Tlvresult	272
8.300.2.3 type	272
8.301unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	273
8.301.1 Detailed Description	273
8.301.2 Field Documentation	273
8.301.2.1 Tlvresult	273
8.302unpack_dms_SLQSGetBandCapability_t Struct Reference	273
8.302.1 Detailed Description	273
8.302.2 Field Documentation	276
8.302.2.1 bandCapability	276
8.302.2.2 is_LteBandCapability_Available	276

8.302.2.3 is_TdsBandCapability_Available	276
8.302.2.4 LteBandCapability	276
8.302.2.5 TdsBandCapability	276
8.303unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference	276
8.303.1 Detailed Description	276
8.303.2 Field Documentation	276
8.303.2.1 Tlvresult	276
8.304unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference	276
8.304.1 Detailed Description	277
8.304.2 Field Documentation	277
8.304.2.1 pGetDyingGaspCfg	277
8.304.2.2 Tlvresult	277
8.305unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference	277
8.305.1 Detailed Description	277
8.305.2 Field Documentation	277
8.305.2.1 pGetDyingGaspStatistics	277
8.305.2.2 Tlvresult	277
8.306unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference	277
8.306.1 Detailed Description	277
8.306.2 Field Documentation	278
8.306.2.1 carrier	278
8.306.2.2 fwvers	278
8.306.2.3 numEntries	278
8.306.2.4 pCurrImglInfo	278
8.306.2.5 pkgver	278
8.306.2.6 priver	278
8.307unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference	278
8.307.1 Detailed Description	278
8.307.2 Field Documentation	279
8.307.2.1 imgType	279
8.307.2.2 logString	279
8.307.2.3 refData	279
8.307.2.4 refString	279
8.307.2.5 ResCode	279
8.307.2.6 Tlvresult	280
8.308unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	280
8.308.1 Detailed Description	280
8.308.2 Field Documentation	280
8.308.2.1 Tlvresult	280
8.309unpack_dms_UIMGetlCCID_t Struct Reference	280

8.309.1 Detailed Description	280
8.309.2 Field Documentation	280
8.309.2.1 String	280
8.309.2.2 stringSize	280
8.309.2.3 Tlvresult	281
8.310unpack_fms_GetImagesPreference_t Struct Reference	281
8.310.1 Detailed Description	281
8.310.2 Field Documentation	281
8.310.2.1 ImageListSize	281
8.310.2.2 pImageList	281
8.310.2.3 Tlvresult	281
8.311unpack_fms_GetStoredImages_t Struct Reference	281
8.311.1 Detailed Description	281
8.311.2 Field Documentation	282
8.311.2.1 imageList	282
8.311.2.2 imagelistSize	282
8.311.2.3 Tlvresult	282
8.312unpack_fms_SetImagesPreference_t Struct Reference	282
8.312.1 Detailed Description	282
8.312.2 Field Documentation	282
8.312.2.1 ImageTypes	282
8.312.2.2 ImageTypesSize	282
8.312.2.3 Tlvresult	282
8.313unpack_loc_BestAvailPos_Ind_t Struct Reference	282
8.313.1 Detailed Description	283
8.313.2 Field Documentation	287
8.313.2.1 pAltitudeWrtEllipsoid	287
8.313.2.2 pAltitudeWrtMeanSeaLevel	287
8.313.2.3 pGpsTime	287
8.313.2.4 pHeading	287
8.313.2.5 pHeadingUnc	287
8.313.2.6 pHorCirConf	287
8.313.2.7 pHorEllpConf	287
8.313.2.8 pHorReliability	287
8.313.2.9 pHorUncCircular	287
8.313.2.10pHorUncEllipseOrientAzimuth	287
8.313.2.11pHorUncEllipseSemiMajor	287
8.313.2.12pHorUncEllipseSemiMinor	288
8.313.2.13pLatitude	288
8.313.2.14pLongitude	288

8.313.2.15	pMagneticDeviation	288
8.313.2.16	pPrecisionDilution	288
8.313.2.17	pSensorDataUsage	288
8.313.2.18	pSpeedHorizontal	288
8.313.2.19	pSpeedUnc	288
8.313.2.20	pSpeedVertical	288
8.313.2.21	pSpeedVerticalUnc	288
8.313.2.22	pSvUsedforFix	288
8.313.2.23	pTechnologyMask	288
8.313.2.24	pTimeSrc	288
8.313.2.25	pTimestampUtc	288
8.313.2.26	pTimeUnc	288
8.313.2.27	pVertConfidence	288
8.313.2.28	pVertReliability	288
8.313.2.29	pVertUnc	288
8.313.2.30	pXid	288
8.313.2.31	status	288
8.313.2.32	Tlvresult	288
8.314	unpack_loc_Delete_Assist_Data_t Struct Reference	288
8.314.1	Detailed Description	288
8.314.2	Field Documentation	289
8.314.2.1	Tlvresult	289
8.315	unpack_loc_EngineState_Ind_t Struct Reference	289
8.315.1	Detailed Description	289
8.315.2	Field Documentation	289
8.315.2.1	engineState	289
8.315.2.2	Tlvresult	289
8.316	unpack_loc_EventRegister_t Struct Reference	289
8.316.1	Detailed Description	289
8.316.2	Field Documentation	290
8.316.2.1	Tlvresult	290
8.317	unpack_loc_PositionRpt_Ind_t Struct Reference	290
8.317.1	Detailed Description	290
8.317.2	Field Documentation	294
8.317.2.1	pAltitudeAssumed	294
8.317.2.2	pAltitudeWrtEllipsoid	294
8.317.2.3	pAltitudeWrtMeanSeaLevel	294
8.317.2.4	pFixId	294
8.317.2.5	pGpsTime	294
8.317.2.6	pHeading	295

8.317.2.7 pHeadingUnc	295
8.317.2.8 pHorConfidence	295
8.317.2.9 pHorReliability	295
8.317.2.10pHorUncCircular	295
8.317.2.11pHorUncEllipseOrientAzimuth	295
8.317.2.12pHorUncEllipseSemiMajor	295
8.317.2.13pHorUncEllipseSemiMinor	295
8.317.2.14pLatitude	295
8.317.2.15pLeapSeconds	295
8.317.2.16pLongitude	295
8.317.2.17pMagneticDeviation	295
8.317.2.18pPrecisionDilution	295
8.317.2.19pSensorDataUsage	295
8.317.2.20pSpeedHorizontal	295
8.317.2.21pSpeedUnc	295
8.317.2.22pSpeedVertical	295
8.317.2.23pSvUsedforFix	295
8.317.2.24pTechnologyMask	295
8.317.2.25pTimeSrc	295
8.317.2.26pTimestampUtc	295
8.317.2.27pTimeUnc	295
8.317.2.28pVertConfidence	295
8.317.2.29pVertReliability	295
8.317.2.30pVertUnc	295
8.317.2.31sessionId	295
8.317.2.32sessionStatus	295
8.317.2.33Tlvresult	295
8.318unpack_loc_SetExtPowerConfig_Ind_t Struct Reference	296
8.318.1 Detailed Description	296
8.318.2 Field Documentation	296
8.318.2.1 status	296
8.318.2.2 Tlvresult	296
8.319unpack_loc_SetExtPowerState_t Struct Reference	296
8.319.1 Detailed Description	296
8.319.2 Field Documentation	297
8.319.2.1 Tlvresult	297
8.320unpack_loc_SetOperationMode_t Struct Reference	297
8.320.1 Detailed Description	297
8.320.2 Field Documentation	297
8.320.2.1 Tlvresult	297

8.321unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	297
8.321.1 Detailed Description	297
8.321.2 Field Documentation	297
8.321.2.1 Tlvresult	297
8.322unpack_loc_Start_t Struct Reference	297
8.322.1 Detailed Description	298
8.322.2 Field Documentation	298
8.322.2.1 Tlvresult	298
8.323unpack_loc_Stop_t Struct Reference	298
8.323.1 Detailed Description	298
8.323.2 Field Documentation	298
8.323.2.1 Tlvresult	298
8.324unpack_nas_GetCDMANetworkParameters_t Struct Reference	298
8.324.1 Detailed Description	298
8.324.2 Field Documentation	299
8.324.2.1 Application	299
8.324.2.2 Broadcast	299
8.324.2.3 CustomSCP	299
8.324.2.4 ForceRev0	299
8.324.2.5 Protocol	299
8.324.2.6 RegForeignNID	299
8.324.2.7 RegForeignSID	299
8.324.2.8 RegHomeSID	299
8.324.2.9 Roaming	299
8.324.2.10SCI	299
8.324.2.11SCM	299
8.325unpack_nas_GetHomeNetwork_t Struct Reference	299
8.325.1 Detailed Description	299
8.325.2 Field Documentation	300
8.325.2.1 mcc	300
8.325.2.2 mnc	300
8.325.2.3 name	300
8.325.2.4 nid	300
8.325.2.5 sid	300
8.326unpack_nas_GetNetworkPreference_t Struct Reference	300
8.326.1 Detailed Description	300
8.326.2 Field Documentation	301
8.326.2.1 ActiveTechPref	301
8.326.2.2 Duration	301
8.326.2.3 PersistentTechPref	301

8.326.2.4 Tlvresult	301
8.327unpack_nas_GetRFInfo_t Struct Reference	301
8.327.1 Detailed Description	301
8.327.2 Field Documentation	301
8.327.2.1 instancesSize	301
8.327.2.2 RFBandInfoElements	301
8.328unpack_nas_GetServingNetwork_t Struct Reference	301
8.328.1 Detailed Description	302
8.328.2 Field Documentation	302
8.328.2.1 CSDomain	302
8.328.2.2 DataCaps	302
8.328.2.3 DataCapsLen	302
8.328.2.4 MCC	302
8.328.2.5 MNC	302
8.328.2.6 Name	302
8.328.2.7 nameSize	302
8.328.2.8 PSDomain	302
8.328.2.9 Radiolfaces	302
8.328.2.10RadiolfacesSize	302
8.328.2.11RAN	302
8.328.2.12RegistrationState	302
8.328.2.13Roaming	302
8.329unpack_nas_GetServingNetworkCapabilities_t Struct Reference	302
8.329.1 Detailed Description	303
8.329.2 Field Documentation	303
8.329.2.1 DataCaps	303
8.329.2.2 DataCapsLen	303
8.330unpack_nas_GetSignalStrengths_t Struct Reference	303
8.330.1 Detailed Description	303
8.330.2 Field Documentation	303
8.330.2.1 len	303
8.330.2.2 radio	303
8.330.2.3 rssi	303
8.331unpack_nas_PerformNetworkScan_t Struct Reference	303
8.331.1 Detailed Description	303
8.331.2 Field Documentation	304
8.331.2.1 p3GppNetworkInfoInstances	304
8.331.2.2 p3GppNetworkInstanceSize	304
8.331.2.3 pPCSInstance	304
8.331.2.4 pPCSInstanceSize	304

8.331.2.5 pRATINstance	304
8.331.2.6 pRATInstanceSize	304
8.331.2.7 pScanResult	304
8.332unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	304
8.332.1 Detailed Description	304
8.332.2 Field Documentation	304
8.332.2.1 dataCaps	304
8.332.2.2 dataCapsSize	304
8.333unpack_nas_SetEventReportInd_t Struct Reference	304
8.333.1 Detailed Description	304
8.333.2 Field Documentation	305
8.333.2.1 RFTlv	305
8.333.2.2 RRTlv	305
8.333.2.3 SLQSSSTlv	305
8.333.2.4 SSTlv	305
8.334unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference	305
8.334.1 Detailed Description	305
8.334.2 Field Documentation	305
8.334.2.1 sPhyCaAggPcellInfo	305
8.334.2.2 sPhyCaAggScellIDBw	306
8.334.2.3 sPhyCaAggScellIndex	306
8.334.2.4 sPhyCaAggScellIndType	306
8.334.2.5 sPhyCaAggScellInfo	306
8.335unpack_nas_SetNetworkPreference_t Struct Reference	306
8.335.1 Detailed Description	306
8.335.2 Field Documentation	306
8.335.2.1 Tlvresult	306
8.336unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference	307
8.336.1 Detailed Description	307
8.336.2 Field Documentation	307
8.336.2.1 roaming	307
8.337unpack_nas_SetServingSystemCallback_ind_t Struct Reference	307
8.337.1 Detailed Description	307
8.337.2 Field Documentation	307
8.337.2.1 SSInfo	307
8.337.2.2 Tlvresult	307
8.338unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference	307
8.338.1 Detailed Description	308
8.338.2 Field Documentation	308
8.338.2.1 LTECphyCAInfo	308

8.338.2.2 Tlvresult	308
8.339unpack_nas_SLQSGetNetworkTime_t Struct Reference	308
8.339.1 Detailed Description	308
8.339.2 Field Documentation	308
8.339.2.1 p3GPP2TimeInfo	308
8.339.2.2 p3GPPTIMEInfo	308
8.340unpack_nas_SLQSGetPLMNName_t Struct Reference	308
8.340.1 Field Documentation	309
8.340.1.1 longName	309
8.340.1.2 longNameCI	309
8.340.1.3 longNameEn	309
8.340.1.4 longNameLen	309
8.340.1.5 longNameSB	309
8.340.1.6 shortName	309
8.340.1.7 shortNameCI	309
8.340.1.8 shortNameEn	309
8.340.1.9 shortNameLen	309
8.340.1.10shortNameSB	309
8.340.1.11spn	309
8.340.1.12spnEncoding	309
8.340.1.13spnLength	309
8.341unpack_nas_SLQSGetServingSystem_t Struct Reference	309
8.341.1 Detailed Description	310
8.341.2 Field Documentation	310
8.341.2.1 BasestationID	311
8.341.2.2 BasestationLatitude	311
8.341.2.3 BasestationLongitude	311
8.341.2.4 CallBarStatus	311
8.341.2.5 CDMA_P_Rev	311
8.341.2.6 CDMASystemInfoExt	311
8.341.2.7 CellID	311
8.341.2.8 ConcSvcInfo	311
8.341.2.9 CurrentPLMN	311
8.341.2.10DataSrvCapabilities	311
8.341.2.11DefaultRoamInd	311
8.341.2.12DetailedSvcInfo	311
8.341.2.13DTMInd	311
8.341.2.14Gpp2Timezone	311
8.341.2.15GppNetworkDSTAdjustment	311
8.341.2.16GppTimezone	311

8.341.2.17HdrPersonality	311
8.341.2.18Lac	311
8.341.2.19NetworkID	311
8.341.2.20PRLInd	311
8.341.2.21RoamIndicatorVal	311
8.341.2.22RoamingIndicatorList	311
8.341.2.23ServingSystem	311
8.341.2.24SystemID	311
8.341.2.25TrackAreaCode	311
8.342unpack_nas_SLQSGetSignalStrength_t Struct Reference	311
8.342.1 Detailed Description	312
8.342.2 Field Documentation	312
8.342.2.1 ecioList	312
8.342.2.2 ecioListLen	312
8.342.2.3 errorRateList	312
8.342.2.4 errorRateListLen	312
8.342.2.5 lo	312
8.342.2.6 ltersrp	312
8.342.2.7 ltesnr	312
8.342.2.8 rsrqInfo	312
8.342.2.9 rxSignalStrengthList	312
8.342.2.10rxSignalStrengthListLen	312
8.342.2.11signalStrengthReqMask	312
8.342.2.12sinr	312
8.343unpack_nas_SLQSGetSysInfo_t Struct Reference	313
8.343.1 Detailed Description	313
8.343.2 Field Documentation	315
8.343.2.1 pAddCDMASysInfo	315
8.343.2.2 pAddGSMSysInfo	315
8.343.2.3 pAddHDRSysInfo	315
8.343.2.4 pAddLTESysInfo	315
8.343.2.5 pAddWCDMASysInfo	315
8.343.2.6 pCDMASrvStatusInfo	315
8.343.2.7 pCDMASysInfo	315
8.343.2.8 pGSMCallBarringSysInfo	315
8.343.2.9 pGSMCipherDomainSysInfo	315
8.343.2.10pGSMSrvStatusInfo	315
8.343.2.11pGSMSysInfo	315
8.343.2.12pHDRSrvStatusInfo	315
8.343.2.13pHDRSysInfo	315

8.343.2.14	pLTESrvStatusInfo	315
8.343.2.15	pLTESysInfo	315
8.343.2.16	pLTEVoiceSupportSysInfo	315
8.343.2.17	pWCDMACallBarringSysInfo	315
8.343.2.18	pWCDMACipherDomainSysInfo	315
8.343.2.19	pWCDMASrvStatusInfo	315
8.343.2.20	pWCDMASysInfo	315
8.344	unpack_nas_SLQSGetSysSelectionPref_t Struct Reference	315
8.344.1	Detailed Description	316
8.344.2	Field Documentation	319
8.344.2.1	pBandPref	319
8.344.2.2	pEmerMode	319
8.344.2.3	pGWAcqOrderPref	319
8.344.2.4	pLTEBandPref	319
8.344.2.5	pModePref	319
8.344.2.6	pNetSelPref	319
8.344.2.7	pPRLPref	319
8.344.2.8	pRoamPref	319
8.344.2.9	pSrvDomainPref	319
8.345	unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference	319
8.345.1	Detailed Description	320
8.345.2	Field Documentation	320
8.345.2.1	pCDMAInfo	320
8.345.2.2	pGERANInfo	320
8.345.2.3	pLTEInfoInterfreq	320
8.345.2.4	pLTEInfoIntrafreq	320
8.345.2.5	pLTEInfoNeighboringGSM	320
8.345.2.6	pLTEInfoNeighboringWCDMA	320
8.345.2.7	pUMTSCellID	321
8.345.2.8	pUMTSInfo	321
8.345.2.9	pWCDMAInfoLTENeighborCell	321
8.346	unpack_nas_SLQSNasGetSigInfo_t Struct Reference	321
8.346.1	Detailed Description	321
8.346.2	Field Documentation	321
8.346.2.1	CDMASSInfo	321
8.346.2.2	GSMSSInfo	321
8.346.2.3	HDRSSInfo	321
8.346.2.4	LTSSInfo	321
8.346.2.5	WCDMASSInfo	321
8.347	unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference	321

8.347.1 Detailed Description	321
8.347.2 Field Documentation	322
8.347.2.1 pDayltSavAdj	322
8.347.2.2 pRadioInterface	322
8.347.2.3 pTimeZone	322
8.347.2.4 universalTime	322
8.348unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference	322
8.348.1 Detailed Description	322
8.348.2 Field Documentation	323
8.348.2.1 pCDMASigInfo	323
8.348.2.2 pGSMSigInfo	323
8.348.2.3 pHDRSigInfo	323
8.348.2.4 pLTESigInfo	323
8.348.2.5 pRscp	323
8.348.2.6 pTDSCDMASigInfoExt	323
8.348.2.7 pWCDMASigInfo	323
8.349unpack_nas_SLQSNasSwiModemStatus_t Struct Reference	323
8.349.1 Detailed Description	323
8.349.2 Field Documentation	323
8.349.2.1 commonInfo	323
8.349.2.2 pLTEInfo	323
8.350unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference	324
8.350.1 Detailed Description	324
8.350.2 Field Documentation	324
8.350.2.1 Info	324
8.350.2.2 Tlvresult	324
8.351unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference	324
8.351.1 Detailed Description	324
8.351.2 Field Documentation	324
8.351.2.1 Info	324
8.351.2.2 Tlvresult	324
8.352unpack_nas_SLQSSwiGetLteCQI_t Struct Reference	324
8.352.1 Detailed Description	325
8.352.2 Field Documentation	325
8.352.2.1 CQIValueCW0	325
8.352.2.2 CQIValueCW1	325
8.352.2.3 ValidityCW0	325
8.352.2.4 ValidityCW1	325
8.353unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference	325
8.353.1 Detailed Description	326

8.353.2 Field Documentation	327
8.353.2.1 pAddCDMASysInfo	327
8.353.2.2 pAddGSMSysInfo	327
8.353.2.3 pAddHDRSysInfo	327
8.353.2.4 pAddLTESysInfo	327
8.353.2.5 pAddWCDMASysInfo	328
8.353.2.6 pCDMASrvStatusInfo	328
8.353.2.7 pCDMASysInfo	328
8.353.2.8 pGSMCallBarringSysInfo	328
8.353.2.9 pGSMCipherDomainSysInfo	328
8.353.2.10pGSMSrvStatusInfo	328
8.353.2.11pGSMSysInfo	328
8.353.2.12pHDRSrvStatusInfo	328
8.353.2.13pHDRSysInfo	328
8.353.2.14pLTESrvStatusInfo	328
8.353.2.15pLTESysInfo	328
8.353.2.16pLTEVoiceSupportSysInfo	328
8.353.2.17pSysInfoNoChange	328
8.353.2.18pWCDMACallBarringSysInfo	328
8.353.2.19pWCDMACipherDomainSysInfo	328
8.353.2.20pWCDMASrvStatusInfo	328
8.353.2.21pWCDMASysInfo	328
8.354unpack_omaDmConfigTlv_t Struct Reference	328
8.354.1 Detailed Description	328
8.354.2 Field Documentation	329
8.354.2.1 alertmsg	329
8.354.2.2 alertmsglength	329
8.354.2.3 state	329
8.354.2.4 userInputReq	329
8.354.2.5 userInputTimeout	329
8.355unpack_omaDmFotaTlv_t Struct Reference	329
8.355.1 Detailed Description	329
8.355.2 Field Documentation	331
8.355.2.1 description	331
8.355.2.2 descriptionlength	331
8.355.2.3 fwdloadsize	331
8.355.2.4 fwloadComplete	331
8.355.2.5 namelength	331
8.355.2.6 package_name	331
8.355.2.7 sessionType	331

8.355.2.8 severity	331
8.355.2.9 state	331
8.355.2.10updateCompleteStatus	331
8.355.2.11userInputReq	331
8.355.2.12userInputTimeout	331
8.355.2.13version	331
8.355.2.14versionlength	331
8.356unpack_omaDmNotificationsTlv_t Struct Reference	331
8.356.1 Field Documentation	331
8.356.1.1 notification	331
8.356.1.2 sessionStatus	331
8.357unpack_qmi_t Struct Reference	331
8.357.1 Detailed Description	332
8.357.2 Field Documentation	332
8.357.2.1 msgid	332
8.357.2.2 type	332
8.357.2.3 xid	332
8.358unpack_qos_dataRate_t Struct Reference	332
8.358.1 Detailed Description	332
8.358.2 Field Documentation	332
8.358.2.1 dataRateMax	332
8.358.2.2 guaranteedRate	332
8.359unpack_qos_IPv4Addr_t Struct Reference	332
8.359.1 Detailed Description	332
8.359.2 Field Documentation	333
8.359.2.1 addr	333
8.359.2.2 subnetMask	333
8.360unpack_qos_IPv6Addr_t Struct Reference	333
8.360.1 Detailed Description	333
8.360.2 Field Documentation	333
8.360.2.1 addr	333
8.360.2.2 prefixLen	333
8.361unpack_qos_IPv6TrafCls_t Struct Reference	333
8.361.1 Detailed Description	333
8.361.2 Field Documentation	334
8.361.2.1 mask	334
8.361.2.2 val	334
8.362unpack_qos_pktErrRate_t Struct Reference	334
8.362.1 Detailed Description	334
8.362.2 Field Documentation	334

8.362.2.1 exponent	334
8.362.2.2 multiplier	334
8.363unpack_qos_Port_t Struct Reference	334
8.363.1 Detailed Description	334
8.363.2 Field Documentation	335
8.363.2.1 port	335
8.363.2.2 range	335
8.364unpack_qos_QosFlowInfo_t Struct Reference	335
8.364.1 Detailed Description	335
8.364.2 Field Documentation	336
8.364.2.1 BearerID	336
8.364.2.2 is_RxQFlowGranted_Available	336
8.364.2.3 is_TxQFlowGranted_Available	336
8.364.2.4 NumRxFilters	336
8.364.2.5 NumTxFilters	336
8.364.2.6 QFlowState	336
8.364.2.7 RxQFilter	336
8.364.2.8 RxQFlowGranted	336
8.364.2.9 TxQFilter	336
8.364.2.10TxQFlowGranted	336
8.365unpack_qos_QosFlowInfoState_t Struct Reference	336
8.365.1 Detailed Description	336
8.365.2 Field Documentation	337
8.365.2.1 id	337
8.365.2.2 isNewFlow	337
8.365.2.3 state	337
8.366unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference	337
8.366.1 Detailed Description	337
8.366.2 Field Documentation	337
8.366.2.1 NWQoSStatus	337
8.367unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference	337
8.367.1 Detailed Description	338
8.367.2 Field Documentation	338
8.367.2.1 ambr_dl	338
8.367.2.2 ambr_dl_ext	338
8.367.2.3 ambr_dl_ext2	338
8.367.2.4 ambr_ul	338
8.367.2.5 ambr_ul_ext	338
8.367.2.6 ambr_ul_ext2	338
8.367.2.7 apnId	339

8.368unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference	339
8.368.1 Detailed Description	339
8.368.2 Field Documentation	340
8.368.2.1 apnId	340
8.368.2.2 numQosFlow	340
8.368.2.3 qosFlow	340
8.368.2.4 total_rx_bytes	340
8.368.2.5 total_rx_pkt	340
8.368.2.6 total_tx_bytes	340
8.368.2.7 total_tx_bytes_drp	340
8.368.2.8 total_tx_pkt	340
8.368.2.9 total_tx_pkt_drp	340
8.369unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference	340
8.369.1 Detailed Description	340
8.369.2 Field Documentation	340
8.369.2.1 NumFlows	340
8.369.2.2 QosFlowInfo	340
8.370unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference	340
8.370.1 Detailed Description	341
8.370.2 Field Documentation	341
8.370.2.1 status	341
8.371unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference	341
8.371.1 Detailed Description	341
8.371.2 Field Documentation	341
8.371.2.1 event	341
8.372unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference	341
8.372.1 Detailed Description	341
8.372.2 Field Documentation	342
8.372.2.1 event	342
8.372.2.2 id	342
8.372.2.3 reason	342
8.372.2.4 status	342
8.373unpack_qos_swiQosFilter_t Struct Reference	343
8.373.1 Detailed Description	343
8.373.2 Field Documentation	345
8.373.2.1 EspSpi	345
8.373.2.2 Id	345
8.373.2.3 index	345
8.373.2.4 IPv4DstAddr	345
8.373.2.5 IPv4SrcAddr	345

8.373.2.6 IPv4Tos	345
8.373.2.7 IPv6DstAddr	345
8.373.2.8 IPv6Label	345
8.373.2.9 IPv6SrcAddr	345
8.373.2.10 IPv6TrafCls	345
8.373.2.11 is_EspSpi_Available	345
8.373.2.12 is_Id_Available	345
8.373.2.13 is_IPv4DstAddr_Available	345
8.373.2.14 is_IPv4SrcAddr_Available	345
8.373.2.15 is_IPv4Tos_Available	345
8.373.2.16 is_IPv6DstAddr_Available	345
8.373.2.17 is_IPv6Label_Available	345
8.373.2.18 is_IPv6SrcAddr_Available	345
8.373.2.19 is_IPv6TrafCls_Available	345
8.373.2.20 is_NxtHdrProto_Available	345
8.373.2.21 is_Precedence_Available	345
8.373.2.22 is_TCPDstPort_Available	345
8.373.2.23 is_TCPSrcPort_Available	346
8.373.2.24 is_TranDstPort_Available	346
8.373.2.25 is_TranSrcPort_Available	346
8.373.2.26 is_UDPDstPort_Available	346
8.373.2.27 is_UDPSrcPort_Available	346
8.373.2.28 NxtHdrProto	346
8.373.2.29 Precedence	346
8.373.2.30 TCPDstPort	346
8.373.2.31 TCPSrcPort	346
8.373.2.32 TranDstPort	346
8.373.2.33 TranSrcPort	346
8.373.2.34 UDPDstPort	346
8.373.2.35 UDPSrcPort	346
8.373.2.36 version	346
8.374unpack_qos_swiQosFlow_t Struct Reference	346
8.374.1 Detailed Description	347
8.374.2 Field Documentation	349
8.374.2.1 DataRate	349
8.374.2.2 index	349
8.374.2.3 is_DataRate_Available	349
8.374.2.4 is_Jitter_Available	349
8.374.2.5 is_Latency_Available	349
8.374.2.6 is_LteQci_Available	349

8.374.2.7 is_MaxAllowedPktSz_Available	349
8.374.2.8 is_MinPolicedPktSz_Available	349
8.374.2.9 is_PktErrRate_Available	349
8.374.2.10s_ProfileId3GPP2_Available	349
8.374.2.11is_TokenBucket_Available	349
8.374.2.12s_TrafficClass_Available	349
8.374.2.13s_val_3GPP2Pri_Available	349
8.374.2.14s_val_3GPPImCn_Available	349
8.374.2.15s_val_3GPPResResidualBER_Available	349
8.374.2.16s_val_3GPPSigInd_Available	349
8.374.2.17s_val_3GPPTraHdlPri_Available	349
8.374.2.18Jitter	349
8.374.2.19Latency	350
8.374.2.20LteQci	350
8.374.2.21MaxAllowedPktSz	350
8.374.2.22MinPolicedPktSz	350
8.374.2.23PktErrRate	350
8.374.2.24ProfileId3GPP2	350
8.374.2.25TokenBucket	350
8.374.2.26TrafficClass	350
8.374.2.27val_3GPP2Pri	350
8.374.2.28val_3GPPImCn	350
8.374.2.29val_3GPPResResidualBER	350
8.374.2.30val_3GPPSigInd	350
8.374.2.31val_3GPPTraHdlPri	350
8.375unpack_qos_tokenBucket_t Struct Reference	350
8.375.1 Detailed Description	350
8.375.2 Field Documentation	350
8.375.2.1 bucketSz	350
8.375.2.2 peakRate	350
8.375.2.3 tokenRate	350
8.376unpack_qos_Tos_t Struct Reference	351
8.376.1 Detailed Description	351
8.376.2 Field Documentation	351
8.376.2.1 mask	351
8.376.2.2 val	351
8.377unpack_QosFlowStat_t Struct Reference	351
8.377.1 Detailed Description	351
8.377.2 Field Documentation	352
8.377.2.1 bearerId	352

8.377.2.2 tx_bytes	352
8.377.2.3 tx_bytes_drp	352
8.377.2.4 tx_pkt	352
8.377.2.5 tx_pkt_drp	352
8.378unpack_sms_SendSMS_t Struct Reference	352
8.378.1 Detailed Description	352
8.378.2 Field Documentation	352
8.378.2.1 messageFailureCode	352
8.378.2.2 messageID	352
8.379unpack_sms_SetNewSMSCallback_ind_t Struct Reference	352
8.379.1 Detailed Description	353
8.379.2 Field Documentation	353
8.379.2.1 ETWSPLMNTlv	353
8.379.2.2 ETWSTlv	353
8.379.2.3 IMSTlv	353
8.379.2.4 MMTlv	353
8.379.2.5 NewMMTlv	353
8.379.2.6 SMSCTlv	353
8.379.2.7 TRMessageTlv	353
8.380unpack_sms_SetNewSMSCallback_t Struct Reference	353
8.381unpack_sms_SLQSDeleteSMS_t Struct Reference	354
8.382unpack_sms_SLQSGetSMS_t Struct Reference	354
8.382.1 Detailed Description	354
8.382.2 Field Documentation	354
8.382.2.1 message	354
8.382.2.2 messageFormat	354
8.382.2.3 messageSize	354
8.382.2.4 messageTag	354
8.383unpack_sms_SLQSGetSMSList_t Struct Reference	354
8.383.1 Detailed Description	355
8.383.2 Field Documentation	355
8.383.2.1 messageList	355
8.383.2.2 messageListSize	355
8.384unpack_sms_SLQSModifySMSStatus_t Struct Reference	355
8.385unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference	355
8.385.1 Detailed Description	355
8.385.2 Field Documentation	355
8.385.2.1 messageMode	355
8.385.2.2 storageType	356
8.386unpack_swiloc_SwiLocGetAutoStart_t Struct Reference	356

8.386.1 Detailed Description	356
8.386.2 Field Documentation	357
8.386.2.1 fix_rate	357
8.386.2.2 fix_rate_reported	357
8.386.2.3 fix_type	357
8.386.2.4 fix_type_reported	357
8.386.2.5 function	357
8.386.2.6 function_reported	357
8.386.2.7 max_dist	357
8.386.2.8 max_dist_reported	357
8.386.2.9 max_time	357
8.386.2.10max_time_reported	357
8.387unpack_swioma_SLQSOMADMAAlertCallback_ind_t Struct Reference	357
8.387.1 Detailed Description	357
8.387.2 Field Documentation	358
8.387.2.1 eventType	358
8.387.2.2 SessionInfoConfig	358
8.387.2.3 SessionInfoFota	358
8.387.2.4 SessionInfoNotification	358
8.388unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference	358
8.388.1 Detailed Description	358
8.388.2 Field Documentation	360
8.388.2.1 Date	360
8.388.2.2 DateLength	360
8.388.2.3 PkgDescLength	360
8.388.2.4 PkgDescription	360
8.388.2.5 PkgName	360
8.388.2.6 PkgNameLength	360
8.388.2.7 RetryCount	360
8.388.2.8 SessionState	360
8.388.2.9 SessionType	360
8.388.2.10Severity	360
8.388.2.11Source	360
8.388.2.12SourceLength	360
8.388.2.13Status	361
8.388.2.14Time	361
8.388.2.15TimeLength	361
8.388.2.16UpdateCompleteStatus	361
8.389unpack_swioma_SLQSOMADMGetSettings_t Struct Reference	361
8.389.1 Detailed Description	361

8.389.2 Field Documentation	362
8.389.2.1 Autosdm	362
8.389.2.2 FOTAdownload	362
8.389.2.3 FOTAUpdate	362
8.389.2.4 FwAutoCheck	362
8.389.2.5 OMADMEEnabled	362
8.390unpack_swioma_SLQSOMADMStartSession_t Struct Reference	362
8.390.1 Detailed Description	362
8.390.2 Field Documentation	362
8.390.2.1 FwAvailability	363
8.391unpack_uim_ChangePin_t Struct Reference	363
8.391.1 Detailed Description	363
8.391.2 Field Documentation	363
8.391.2.1 pEncryptedPIN1	363
8.391.2.2 pIndicationToken	363
8.391.2.3 pRemainingRetries	363
8.391.2.4 Tlvresult	363
8.392unpack_uim_GetCardStatus_t Struct Reference	363
8.392.1 Detailed Description	364
8.392.2 Field Documentation	364
8.392.2.1 pCardStatus	364
8.392.2.2 pHotSwapStatus	364
8.392.2.3 Tlvresult	364
8.393unpack_uim_ReadTransparent_t Struct Reference	364
8.393.1 Detailed Description	364
8.393.2 Field Documentation	365
8.393.2.1 pCardResult	365
8.393.2.2 pEncryptedData	365
8.393.2.3 pIndicationToken	365
8.393.2.4 pReadResult	365
8.393.2.5 Tlvresult	365
8.394unpack_uim_SetPinProtection_t Struct Reference	365
8.394.1 Detailed Description	365
8.394.2 Field Documentation	365
8.394.2.1 pEncryptedPIN1	365
8.394.2.2 pIndicationToken	365
8.394.2.3 pRemainingRetries	365
8.394.2.4 Tlvresult	366
8.395unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference	366
8.395.1 Detailed Description	366

8.395.2 Field Documentation	366
8.395.2.1 bNumberOfPhySlots	366
8.395.2.2 slotsstatusChange	366
8.396unpack_uim_SLQSUIEventRegister_t Struct Reference	366
8.396.1 Detailed Description	366
8.396.2 Field Documentation	366
8.396.2.1 eventMask	366
8.397unpack_uim_SLQSUIGetSlotsStatus_t Struct Reference	366
8.397.1 Detailed Description	367
8.397.2 Field Documentation	367
8.397.2.1 pNumberOfPhySlot	367
8.397.2.2 pUimSlotsStatus	367
8.398unpack_uim_SLQSUISetStatusChangeCallBack_ind_t Struct Reference	367
8.398.1 Detailed Description	367
8.398.2 Field Documentation	367
8.398.2.1 pCardStatus	367
8.399unpack_uim_UnblockPin_t Struct Reference	367
8.399.1 Detailed Description	368
8.399.2 Field Documentation	368
8.399.2.1 pEncryptedPIN1	368
8.399.2.2 pIndicationToken	368
8.399.2.3 pRemainingRetries	368
8.399.2.4 Tlvresult	368
8.400unpack_uim_VerifyPin_t Struct Reference	368
8.400.1 Detailed Description	368
8.400.2 Field Documentation	369
8.400.2.1 pEncryptedPIN1	369
8.400.2.2 pIndicationToken	369
8.400.2.3 pRemainingRetries	369
8.400.2.4 Tlvresult	369
8.401unpack_wds_GetConnectionRate_t Struct Reference	369
8.401.1 Detailed Description	369
8.401.2 Field Documentation	369
8.401.2.1 currentChannelRXRate	369
8.401.2.2 currentChannelTXRate	369
8.401.2.3 maxChannelRXRate	369
8.401.2.4 maxChannelTXRate	369
8.402unpack_wds_GetDefaultProfile_t Struct Reference	369
8.402.1 Detailed Description	370
8.402.2 Field Documentation	370

8.402.2.1 apnname	370
8.402.2.2 apnsize	370
8.402.2.3 auth	370
8.402.2.4 ipaddr	370
8.402.2.5 ipaddrv6	370
8.402.2.6 name	370
8.402.2.7 namesize	370
8.402.2.8 pdptype	370
8.402.2.9 pridns	370
8.402.2.10pridnsv6	370
8.402.2.11secdns	370
8.402.2.12secdnsv6	370
8.402.2.13username	371
8.402.2.14usersize	371
8.403unpack_wds_GetDefaultProfileNum_t Struct Reference	371
8.403.1 Detailed Description	371
8.403.2 Field Documentation	371
8.403.2.1 index	371
8.404unpack_wds_GetDormancyState_t Struct Reference	371
8.404.1 Detailed Description	371
8.404.2 Field Documentation	371
8.404.2.1 dormancyState	371
8.405unpack_wds_GetLastMobileIPError_t Struct Reference	371
8.405.1 Detailed Description	371
8.405.2 Field Documentation	371
8.405.2.1 error	371
8.406unpack_wds_GetMobileIP_t Struct Reference	372
8.406.1 Detailed Description	372
8.406.2 Field Documentation	372
8.406.2.1 mipMode	372
8.407unpack_wds_GetMobileIPProfile_t Struct Reference	372
8.407.1 Detailed Description	372
8.407.2 Field Documentation	372
8.407.2.1 AAASPI	372
8.407.2.2 AAASState	373
8.407.2.3 address	373
8.407.2.4 enabled	373
8.407.2.5 HASPI	373
8.407.2.6 HASState	373
8.407.2.7 NAI	373

8.407.2.8 naiSize	373
8.407.2.9 primaryHA	373
8.407.2.10 revTunneling	373
8.407.2.11 secondaryHA	373
8.408unpack_wds_GetPacketStatus_t Struct Reference	373
8.408.1 Detailed Description	373
8.408.2 Field Documentation	374
8.408.2.1 rXDroppedCount	374
8.408.2.2 rXOkBytesCount	374
8.408.2.3 rXOKBytesLastCall	374
8.408.2.4 rXPacketErrors	374
8.408.2.5 rXPacketOverflows	374
8.408.2.6 rXPacketSuccesses	374
8.408.2.7 tXDroppedCount	374
8.408.2.8 tXOkBytesCount	374
8.408.2.9 tXOKBytesLastCall	374
8.408.2.10 tXPacketErrors	374
8.408.2.11 tXPacketOverflows	374
8.408.2.12 tXPacketSuccesses	374
8.409unpack_wds_GetSessionDuration_t Struct Reference	374
8.409.1 Detailed Description	374
8.409.2 Field Documentation	374
8.409.2.1 callDuration	374
8.410unpack_wds_GetSessionState_t Struct Reference	374
8.410.1 Detailed Description	375
8.410.2 Field Documentation	375
8.410.2.1 connectionStatus	375
8.411unpack_wds_RMSetTransferStatistics_t Struct Reference	375
8.412unpack_wds_SetMobileIPProfile_t Struct Reference	375
8.413unpack_wds_SLQSCreateProfile_t Struct Reference	375
8.413.1 Detailed Description	375
8.413.2 Field Documentation	375
8.413.2.1 pCreateProfileOut	375
8.413.2.2 pProfileID	375
8.413.2.3 Tlvresult	375
8.414unpack_wds_SLQSDeleteProfile_t Struct Reference	375
8.414.1 Detailed Description	375
8.414.2 Field Documentation	376
8.414.2.1 extendedErrorCode	376
8.415unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference	376

8.415.1 Detailed Description	376
8.415.2 Field Documentation	376
8.415.2.1 _3gppRelease	376
8.415.2.2 defaultPDNEnabled	376
8.415.2.3 LTEAttachProfileList	377
8.415.2.4 LTEAttachProfileListLen	377
8.415.2.5 profileList	377
8.416unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	377
8.416.1 Detailed Description	377
8.416.2 Field Documentation	377
8.416.2.1 currNetworkInfo	377
8.416.2.2 networkInfoLen	377
8.416.2.3 prefNetwork	377
8.417unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference	377
8.417.1 Detailed Description	377
8.417.2 Field Documentation	377
8.417.2.1 curDataBearerTechnology	377
8.417.2.2 dataBearerMask	378
8.417.2.3 lastCallDataBearerTechnology	378
8.418unpack_wds_SLQSGetDUNCallInfo_t Struct Reference	378
8.418.1 Detailed Description	378
8.418.2 Field Documentation	378
8.418.2.1 callEndReason	378
8.418.2.2 channelRate	378
8.418.2.3 connectionStatus	378
8.418.2.4 dataBearerTech	378
8.418.2.5 dormancyStatus	378
8.418.2.6 lastCallDataBearerTech	379
8.418.2.7 lastCallRXOKBytesCnt	379
8.418.2.8 lastCallTXOKBytesCnt	379
8.418.2.9 mdmCallDurationActive	379
8.418.2.10rxOKBytesCount	379
8.418.2.11txOKBytesCount	379
8.419unpack_wds_SLQSGetProfileSettings_t Struct Reference	379
8.419.1 Field Documentation	379
8.419.1.1 pProfileSettings	379
8.419.1.2 ProfileType	379
8.419.1.3 Tlvresult	379
8.420unpack_wds_SLQSGetRuntimeSettings_t Struct Reference	379
8.420.1 Detailed Description	380

8.420.2 Field Documentation	380
8.420.2.1 APNName	380
8.420.2.2 Authentication	380
8.420.2.3 DomainList	380
8.420.2.4 GPRSGrantedQoS	380
8.420.2.5 GWAddressV4	380
8.420.2.6 IMCNflag	380
8.420.2.7 IPFamilyPreference	380
8.420.2.8 IPv4	381
8.420.2.9 IPV6AddrInfo	381
8.420.2.10 IPV6GWAddrInfo	381
8.420.2.11 Mtu	381
8.420.2.12 PCSCFAddrPCO	381
8.420.2.13 PCSCFFQDNAddrList	381
8.420.2.14 PDPTType	381
8.420.2.15 PrimaryDNSV4	381
8.420.2.16 PrimaryDNSV6	381
8.420.2.17 ProfileID	381
8.420.2.18 ProfileName	381
8.420.2.19 SecondaryDNSV4	381
8.420.2.20 SecondaryDNSV6	381
8.420.2.21 ServerAddrList	381
8.420.2.22 SubnetMaskV4	381
8.420.2.23 Technology	381
8.420.2.24 UMTSGrantedQoS	381
8.420.2.25 Username	381
8.421unpack_wds_SLQSMModifyProfile_t Struct Reference	381
8.421.1 Detailed Description	381
8.421.2 Field Documentation	381
8.421.2.1 pExtErrorCode	381
8.422unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference	381
8.422.1 Detailed Description	382
8.422.2 Field Documentation	382
8.422.2.1 Tlvresult	382
8.423unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference	382
8.423.1 Detailed Description	382
8.423.2 Field Documentation	382
8.423.2.1 bearerID	382
8.423.2.2 conn_status	382
8.423.2.3 ipFamily	383

8.423.2.4 reconfigReqd	383
8.423.2.5 sessionEndReason	383
8.423.2.6 techName	383
8.423.2.7 verboseSessnEndReason	383
8.423.2.8 verboseSessnEndReasonType	383
8.424unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference	383
8.424.1 Detailed Description	383
8.424.2 Field Documentation	384
8.424.2.1 currDBTechAvail	384
8.424.2.2 currNWInfo	384
8.424.2.3 dataSysStatAvail	384
8.424.2.4 dBTechAvail	384
8.424.2.5 dBTechnology	384
8.424.2.6 dormancyStatAvail	384
8.424.2.7 dormancyStatus	384
8.424.2.8 mipstatAvail	384
8.424.2.9 mipStatus	384
8.424.2.10netInfoLen	384
8.424.2.11prefNetwork	384
8.424.2.12atMask	384
8.424.2.13rx_bytes	384
8.424.2.14rx_pkts	384
8.424.2.15soMask	384
8.424.2.16tx_bytes	384
8.424.2.17tx_pkts	384
8.424.2.18xferStatAvail	384
8.425unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	384
8.425.1 Detailed Description	384
8.425.2 Field Documentation	384
8.425.2.1 pHwConfig	385
8.425.2.2 pRequestOptionList	385
8.426unpack_wds_SLQSSStartDataSession_t Struct Reference	385
8.426.1 Detailed Description	385
8.426.2 Field Documentation	385
8.426.2.1 pFailureReason	385
8.426.2.2 psid	385
8.426.2.3 pVerboseFailReasonType	385
8.426.2.4 pVerboseFailureReason	385
8.427unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	385
8.427.1 Detailed Description	386

8.427.2 Field Documentation	386
8.427.2.1 apnName	386
8.427.2.2 bearerId	386
8.427.2.3 contextId	386
8.427.2.4 ipv4Address	386
8.427.2.5 ipv4GWAddress	386
8.427.2.6 ipv6Address	386
8.427.2.7 ipv6GWAddress	386
8.427.2.8 prDNSIPv4Address	386
8.427.2.9 prDNSIPv6Address	387
8.427.2.10prPCSCFIPv4Address	387
8.427.2.11prPCSCFIPv6Address	387
8.427.2.12seDNSIPv4Address	387
8.427.2.13seDNSIPv6Address	387
8.427.2.14sePCSCFIPv4Address	387
8.427.2.15sePCSCFIPv6Address	387
8.428UnPackGetProfileSettingOut Struct Reference	387
8.428.1 Field Documentation	387
8.428.1.1 curProfile	387
8.428.1.2 pExtErrCode	387
8.429unpackWdsProfileParam Union Reference	387
8.429.1 Field Documentation	387
8.429.1.1 SlqsProfile3GPP	387
8.429.1.2 SlqsProfile3GPP2	387
8.430wds_currNetworkInfo Struct Reference	387
8.430.1 Detailed Description	387
8.430.2 Field Documentation	389
8.430.2.1 NetworkType	389
8.430.2.2 RATMask	389
8.430.2.3 SOMask	389
8.431wds_Domain Struct Reference	389
8.431.1 Detailed Description	389
8.431.2 Field Documentation	390
8.431.2.1 domainLen	390
8.431.2.2 domainName	390
8.432wds_DomainNameList Struct Reference	390
8.432.1 Detailed Description	390
8.432.2 Field Documentation	390
8.432.2.1 domain	390
8.432.2.2 numInstances	390

8.433wds_GPRSQoS Struct Reference	390
8.433.1 Detailed Description	390
8.433.2 Field Documentation	391
8.433.2.1 delayClass	391
8.433.2.2 meanThroughputClass	391
8.433.2.3 peakThroughputClass	391
8.433.2.4 precedenceClass	391
8.433.2.5 reliabilityClass	391
8.434wds_IPV6AddressInfo Struct Reference	391
8.434.1 Detailed Description	391
8.434.2 Field Documentation	391
8.434.2.1 IPAddressV6	391
8.434.2.2 IPV6PrefixLen	391
8.435wds_IPV6GWAddressInfo Struct Reference	391
8.435.1 Detailed Description	392
8.435.2 Field Documentation	392
8.435.2.1 gwAddressV6	392
8.435.2.2 gwV6PrefixLen	392
8.436wds_PCSCFFQDNAddress Struct Reference	392
8.436.1 Detailed Description	392
8.436.2 Field Documentation	392
8.436.2.1 fqdnAddr	392
8.436.2.2 fqdnLen	392
8.437wds_PCSCFFQDNAddressList Struct Reference	392
8.437.1 Detailed Description	393
8.437.2 Field Documentation	393
8.437.2.1 numInstances	393
8.437.2.2 pcsfFQDNAddress	393
8.438wds_PCSCFIPv4ServerAddressList Struct Reference	393
8.438.1 Detailed Description	393
8.438.2 Field Documentation	393
8.438.2.1 numInstances	393
8.438.2.2 pscsfIPv4Addr	393
8.439wds_ProfileIdentifier Struct Reference	393
8.439.1 Detailed Description	394
8.439.2 Field Documentation	394
8.439.2.1 profileIndex	394
8.439.2.2 profileType	394
8.440wds_profileInfo Union Reference	394
8.440.1 Detailed Description	394

8.440.2 Field Documentation	394
8.440.2.1 SlqsProfile3GPP	394
8.440.2.2 SlqsProfile3GPP2	394
8.441 wds_UMTSMInQoS Struct Reference	394
8.441.1 Detailed Description	395
8.441.2 Field Documentation	396
8.441.2.1 deliveryErrSDU	396
8.441.2.2 grntDownlinkBitrate	396
8.441.2.3 grntUplinkBitrate	396
8.441.2.4 maxDownlinkBitrate	396
8.441.2.5 maxSDUSize	396
8.441.2.6 maxUplinkBitrate	396
8.441.2.7 qosDeliveryOrder	396
8.441.2.8 resBerRatio	397
8.441.2.9 sduErrorRatio	397
8.441.2.10 trafficClass	397
8.441.2.11 trafficPriority	397
8.441.2.12 transferDelay	397
8.442 wdsDhcpv4HwConfig Struct Reference	397
8.442.1 Detailed Description	397
8.442.2 Field Documentation	397
8.442.2.1 chaddr	397
8.442.2.2 chaddrLen	397
8.442.2.3 hwType	397
8.443 wdsDhcpv4Option Struct Reference	397
8.443.1 Detailed Description	397
8.443.2 Field Documentation	398
8.443.2.1 optCode	398
8.443.2.2 optVal	398
8.443.2.3 optValLen	398
8.444 wdsDhcpv4OptionList Struct Reference	398
8.444.1 Detailed Description	398
8.444.2 Field Documentation	398
8.444.2.1 numOpt	398
8.444.2.2 pOptList	398
8.445 wdsDhcpv4ProfileId Struct Reference	398
8.445.1 Detailed Description	398
8.445.2 Field Documentation	399
8.445.2.1 profileId	399
8.445.2.2 profileType	399

9	File Documentation	401
9.1	apdoxypages.c File Reference	401
9.1.1	Detailed Description	401
9.2	common.h File Reference	401
9.2.1	Macro Definition Documentation	403
9.2.1.1	DEAULT_LOC_TIMEOUT_IN_SEC	403
9.2.1.2	MINREQBKLEN	403
9.2.1.3	MSGID_AND_LEN	403
9.2.1.4	MSGID_DONT_CARE	403
9.2.1.5	SDK_VALIDATE_INPUT_PACK_PARAM	403
9.2.1.6	SDU_HDR_LEN	403
9.2.1.7	UNUSEDPARAM	403
9.2.2	Typedef Documentation	403
9.2.2.1	logger	403
9.2.3	Enumeration Type Documentation	403
9.2.3.1	eLOG_LEVEL	403
9.2.3.2	eQMI_SVC	403
9.2.3.3	eTimeout	404
9.2.3.4	msgtype	404
9.2.4	Function Documentation	404
9.2.4.1	fill_pack_ctx	404
9.2.4.2	fill_sdu_hdr	404
9.2.4.3	get_version	404
9.2.4.4	helper_get_resp_ctx	404
9.2.4.5	helper_get_xid	405
9.2.4.6	helper_set_log_func	405
9.2.4.7	helper_set_log_lvl	405
9.2.4.8	libpack_GetVersion	405
9.2.4.9	libpack_log	405
9.2.4.10	unpack_result_code_only	405
9.2.5	Variable Documentation	405
9.2.5.1	glog	405
9.2.5.2	gloglvl	405
9.3	dms.h File Reference	405
9.3.1	Macro Definition Documentation	409
9.3.1.1	DMS_IMGDETAILS_LEN	409
9.3.1.2	DMS_MAX_CUST_ID_LEN	409
9.3.1.3	DMS_MAX_CUST_VALUE_LEN	410
9.3.1.4	DMS_MAX_FWUPDATE_LOG_STR_SZ	410
9.3.1.5	DMS_MAX_FWUPDATE_REF_STR_SZ	410

9.3.1.6	DMS_PM_FACTORY	410
9.3.1.7	DMS_PM_LOW	410
9.3.1.8	DMS_PM_OFFLINE	410
9.3.1.9	DMS_PM_ONLINE	410
9.3.1.10	DMS_PM_PERSISTENT_LOW	410
9.3.1.11	DMS_PM_RESET	410
9.3.1.12	DMS_PM_SHUT_DOWN	410
9.3.1.13	DMS_SET_REPORT_DISABLE	410
9.3.1.14	DMS_SET_REPORT_ENABLE	410
9.3.1.15	DMS_SLQSFWINFO_APPVERSION_SZ	410
9.3.1.16	DMS_SLQSFWINFO_BOOTVERSION_SZ	410
9.3.1.17	DMS_SLQSFWINFO_CARRIER_SZ	410
9.3.1.18	DMS_SLQSFWINFO_CUR_CARR_NAME	410
9.3.1.19	DMS_SLQSFWINFO_CUR_CARR_REV	410
9.3.1.20	DMS_SLQSFWINFO_MODELID_SZ	410
9.3.1.21	DMS_SLQSFWINFO_PACKAGEID_SZ	410
9.3.1.22	DMS_SLQSFWINFO_PRIVERSION_SZ	410
9.3.1.23	DMS_SLQSFWINFO_SKU_SZ	410
9.3.1.24	DMS_SWI_SET_IND_DISABLE	410
9.3.1.25	DMS_SWI_SET_IND_ENABLE	410
9.3.1.26	DMS_UINT8_MAX_STRING_SZ	410
9.3.1.27	MAX_BUILD_ID_LEN	410
9.3.1.28	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	410
9.3.1.29	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	410
9.3.1.30	UNIQUE_ID_LEN	410
9.3.2	Function Documentation	411
9.3.2.1	pack_dms_GetActivationState	411
9.3.2.2	pack_dms_GetBandCapability	411
9.3.2.3	pack_dms_GetCrashAction	411
9.3.2.4	pack_dms_GetCustFeature	412
9.3.2.5	pack_dms_GetCustFeaturesV2	412
9.3.2.6	pack_dms_GetDeviceCap	412
9.3.2.7	pack_dms_GetDeviceCapabilities	412
9.3.2.8	pack_dms_GetDeviceHardwareRev	413
9.3.2.9	pack_dms_GetDeviceMfr	413
9.3.2.10	pack_dms_GetDeviceSerialNumbers	413
9.3.2.11	pack_dms_GetFirmwareInfo	414
9.3.2.12	pack_dms_GetFirmwareRevision	414
9.3.2.13	pack_dms_GetFirmwareRevisions	414
9.3.2.14	pack_dms_GetFSN	415

9.3.2.15	pack_dms_GetHardwareRevision	415
9.3.2.16	pack_dms_GetIMSI	415
9.3.2.17	pack_dms_GetModelID	416
9.3.2.18	pack_dms_GetNetworkTime	416
9.3.2.19	pack_dms_GetPower	417
9.3.2.20	pack_dms_GetPRLVersion	417
9.3.2.21	pack_dms_GetSerialNumbers	417
9.3.2.22	pack_dms_GetUSBComp	418
9.3.2.23	pack_dms_GetVoiceNumber	418
9.3.2.24	pack_dms_SetCrashAction	418
9.3.2.25	pack_dms_SetCustFeature	419
9.3.2.26	pack_dms_SetCustFeaturesV2	419
9.3.2.27	pack_dms_SetEventReport	420
9.3.2.28	pack_dms_SetFirmwarePreference	420
9.3.2.29	pack_dms_SetPower	420
9.3.2.30	pack_dms_SetUSBComp	421
9.3.2.31	pack_dms_SLQSDmsSwiGetResetInfo	421
9.3.2.32	pack_dms_SLQSDmsSwiIndicationRegister	422
9.3.2.33	pack_dms_SLQSGetBandCapability	422
9.3.2.34	pack_dms_SLQSSwiClearDyingGaspStatistics	422
9.3.2.35	pack_dms_SLQSSwiGetDyingGaspCfg	423
9.3.2.36	pack_dms_SLQSSwiGetDyingGaspStatistics	423
9.3.2.37	pack_dms_SLQSSwiGetFirmwareCurr	423
9.3.2.38	pack_dms_SLQSSwiGetFwUpdateStatus	424
9.3.2.39	pack_dms_SLQSSwiSetDyingGaspCfg	424
9.3.2.40	pack_dms_UIMGetICCID	424
9.3.2.41	unpack_dms_GetActivationState	425
9.3.2.42	unpack_dms_GetBandCapability	425
9.3.2.43	unpack_dms_GetCrashAction	426
9.3.2.44	unpack_dms_GetCustFeature	426
9.3.2.45	unpack_dms_GetCustFeaturesV2	426
9.3.2.46	unpack_dms_GetDeviceCap	426
9.3.2.47	unpack_dms_GetDeviceCapabilities	427
9.3.2.48	unpack_dms_GetDeviceHardwareRev	427
9.3.2.49	unpack_dms_GetDeviceMfr	427
9.3.2.50	unpack_dms_GetDeviceSerialNumbers	428
9.3.2.51	unpack_dms_GetFirmwareInfo	428
9.3.2.52	unpack_dms_GetFirmwareRevision	428
9.3.2.53	unpack_dms_GetFirmwareRevisions	429
9.3.2.54	unpack_dms_GetFSN	429

9.3.2.55	unpack_dms_GetHardwareRevision	429
9.3.2.56	unpack_dms_GetIMSI	430
9.3.2.57	unpack_dms_GetModelID	430
9.3.2.58	unpack_dms_GetNetworkTime	430
9.3.2.59	unpack_dms_GetPower	431
9.3.2.60	unpack_dms_GetPRLVersion	431
9.3.2.61	unpack_dms_GetSerialNumbers	431
9.3.2.62	unpack_dms_GetUSBComp	432
9.3.2.63	unpack_dms_GetVoiceNumber	432
9.3.2.64	unpack_dms_SetCrashAction	432
9.3.2.65	unpack_dms_SetCustFeature	433
9.3.2.66	unpack_dms_SetCustFeaturesV2	433
9.3.2.67	unpack_dms_SetEventReport	433
9.3.2.68	unpack_dms_SetEventReport_ind	434
9.3.2.69	unpack_dms_SetFirmwarePreference	434
9.3.2.70	unpack_dms_SetPower	434
9.3.2.71	unpack_dms_SetUSBComp	435
9.3.2.72	unpack_dms_SLQSDmsSwiGetResetInfo	435
9.3.2.73	unpack_dms_SLQSDmsSwiGetResetInfo_Ind	435
9.3.2.74	unpack_dms_SLQSDmsSwiIndicationRegister	436
9.3.2.75	unpack_dms_SLQSGetBandCapability	436
9.3.2.76	unpack_dms_SLQSSwiClearDyingGaspStatistics	437
9.3.2.77	unpack_dms_SLQSSwiGetDyingGaspCfg	437
9.3.2.78	unpack_dms_SLQSSwiGetDyingGaspStatistics	437
9.3.2.79	unpack_dms_SLQSSwiGetFirmwareCurr	438
9.3.2.80	unpack_dms_SLQSSwiGetFwUpdateStatus	438
9.3.2.81	unpack_dms_SLQSSwiSetDyingGaspCfg	438
9.3.2.82	unpack_dms_UIMGetICCID	439
9.4	fms.h File Reference	439
9.4.1	Macro Definition Documentation	440
9.4.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	440
9.4.1.2	FMS_GOBI_LISTENTRIES_MAX	440
9.4.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	440
9.4.1.4	FMS_GOBI_MBN_IMG_ID_STR_LEN	440
9.4.1.5	FMS_IMAGE_ID_BUILD_ID_LEN	440
9.4.1.6	FMS_IMAGE_ID_IMG_ID_LEN	440
9.4.1.7	FMS_IMAGE_ID_MAX_ENTRIES	440
9.4.1.8	FMS_IMAGE_ID_PRI_IMGTYPE	440
9.4.1.9	FMS_MAX_IMAGE_ID_ELEMENT	440
9.4.1.10	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	440

9.4.2	Function Documentation	440
9.4.2.1	GetValidFwPriCombinations	440
9.4.2.2	pack_fms_GetImagesPreference	441
9.4.2.3	pack_fms_GetStoredImages	441
9.4.2.4	pack_fms_SetImagesPreference	441
9.4.2.5	unpack_fms_GetImagesPreference	442
9.4.2.6	unpack_fms_GetStoredImages	442
9.4.2.7	unpack_fms_SetImagesPreference	442
9.5	loc.h File Reference	442
9.5.1	Macro Definition Documentation	444
9.5.1.1	LOC_UINT8_MAX_STRING_SZ	444
9.5.1.2	LOCEVENTMASKBATCHFULLNOTIFICATION	444
9.5.1.3	LOCEVENTMASKENGINESTATE	444
9.5.1.4	LOCEVENTMASKFIXSESSIONSTATE	444
9.5.1.5	LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION	444
9.5.1.6	LOCEVENTMASKGEOFENCEBREACHNOTIFICATION	445
9.5.1.7	LOCEVENTMASKGEOFENCEGENALERT	445
9.5.1.8	LOCEVENTMASKGNSSMEASUREMENTREPORT	445
9.5.1.9	LOCEVENTMASKGNSSSVINFO	445
9.5.1.10	LOCEVENTMASKINJECTPOSITIONREQ	445
9.5.1.11	LOCEVENTMASKINJECTPREDICTEDORBITSREQ	445
9.5.1.12	LOCEVENTMASKINJECTTIMERREQ	445
9.5.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	445
9.5.1.14	LOCEVENTMASKINVALIDVALUE	445
9.5.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	445
9.5.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	445
9.5.1.17	LOCEVENTMASKMOTIONDATACONTROL	446
9.5.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	446
9.5.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	446
9.5.1.20	LOCEVENTMASKNMEA	446
9.5.1.21	LOCEVENTMASKPEDOMETERCONTROL	446
9.5.1.22	LOCEVENTMASKPOSITIONREPORT	446
9.5.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	446
9.5.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	446
9.5.1.25	LOCEVENTMASKTIMESYNCREQ	446
9.5.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	446
9.5.1.27	LOCEVENTMASKWIFIREQ	446
9.5.2	Enumeration Type Documentation	447
9.5.2.1	anonymous enum	447
9.5.3	Function Documentation	447

9.5.3.1	pack_loc_DeleteAssistData	447
9.5.3.2	pack_loc_EventRegister	447
9.5.3.3	pack_loc_SetExtPowerState	447
9.5.3.4	pack_loc_SetOperationMode	448
9.5.3.5	pack_loc_SLQSLOCGetBestAvailPos	448
9.5.3.6	pack_loc_Start	448
9.5.3.7	pack_loc_Stop	449
9.5.3.8	unpack_loc_BestAvailPos_Ind	449
9.5.3.9	unpack_loc_DeleteAssistData	449
9.5.3.10	unpack_loc_EngineState_Ind	450
9.5.3.11	unpack_loc_EventRegister	450
9.5.3.12	unpack_loc_PositionRpt_Ind	450
9.5.3.13	unpack_loc_SetExtPowerConfig_Ind	451
9.5.3.14	unpack_loc_SetExtPowerState	451
9.5.3.15	unpack_loc_SetOperationMode	452
9.5.3.16	unpack_loc_SLQSLOCGetBestAvailPos	452
9.5.3.17	unpack_loc_Start	452
9.5.3.18	unpack_loc_Stop	453
9.6	nas.h File Reference	453
9.6.1	Macro Definition Documentation	458
9.6.1.1	NAS_MAX_DESCRIPTION_LENGTH	458
9.6.1.2	NAS_MAX_NUM_NETWORKS	458
9.6.1.3	NAS_OTA_MESSAGE_MAX_BUF_SIZE	458
9.6.1.4	NAS_PLMN_LENGTH	459
9.6.1.5	NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST	459
9.6.2	Enumeration Type Documentation	459
9.6.2.1	LIBPACK_NAS_LTE_CPHY_CA_BW_NRB	459
9.6.2.2	LIBPACK_NAS_LTE_CPHY_SCELL_STATE	459
9.6.2.3	NAS_LTE_CPHY_CA_BW_NRB_LITE	459
9.6.2.4	NAS_LTE_CPHY_SCELL_STATE_LITE	459
9.6.3	Function Documentation	459
9.6.3.1	pack_nas_GetACCOLC	459
9.6.3.2	pack_nas_GetANAAAAAuthenticationStatus	460
9.6.3.3	pack_nas_GetCDMANetworkParameters	460
9.6.3.4	pack_nas_GetHomeNetwork	460
9.6.3.5	pack_nas_GetNetworkPreference	461
9.6.3.6	pack_nas_GetRFInfo	461
9.6.3.7	pack_nas_GetServingNetwork	461
9.6.3.8	pack_nas_GetServingNetworkCapabilities	461
9.6.3.9	pack_nas_GetSignalStrengths	462

9.6.3.10	pack_nas_PerformNetworkScan	462
9.6.3.11	pack_nas_SetACCOLC	462
9.6.3.12	pack_nas_SetLURejectCallback	463
9.6.3.13	pack_nas_SetNetworkPreference	463
9.6.3.14	pack_nas_SetRFInfoCallback	463
9.6.3.15	pack_nas_SlqsGetLTECphyCAInfo	463
9.6.3.16	pack_nas_SLQSGetNetworkTime	464
9.6.3.17	pack_nas_SLQSGetPLMNName	464
9.6.3.18	pack_nas_SLQSGetServingSystem	464
9.6.3.19	pack_nas_SLQSGetSignalStrength	465
9.6.3.20	pack_nas_SLQSGetSysInfo	465
9.6.3.21	pack_nas_SLQSGetSysSelectionPref	465
9.6.3.22	pack_nas_SLQSInitiateNetworkRegistration	465
9.6.3.23	pack_nas_SLQSNasConfigSigInfo2	466
9.6.3.24	pack_nas_SLQSNasGetCellLocationInfo	466
9.6.3.25	pack_nas_SLQSNasGetSigInfo	466
9.6.3.26	pack_nas_SLQSNasIndicationRegisterExt	467
9.6.3.27	pack_nas_SLQSNasSwiModemStatus	467
9.6.3.28	pack_nas_SLQSNasSwiOTAMessageCallback	467
9.6.3.29	pack_nas_SLQSSetBandPreference	468
9.6.3.30	pack_nas_SLQSSetSignalStrengthsCallback	468
9.6.3.31	pack_nas_SLQSSetSysSelectionPref	468
9.6.3.32	pack_nas_SLQSSwiGetLteCQI	469
9.6.3.33	unpack_nas_GetACCOLC	469
9.6.3.34	unpack_nas_GetANAAAAuthenticationStatus	469
9.6.3.35	unpack_nas_GetCDMANetworkParameters	470
9.6.3.36	unpack_nas_GetHomeNetwork	470
9.6.3.37	unpack_nas_GetNetworkPreference	470
9.6.3.38	unpack_nas_GetRFInfo	470
9.6.3.39	unpack_nas_GetServingNetwork	471
9.6.3.40	unpack_nas_GetServingNetworkCapabilities	471
9.6.3.41	unpack_nas_GetSignalStrengths	471
9.6.3.42	unpack_nas_PerformNetworkScan	472
9.6.3.43	unpack_nas_SetACCOLC	472
9.6.3.44	unpack_nas_SetDataCapabilitiesCallback_ind	472
9.6.3.45	unpack_nas_SetEventReportInd	473
9.6.3.46	unpack_nas_SetLURejectCallback	473
9.6.3.47	unpack_nas_SetNasLTECphyCalndCallback_ind	473
9.6.3.48	unpack_nas_SetNetworkPreference	473
9.6.3.49	unpack_nas_SetRFInfoCallback	473

9.6.3.50	unpack_nas_SetRoamingIndicatorCallback_ind	474
9.6.3.51	unpack_nas_SetServingSystemCallback_ind	474
9.6.3.52	unpack_nas_SlqsGetLTECphyCAInfo	474
9.6.3.53	unpack_nas_SLQSGetNetworkTime	474
9.6.3.54	unpack_nas_SLQSGetPLMNName	474
9.6.3.55	unpack_nas_SLQSGetServingSystem	475
9.6.3.56	unpack_nas_SLQSGetSignalStrength	475
9.6.3.57	unpack_nas_SLQSGetSysInfo	475
9.6.3.58	unpack_nas_SLQSGetSysSelectionPref	476
9.6.3.59	unpack_nas_SLQSInitiateNetworkRegistration	476
9.6.3.60	unpack_nas_SLQSNasConfigSigInfo2	476
9.6.3.61	unpack_nas_SLQSNasGetCellLocationInfo	476
9.6.3.62	unpack_nas_SLQSNasGetSigInfo	477
9.6.3.63	unpack_nas_SLQSNasIndicationRegisterExt	477
9.6.3.64	unpack_nas_SLQSNasNetworkTimeCallBack_ind	477
9.6.3.65	unpack_nas_SLQSNasSigInfoCallback_ind	478
9.6.3.66	unpack_nas_SLQSNasSwiModemStatus	478
9.6.3.67	unpack_nas_SLQSNasSwiOTAMessageCallback	478
9.6.3.68	unpack_nas_SLQSNasSwiOTAMessageCallback_ind	479
9.6.3.69	unpack_nas_SLQSNasSysInfoCallback_ind	479
9.6.3.70	unpack_nas_SLQSSetBandPreference	479
9.6.3.71	unpack_nas_SLQSSetSignalStrengthsCallback	479
9.6.3.72	unpack_nas_SLQSSetSysSelectionPref	480
9.6.3.73	unpack_nas_SLQSSetSysSelectionPrefCallBack_ind	480
9.6.3.74	unpack_nas_SLQSSwiGetLteCQI	480
9.7	qaGobiApiTableBandClasses.h File Reference	481
9.7.1	Detailed Description	481
9.7.2	Band Classes (Value - Description)	481
9.7.2.1	LTE Bands	482
9.8	qaGobiApiTableCallControlReturnReasons.h File Reference	484
9.8.1	Detailed Description	484
9.8.2	Call Control Result Reasons (Value - Name - Description)	484
9.9	qaGobiApiTableCallEndReasons.h File Reference	484
9.9.1	Detailed Description	484
9.9.2	Call end reason codes (Code - Reason)	485
9.9.2.1	Technology-agnostic call end reasons	485
9.9.2.2	EVDO CDMA 1xEV-DO	485
9.9.2.3	WCDMA/GSM call end reasons	486
9.9.2.4	EVDO CDMA 1xEV-DO	488
9.9.2.5	call end reason type	488

9.9.2.6	Mobile IP call end reasons (Type=1)	488
9.9.2.7	Internal call end reasons (Type=2)	490
9.9.2.8	Call Manager defined call end reasons (Type=3)	491
9.9.2.9	3GPP specification defined call end reasons (Type=6)	496
9.9.2.10	PPP call end reasons (Type=7)	498
9.9.2.11	EHRPD call end reasons (Type=8)	499
9.9.2.12	IPv6 call end reasons (Type=9)	500
9.10	qaGobiApiTableCarrierCodes.h File Reference	500
9.10.1	Detailed Description	500
9.10.2	Carrier Codes (Number - Carrier)	500
9.11	qaGobiApiTableCodingScheme.h File Reference	502
9.11.1	Detailed Description	502
9.11.2	Call Control Result Reasons (Value - Name - Description)	502
9.11.2.1	Use of bits 3..0\n\n	502
9.11.3	Coding Group Bits 7..4(0001)	502
9.11.3.1	use of bits 3..0	502
9.11.4	Coding Group Bits 7..4(0010)	503
9.11.4.1	use of bits 3..0	503
9.11.5	Coding Group Bits 7..4(0011)	503
9.11.5.1	use of bits 3..0	503
9.11.6	Coding Group Bits 7..4(01xx)	503
9.11.6.1	use of bits 3..0	503
9.11.7	Coding Group Bits 7..4(1001)	504
9.11.7.1	Reserved coding groups	504
9.11.8	Coding Group Bits 7..4(1010..1101)	504
9.11.8.1	Reserved coding groups	504
9.11.9	Coding Group Bits 7..4(1110)	504
9.11.9.1	Defined by the WAP Forum	504
9.11.10	Coding Group Bits 7..4 (1111)	504
9.11.10.1	Data coding / message handling	504
9.11.11	Macro Definition Documentation	504
9.11.11.1	__GOBI_API_CODING_SCHEME_H__	504
9.12	qaGobiApiTableGpsCapabilityCodes.h File Reference	504
9.12.1	Detailed Description	505
9.12.2	GPS capability (Value - Capability)	505
9.13	qaGobiApiTablePowerModes.h File Reference	505
9.13.1	Detailed Description	505
9.13.2	Power Modes (Value - Description)	505
9.14	qaGobiApiTableRadioInterfaces.h File Reference	506
9.14.1	Detailed Description	506

9.14.2	Radio interface	506
9.14.2.1	Technology (Value - Radio Interface Technology)	506
9.15	qaGobiApiTableRegionCodes.h File Reference	506
9.15.1	Detailed Description	506
9.15.2	Region Codes (Code - Region)	506
9.16	qaGobiApiTableServiceOptions.h File Reference	507
9.16.1	Detailed Description	507
9.16.2	Service Option codes (Code - Reason)	507
9.16.2.1	Description	507
9.17	qaGobiApiTableSupServiceInfoClasses.h File Reference	509
9.17.1	Detailed Description	509
9.17.2	Supplementary Service Information Classes (Value - Service Class)	509
9.18	qaGobiApiTableSwiAudio.h File Reference	509
9.18.1	Detailed Description	509
9.18.2	ACDB Device (Device ID - description)	510
9.18.3	Physical Interface (Device ID - description - Interface parameters)	510
9.19	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	510
9.19.1	Detailed Description	510
9.19.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	510
9.20	qaGobiApiTableVoiceCallEndReasons.h File Reference	511
9.20.1	Detailed Description	511
9.20.2	Voice Call and supplementary services end reason codes (Code - Reason)	512
9.20.2.1	General	512
9.20.2.2	service Errors	513
9.20.2.3	control cause values	514
9.20.2.4	reject causes	516
9.20.2.5	reject causes	516
9.20.2.6	reject causes	517
9.20.2.7	stratum reject causes	517
9.20.2.8	reject causes	517
9.20.2.9	IP end reasons	517
9.21	qmerrno.h File Reference	518
9.21.1	Enumeration Type Documentation	520
9.21.1.1	eQCWWANError	520
9.21.1.2	qm_wds_ds_profile_extended_err_codes	525
9.22	qos.h File Reference	525
9.22.1	Macro Definition Documentation	527
9.22.1.1	LIBPACK_MAX_QOS_FILTERS	527
9.22.1.2	LIBPACK_MAX_QOS_FLOW_PER_APN_STATS	527
9.22.1.3	LIBPACK_MAX_QOS_FLOWS	527

9.22.2	Function Documentation	527
9.22.2.1	pack_qos_SLQSQosGetNetworkStatus	527
9.22.2.2	pack_qos_SLQSQosSwiReadApnExtraParams	527
9.22.2.3	pack_qos_SLQSQosSwiReadDataStats	528
9.22.2.4	pack_qos_SLQSSetQosEventCallback	529
9.22.2.5	unpack_qos_SLQSQosGetNetworkStatus	529
9.22.2.6	unpack_qos_SLQSQosSwiReadApnExtraParams	530
9.22.2.7	unpack_qos_SLQSQosSwiReadDataStats	530
9.22.2.8	unpack_qos_SLQSSetQosEventCallback	531
9.22.2.9	unpack_qos_SLQSSetQosEventCallback_ind	531
9.22.2.10	unpack_qos_SLQSSetQosNWStatusCallback_ind	532
9.22.2.11	unpack_qos_SLQSSetQosPriEventCallback_ind	532
9.22.2.12	unpack_qos_SLQSSetQosStatusCallback_ind	533
9.23	sms.h File Reference	533
9.23.1	Macro Definition Documentation	535
9.23.1.1	MAX_CDMA_ENC_MO_TXT_MSG_SIZE	535
9.23.1.2	MAX_MS_TRANSFER_ROUTE_MSG	535
9.23.1.3	MAX_MSC_ADDRESS_SIZE	535
9.23.1.4	MAX_MSE_TWS_MSG	535
9.23.1.5	MAX_SMS_LIST_SIZE	535
9.23.1.6	MAX_SMS_MESSAGE_SIZE	535
9.23.2	Typedef Documentation	535
9.23.2.1	sMSCAddressInfo	535
9.23.2.2	sMSEtwSMessageInfo	536
9.23.2.3	sMSEtwSPImnInfo	536
9.23.2.4	sSMMessageModelInfo	536
9.23.2.5	sMSMTMessageInfo	536
9.23.2.6	sMSOnIMSInfo	536
9.23.2.7	sMSTransferRouteMTMessageInfo	536
9.23.3	Enumeration Type Documentation	537
9.23.3.1	eqmiCbKSetStatus	537
9.23.4	Function Documentation	537
9.23.4.1	pack_sms_SendSMS	537
9.23.4.2	pack_sms_SetNewSMSCallback	537
9.23.4.3	pack_sms_SLQSDeleteSMS	538
9.23.4.4	pack_sms_SLQSGetSMS	538
9.23.4.5	pack_sms_SLQSGetSMSList	539
9.23.4.6	pack_sms_SLQSModifySMSStatus	539
9.23.4.7	unpack_sms_SendSMS	539
9.23.4.8	unpack_sms_SetNewSMSCallback	540

9.23.4.9	unpack_sms_SetNewSMSCallback_ind	540
9.23.4.10	unpack_sms_SLQSDeleteSMS	540
9.23.4.11	unpack_sms_SLQSGetSMS	541
9.23.4.12	unpack_sms_SLQSGetSMSList	541
9.23.4.13	unpack_sms_SLQSModifySMSStatus	541
9.23.4.14	unpack_sms_SLQSWmsMemoryFullCallBack_ind	542
9.24	SwiDataTypes.h File Reference	542
9.24.1	Detailed Description	543
9.24.2	Macro Definition Documentation	543
9.24.2.1	QMI_NO_LTE_FW_SUPPORT	543
9.24.2.2	QMI_TLV_PLACEHOLDER	543
9.24.2.3	SWI_API	543
9.24.2.4	UNUSEDPARAM	543
9.24.3	Typedef Documentation	543
9.24.3.1	BOOL	543
9.24.3.2	BYTE	543
9.24.3.3	CHAR	543
9.24.3.4	FLOAT	543
9.24.3.5	INT32	543
9.24.3.6	INT8	543
9.24.3.7	LPCSTR	543
9.24.3.8	SHORT	543
9.24.3.9	ULONG	543
9.24.3.10	ULONGLONG	543
9.24.3.11	USHORT	543
9.24.3.12	WORD	543
9.25	swiloc.h File Reference	543
9.25.1	Function Documentation	544
9.25.1.1	pack_swiloc_SwiLocGetAutoStart	544
9.25.1.2	pack_swiloc_SwiLocSetAutoStart	544
9.25.1.3	unpack_swiloc_SwiLocGetAutoStart	544
9.25.1.4	unpack_swiloc_SwiLocSetAutoStart	545
9.26	swioma.h File Reference	545
9.26.1	Macro Definition Documentation	546
9.26.1.1	LIBPACK_MAX_SWIOMA_STR_LEN	546
9.26.2	Function Documentation	546
9.26.2.1	pack_swioma_SLQSOMADMAAlertCallback	546
9.26.2.2	pack_swioma_SLQSOMADMCancelSession	547
9.26.2.3	pack_swioma_SLQSOMADMGetSessionInfo	547
9.26.2.4	pack_swioma_SLQSOMADMGetSettings	548

9.26.2.5	pack_swima_SLQSOMADMSelect	549
9.26.2.6	pack_swima_SLQSOMADMSetSettings	549
9.26.2.7	pack_swima_SLQSOMADMStartSession	550
9.26.2.8	unpack_swima_SLQSOMADMAAlertCallback	550
9.26.2.9	unpack_swima_SLQSOMADMAAlertCallback_ind	551
9.26.2.10	unpack_swima_SLQSOMADMCancelSession	551
9.26.2.11	unpack_swima_SLQSOMADMGetSessionInfo	552
9.26.2.12	unpack_swima_SLQSOMADMGetSettings	552
9.26.2.13	unpack_swima_SLQSOMADMSelect	553
9.26.2.14	unpack_swima_SLQSOMADMSetSettings	553
9.26.2.15	unpack_swima_SLQSOMADMStartSession	554
9.27	SWIWWANCMAPI.h File Reference	554
9.28	uim.h File Reference	554
9.28.1	Macro Definition Documentation	556
9.28.1.1	MAX_DESCRIPTION_LENGTH	556
9.28.1.2	MAX_ICCID_LENGTH	556
9.28.1.3	MAX_NO_OF_APPLICATIONS	556
9.28.1.4	MAX_NO_OF_SLOTS	556
9.28.1.5	MAX_SLOTS_STATUS	556
9.28.1.6	UIM_MAX_DESCRIPTION_LENGTH	556
9.28.1.7	UIM_MAX_NO_OF_APPLICATIONS	556
9.28.1.8	UIM_MAX_NO_OF_SLOTS	556
9.28.1.9	UIM_UINT8_MAX_STRING_SZ	556
9.28.2	Function Documentation	556
9.28.2.1	pack_uim_ChangePin	556
9.28.2.2	pack_uim_GetCardStatus	556
9.28.2.3	pack_uim_ReadTransparent	557
9.28.2.4	pack_uim_SetPinProtection	557
9.28.2.5	pack_uim_SLQSUIEventRegister	558
9.28.2.6	pack_uim_SLQSUIGetSlotsStatus	558
9.28.2.7	pack_uim_SLQSUISwitchSlot	558
9.28.2.8	pack_uim_UnblockPin	559
9.28.2.9	pack_uim_VerifyPin	559
9.28.2.10	unpack_uim_ChangePin	559
9.28.2.11	unpack_uim_GetCardStatus	560
9.28.2.12	unpack_uim_ReadTransparent	560
9.28.2.13	unpack_uim_SetPinProtection	560
9.28.2.14	unpack_uim_SetUimSlotStatusChangeCallback_ind	561
9.28.2.15	unpack_uim_SLQSUIEventRegister	561
9.28.2.16	unpack_uim_SLQSUIGetSlotsStatus	561

9.28.2.17 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind	562
9.28.2.18 unpack_uim_SLQSUIMSwitchSlot	562
9.28.2.19 unpack_uim_UnblockPin	563
9.28.2.20 unpack_uim_VerifyPin	563
9.29 wds.h File Reference	563
9.29.1 Macro Definition Documentation	567
9.29.1.1 IPV6_ADDRESS_ARRAY_SIZE	567
9.29.1.2 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	567
9.29.1.3 PACK_WDS_IPV4	567
9.29.1.4 PACK_WDS_IPV6	567
9.29.2 Typedef Documentation	567
9.29.2.1 UnpackQmiProfileInfo	567
9.29.3 Function Documentation	567
9.29.3.1 pack_wds_GetConnectionRate	568
9.29.3.2 pack_wds_GetDefaultProfile	568
9.29.3.3 pack_wds_GetDefaultProfileNum	568
9.29.3.4 pack_wds_GetDormancyState	569
9.29.3.5 pack_wds_GetLastMobileIPError	569
9.29.3.6 pack_wds_GetMobileIP	570
9.29.3.7 pack_wds_GetMobileIPProfile	570
9.29.3.8 pack_wds_GetPacketStatus	570
9.29.3.9 pack_wds_GetSessionDuration	571
9.29.3.10 pack_wds_GetSessionState	571
9.29.3.11 pack_wds_RMSetTransferStatistics	572
9.29.3.12 pack_wds_SetDefaultProfile	572
9.29.3.13 pack_wds_SetDefaultProfileNum	572
9.29.3.14 pack_wds_SetMobileIPProfile	573
9.29.3.15 pack_wds_SLQSCreateProfile	573
9.29.3.16 pack_wds_SLQSDeleteProfile	574
9.29.3.17 pack_wds_SLQSGet3GPPConfigItem	574
9.29.3.18 pack_wds_SLQSGetCurrDataSystemStat	574
9.29.3.19 pack_wds_SLQSGetDataBearerTechnology	575
9.29.3.20 pack_wds_SLQSGetDUNCallInfo	575
9.29.3.21 pack_wds_SLQSGetProfileSettings	576
9.29.3.22 pack_wds_SLQSGetRuntimeSettings	576
9.29.3.23 pack_wds_SLQSModifyProfile	577
9.29.3.24 pack_wds_SLQSSet3GPPConfigItem	577
9.29.3.25 pack_wds_SLQSSetIPFamilyPreference	577
9.29.3.26 pack_wds_SLQSSetWdsEventCallback	578
9.29.3.27 pack_wds_SLQSSetDHCPv4ClientConfig	578

9.29.3.28 pack_wds_SLQSStartDataSession	578
9.29.3.29 pack_wds_SLQSStopDataSession	579
9.29.3.30 pack_wds_SLQSWdsSwiPDPRuntimeSettings	579
9.29.3.31 unpack_wds_GetConnectionRate	580
9.29.3.32 unpack_wds_GetDefaultProfile	580
9.29.3.33 unpack_wds_GetDefaultProfileNum	580
9.29.3.34 unpack_wds_GetDormancyState	581
9.29.3.35 unpack_wds_GetLastMobileIPError	581
9.29.3.36 unpack_wds_GetMobileIP	581
9.29.3.37 unpack_wds_GetMobileIPProfile	582
9.29.3.38 unpack_wds_GetPacketStatus	582
9.29.3.39 unpack_wds_GetSessionDuration	583
9.29.3.40 unpack_wds_GetSessionState	583
9.29.3.41 unpack_wds_RMSetTransferStatistics	583
9.29.3.42 unpack_wds_SetDefaultProfile	584
9.29.3.43 unpack_wds_SetDefaultProfileNum	584
9.29.3.44 unpack_wds_SetMobileIPProfile	584
9.29.3.45 unpack_wds_SLQSCreateProfile	585
9.29.3.46 unpack_wds_SLQSDeleteProfile	585
9.29.3.47 unpack_wds_SLQSGet3GPPConfigItem	585
9.29.3.48 unpack_wds_SLQSGetCurrDataSystemStat	586
9.29.3.49 unpack_wds_SLQSGetDataBearerTechnology	586
9.29.3.50 unpack_wds_SLQSGetDUNCallInfo	586
9.29.3.51 unpack_wds_SLQSGetProfileSettings	587
9.29.3.52 unpack_wds_SLQSGetRuntimeSettings	587
9.29.3.53 unpack_wds_SLQSModifyProfile	587
9.29.3.54 unpack_wds_SLQSSet3GPPConfigItem	588
9.29.3.55 unpack_wds_SLQSSetIPFamilyPreference	588
9.29.3.56 unpack_wds_SLQSSetPacketSrvStatusCallback	588
9.29.3.57 unpack_wds_SLQSSetWdsEventCallback	588
9.29.3.58 unpack_wds_SLQSSetWdsEventCallback_ind	589
9.29.3.59 unpack_wds_SLQSSetDHCPv4ClientConfig	589
9.29.3.60 unpack_wds_SLQSStartDataSession	589
9.29.3.61 unpack_wds_SLQSStopDataSession	590
9.29.3.62 unpack_wds_SLQSWdsSwiPDPRuntimeSettings	590

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)	19
----------------------------------	----

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tables	21
----------------------------------	----

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

appStats	23
CarrierImage_t	26
cdmaSSInfo	27
connectionStatus	27
currNetworkInfo	28
dms_ActivationStatusTlv	28
dms_OperatingModeTlv	29
DMScustSettingInfo	30
DMScustSettingList	30
DMSgetCustomFeatureV2	31
DMSgetCustomInput	32
dunchannelRate	32
eTWSPLMNInfoTlv	33
FMSImageElement	33
FMSImageIdElement	34
FMSImageIdEntries	35
FMSImageList	36
FMSPrefImageList	36
hdrSSInfo	37
image_info_t	37
ipv6AddressInfo	38
LibPackGPRSRequestedQoS	38
LibpackProfile3GPP	39
LibpackProfile3GPP2	45
LibPackprofile_3GPP	50
LibPackprofile_3GPP2	56
LibPackQosClassID	61
LibPackTFTIDParams	62
LibPackUMTSQoS	64
LibPackUMTSReqQoSSigInd	66
loc_BdsSV	67
loc_BdsSVInfo	67
loc_CellDb	68
loc_ClkInfo	69
loc_GnssData	70
loc_gpsTime	71
loc_LocApplicationInfo	72
loc_precisionDilution	73

loc_sensorDataUsage	74
loc_SV	75
loc_SVInfo	75
loc_svUsedforFix	76
lteSSInfo	77
messageModeTlv	77
nas_acqOrderPref	78
nas_AddCDMASysInfo	78
nas_AddSysInfo	79
nas_CallBarringSysInfo	79
nas_callBarStatus	80
nas_CDMAECIOThresh	81
nas_CDMAInfo	82
nas_CDMARSSIThresh	83
nas_CDMASysInfo	83
nas_CDMASysInfoExt	87
nas_cellParams	87
nas_CommInfo	88
nas_CSGID	90
nas_currentPLMN	91
nas_dataSrvCapabilities	92
nas_detailSvcInfo	92
nas_ecioListElement	94
nas_errorRateListElement	95
nas_GERANInfo	95
nas_geranInstInfo	97
nas_gsmCellInfo	98
nas_GSMRSSIThresh	99
nas_GSMSrvStatusInfo	100
nas_GSMSysInfo	101
nas_HDRECIOThresh	104
nas_HDRIOThresh	104
nas_HDRRSSIThresh	105
nas_HDRSINRThreshold	105
nas_HDRSysInfo	106
nas_infoInterFreq	108
nas_lteGsmCellInfo	109
nas_LTEInfo	110
nas_LTEInfoInterfreq	113
nas_LTEInfoIntrafreq	113
nas_LTEInfoNeighboringGSM	115
nas_LTEInfoNeighboringWCDMA	116
nas_lteRsrpInformation	117
nas_LTERSRPThresh	117
nas_LTERSQThresh	118
nas_LTERSSIThresh	118
nas_LTESigRptConfig	119
nas_lteSnrinformation	119
nas_LTESNRThreshold	120
nas_LTESysInfo	120
nas_lteWcdmaCellInfo	123
nas_MNRInfo	124
nas_netSelectionPref	125
nas_nmrCellInfo	125
nas_PhyCaAggPcellInfo	127
nas_PhyCaAggScellDIBw	127
nas_PhyCaAggScellIndex	128
nas_PhyCaAggScellIndType	128

nas_PhyCaAggScellInfo	129
nas_qaQmi3Gpp2TimeZone	132
nas_QmiNas3GppNetworkInfo	133
nas_QmiNas3GppNetworkRAT	133
nas_QmisNasPcsDigit	134
nas_RejectReasonTlv	135
nas_RFInfoTlv	135
nas_roamIndList	136
nas_rsrqInformation	137
nas_rxSignalStrengthListElement	137
nas_servSystem	138
nas_SignalStrengthTlv	140
nas_SLQSSignalStrengthsIndReq	140
nas_SLQSSignalStrengthsInformation	141
nas_SLQSSignalStrengthsTlv	142
nas_SrvStatusInfo	142
nas_sysInfoCommon	143
nas_TDSCDMAECIOThresh	146
nas_TDSCDMARSCPTthresh	146
nas_TDSCDMARSSIThresh	147
nas_TDSCDMASINRThresh	147
nas_timeInfo	148
nas_UMTSInfo	149
nas_UMTSinstInfo	151
nas_umtsLTENbrCell	152
nas_UniversalTime	153
nas_wcdmaCellInfo	154
nas_WCDMAECIOThresh	155
nas_WCDMAInfoLTENeighborCell	156
nas_WCDMARSSIThresh	156
nas_WCDMASysInfo	157
NASBandPreferenceTlv	160
NASEmergencyModeTlv	161
NasGetLTECphyCaInfo	161
NASGWAcqOrderPrefTlv	161
NASLTEBandPreferenceTlv	162
NASLteNasReleaseInfoTlv	162
NASModePreferenceTlv	162
NASNetSelPreferenceTlv	162
NASOTAMessageTlv	163
NASPhyCaAggPcellInfo	163
NASPhyCaAggScellIDBw	164
NASPhyCaAggScellIndex	164
NASPhyCaAggScellIndType	165
NASPhyCaAggScellInfo	166
NASPRLPreferenceTlv	167
NASQmiCbkNasSwiOTAMessageInd	167
NASQmiCbkNasSystemSelPrefInd	167
NASRoamPreferenceTlv	168
NASServDomainPrefTlv	168
NASServingSystemInfo	168
NASTimeInfoTlv	170
newMTMessageTlv	170
pack_dms_GetCustFeaturesV2_t	171
pack_dms_SetCrashAction_t	171
pack_dms_SetCustFeature_t	172
pack_dms_SetCustFeaturesV2_t	172
pack_dms_SetEventReport_t	173

pack_dms_SetPower_t	173
pack_dms_SetUSBComp_t	174
pack_dms_SLQSDmsSwiIndicationRegister_t	174
pack_dms_SLQSSwiSetDyingGaspCfg_t	174
pack_dms_UIMGetICCID_t	175
pack_fms_GetImagesPreference_t	175
pack_fms_GetStoredImages_t	176
pack_fms_SetImagesPreference_t	176
pack_loc_Delete_Assist_Data_t	177
pack_loc_EventRegister_t	178
pack_loc_SetExtPowerState_t	180
pack_loc_SetOperationMode_t	180
pack_loc_SLQSLOCGetBestAvailPos_t	181
pack_loc_Start_t	181
pack_loc_Stop_t	183
pack_nas_SetACCOLC_t	183
pack_nas_SetNetworkPreference_t	184
pack_nas_SLQSGetPLMNName_t	185
pack_nas_SLQSIInitiateNetworkRegistration_t	185
pack_nas_SLQSNasConfigSigInfo2_t	186
pack_nas_SLQSNasIndicationRegisterExt_t	191
pack_nas_SLQSNasSwiOTAMessageCallback_t	194
pack_nas_SLQSSetSignalStrengthsCallback_t	195
pack_nas_SLQSSetSysSelectionPref_t	195
pack_qmi_t	200
pack_qos_SLQSQosSwiReadApnExtraParams_t	200
pack_qos_SLQSQosSwiReadDataStats_t	201
pack_qos_SLQSSetQosEventCallback_t	201
pack_sms_SendSMS_t	202
pack_sms_SetNewSMSCallback_t	202
pack_sms_SLQSDeleteSMS_t	203
pack_sms_SLQSGetSMS_t	204
pack_sms_SLQSGetSMSList_t	204
pack_sms_SLQSModifySMSStatus_t	205
pack_swiloc_SwiLocSetAutoStart_t	206
pack_swioama_SLQSOMADMCancelSession_t	207
pack_swioama_SLQSOMADMGetSessionInfo_t	208
pack_swioama_SLQSOMADMSelectSelection_t	208
pack_swioama_SLQSOMADMSetSettings_t	209
pack_swioama_SLQSOMADMStartSession_t	210
pack_uim_ChangePin_t	211
pack_uim_ReadTransparent_t	212
pack_uim_SetPinProtection_t	213
pack_uim_SLQSUIMEventRegister_t	214
pack_uim_SLQSUIMSwitchSlot_t	214
pack_uim_UnblockPin_t	215
pack_uim_VerifyPin_t	216
pack_wds_GetDefaultProfile_t	217
pack_wds_GetDefaultProfileNum_t	217
pack_wds_GetDormancyState_t	218
pack_wds_GetLastMobileIPError_t	218
pack_wds_GetMobileIP_t	218
pack_wds_GetMobileIPProfile_t	218
pack_wds_GetPacketStatus_t	218
pack_wds_GetSessionDuration_t	219
pack_wds_RMSetTransferStatistics_t	219
pack_wds_SetDefaultProfile_t	219
pack_wds_SetDefaultProfileNum_t	220

pack_wds_SetMobileIPProfile_t	220
pack_wds_SLQSCreateProfile_t	222
pack_wds_SLQSDeleteProfile_t	222
pack_wds_SLQSGetCurrDataSystemStat_t	223
pack_wds_SLQSGetDataBearerTechnology_t	223
pack_wds_SLQSGetDUNCallInfo_t	223
pack_wds_SLQSGetProfileSettings_t	224
pack_wds_SLQSGetRuntimeSettings_t	224
pack_wds_SLQSModifyProfile_t	225
pack_wds_SLQSSet3GPPConfigItem_t	226
pack_wds_SLQSSetIPFamilyPreference_t	227
pack_wds_SLQSSetWdsEventCallback_t	227
pack_wds_SLQSSetDHCPv4ClientConfig_t	228
pack_wds_SLQSStartDataSession_t	228
pack_wds_SLQSStopDataSession_t	230
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	230
PackCreateProfileOut	230
packgetDyingGaspCfg	231
packgetDyingGaspStatistics	231
qmiSmsMessageList	232
qmiWDSDataBearerTechnology	232
RFBandInfoElements	232
rmTransferStaticsReq	233
slot_t	233
slotInf	234
slots_t	236
sMSCAddress	236
sMSCAddressTlv	237
sMSEtwsMessage	237
sMSEtwsMessageTlv	238
sMSEtwsPlmn	238
sMSMessageMode	239
sMSMTMessage	239
sMSOnIMS	239
sMSOnIMSTlv	240
sMSTransferRouteMTMessage	240
tdscdmaSigInfoExt	241
transferRouteMessageTlv	241
transferStatInd	242
uim_appStatus	242
uim_cardResult	245
uim_cardStatus	246
uim_changeUIMPIN	247
uim_encryptedPIN1	248
uim_fileInfo	249
uim_hotSwapStatus	249
uim_readResult	250
uim_readTransparentInfo	251
uim_remainingRetries	251
uim_sessionInformation	252
uim_setPINProtection	253
uim_slotInfo	253
uim_UIMSessionInformation	255
uim_unblockUIMPIN	256
uim_verifyUIMPIN	257
unpack_dms_GetActivationState_t	257
unpack_dms_GetBandCapability_t	258
unpack_dms_GetCrashAction_t	258

unpack_dms_GetCustFeature_t	259
unpack_dms_GetCustFeaturesV2_t	259
unpack_dms_GetDeviceCap_t	260
unpack_dms_GetDeviceCapabilities_t	260
unpack_dms_GetDeviceHardwareRev_t	261
unpack_dms_GetDeviceMfr_t	261
unpack_dms_GetDeviceSerialNumbers_t	262
unpack_dms_GetFirmwareInfo_t	262
unpack_dms_GetFirmwareRevision_t	263
unpack_dms_GetFirmwareRevisions_t	264
unpack_dms_GetFSN_t	264
unpack_dms_GetHardwareRevision_t	265
unpack_dms_GetIMSI_t	265
unpack_dms_GetModelID_t	265
unpack_dms_GetNetworkTime_t	265
unpack_dms_GetPower_t	266
unpack_dms_GetPRLVersion_t	267
unpack_dms_GetSerialNumbers_t	267
unpack_dms_GetUSBComp_t	268
unpack_dms_GetVoiceNumber_t	268
unpack_dms_SetCrashAction_t	268
unpack_dms_SetCustFeature_t	269
unpack_dms_SetCustFeaturesV2_t	269
unpack_dms_SetEventReport_ind_t	269
unpack_dms_SetEventReport_t	270
unpack_dms_SetFirmwarePreference_t	270
unpack_dms_SetPower_t	270
unpack_dms_SetUSBComp_t	270
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t	271
unpack_dms_SLQSDmsSwiGetResetInfo_t	272
unpack_dms_SLQSDmsSwiIndicationRegister_t	273
unpack_dms_SLQSSGetBandCapability_t	273
unpack_dms_SLQSSwiClearDyingGaspStatistics_t	276
unpack_dms_SLQSSwiGetDyingGaspCfg_t	276
unpack_dms_SLQSSwiGetDyingGaspStatistics_t	277
unpack_dms_SLQSSwiGetFirmwareCurr_t	277
unpack_dms_SLQSSwiGetFwUpdateStatus_t	278
unpack_dms_SLQSSwiSetDyingGaspCfg_t	280
unpack_dms_UIMGetICCID_t	280
unpack_fms_GetImagesPreference_t	281
unpack_fms_GetStoredImages_t	281
unpack_fms_SetImagesPreference_t	282
unpack_loc_BestAvailPos_Ind_t	282
unpack_loc_Delete_Assist_Data_t	288
unpack_loc_EngineState_Ind_t	289
unpack_loc_EventRegister_t	289
unpack_loc_PositionRpt_Ind_t	290
unpack_loc_SetExtPowerConfig_Ind_t	296
unpack_loc_SetExtPowerState_t	296
unpack_loc_SetOperationMode_t	297
unpack_loc_SLQSLOCGetBestAvailPos_t	297
unpack_loc_Start_t	297
unpack_loc_Stop_t	298
unpack_nas_GetCDMANetworkParameters_t	298
unpack_nas_GetHomeNetwork_t	299
unpack_nas_GetNetworkPreference_t	300
unpack_nas_GetRFInfo_t	301
unpack_nas_GetServingNetwork_t	301

unpack_nas_GetServingNetworkCapabilities_t	302
unpack_nas_GetSignalStrengths_t	303
unpack_nas_PerformNetworkScan_t	303
unpack_nas_SetDataCapabilitiesCallback_ind_t	304
unpack_nas_SetEventReportInd_t	304
unpack_nas_SetNasLTECphyCAIndCallback_ind_t	305
unpack_nas_SetNetworkPreference_t	306
unpack_nas_SetRoamingIndicatorCallback_ind_t	307
unpack_nas_SetServingSystemCallback_ind_t	307
unpack_nas_SLqsGetLTECphyCAInfo_t	307
unpack_nas_SLQSGetNetworkTime_t	308
unpack_nas_SLQSGetPLMNName_t	308
unpack_nas_SLQSGetServingSystem_t	309
unpack_nas_SLQSGetSignalStrength_t	311
unpack_nas_SLQSGetSysInfo_t	313
unpack_nas_SLQSGetSysSelectionPref_t	315
unpack_nas_SLQSNasGetCellLocationInfo_t	319
unpack_nas_SLQSNasGetSigInfo_t	321
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t	321
unpack_nas_SLQSNasSigInfoCallback_ind_t	322
unpack_nas_SLQSNasSwiModemStatus_t	323
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t	324
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t	324
unpack_nas_SLQSSwiGetLteCQI_t	324
unpack_nas_SLQSSysInfoCallback_ind_t	325
unpack_omaDmConfigTlv_t	328
unpack_omaDmFotaTlv_t	329
unpack_omaDmNotificationsTlv_t	331
unpack_qmi_t	331
unpack_qos_dataRate_t	332
unpack_qos_IPv4Addr_t	332
unpack_qos_IPv6Addr_t	333
unpack_qos_IPv6TrafCls_t	333
unpack_qos_pktErrRate_t	334
unpack_qos_Port_t	334
unpack_qos_QosFlowInfo_t	335
unpack_qos_QosFlowInfoState_t	336
unpack_qos_SLQSQosGetNetworkStatus_t	337
unpack_qos_SLQSQosSwiReadApnExtraParams_t	337
unpack_qos_SLQSQosSwiReadDataStats_t	339
unpack_qos_SLQSSetQosEventCallback_ind_t	340
unpack_qos_SLQSSetQosNWStatusCallback_ind_t	340
unpack_qos_SLQSSetQosPriEventCallback_ind_t	341
unpack_qos_SLQSSetQosStatusCallback_ind_t	341
unpack_qos_swiQosFilter_t	343
unpack_qos_swiQosFlow_t	346
unpack_qos_tokenBucket_t	350
unpack_qos_Tos_t	351
unpack_qosFlowStat_t	351
unpack_sms_SendSMS_t	352
unpack_sms_SetNewSMSCallback_ind_t	352
unpack_sms_SetNewSMSCallback_t	353
unpack_sms_SLQSDeleteSMS_t	354
unpack_sms_SLQSGetSMS_t	354
unpack_sms_SLQSGetSMSList_t	354
unpack_sms_SLQSModifySMSStatus_t	355
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	355
unpack_swiloc_SwiLocGetAutoStart_t	356

unpack_swioma_SLQSOMADMAAlertCallback_ind_t	357
unpack_swioma_SLQSOMADMGetSessionInfo_t	358
unpack_swioma_SLQSOMADMGetSettings_t	361
unpack_swioma_SLQSOMADMStartSession_t	362
unpack_uim_ChangePin_t	363
unpack_uim_GetCardStatus_t	363
unpack_uim_ReadTransparent_t	364
unpack_uim_SetPinProtection_t	365
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	366
unpack_uim_SLQSUIEventRegister_t	366
unpack_uim_SLQSUIGetSlotsStatus_t	366
unpack_uim_SLQSUISetStatusChangeCallBack_ind_t	367
unpack_uim_UnblockPin_t	367
unpack_uim_VerifyPin_t	368
unpack_wds_GetConnectionRate_t	369
unpack_wds_GetDefaultProfile_t	369
unpack_wds_GetDefaultProfileNum_t	371
unpack_wds_GetDormancyState_t	371
unpack_wds_GetLastMobileIPError_t	371
unpack_wds_GetMobileIP_t	372
unpack_wds_GetMobileIPProfile_t	372
unpack_wds_GetPacketStatus_t	373
unpack_wds_GetSessionDuration_t	374
unpack_wds_GetSessionState_t	374
unpack_wds_RMSetTransferStatistics_t	375
unpack_wds_SetMobileIPProfile_t	375
unpack_wds_SLQSCreateProfile_t	375
unpack_wds_SLQSDeleteProfile_t	375
unpack_wds_SLQSGet3GPPConfigItem_t	376
unpack_wds_SLQSGetCurrDataSystemStat_t	377
unpack_wds_SLQSGetDataBearerTechnology_t	377
unpack_wds_SLQSGetDUNCallInfo_t	378
unpack_wds_SLQSGetProfileSettings_t	379
unpack_wds_SLQSGetRuntimeSettings_t	379
unpack_wds_SLQSModifyProfile_t	381
unpack_wds_SLQSSetIPFamilyPreference_t	381
unpack_wds_SLQSSetPacketSrvStatusCallback_t	382
unpack_wds_SLQSSetWdsEventCallback_ind_t	383
unpack_wds_SLQSSetDHCPv4ClientConfig_t	384
unpack_wds_SLQSStartDataSession_t	385
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	385
UnPackGetProfileSettingOut	387
unpackWdsProfileParam	387
wds_currNetworkInfo	387
wds_Domain	389
wds_DomainNameList	390
wds_GPRSQoS	390
wds_IPV6AddressInfo	391
wds_IPV6GWAddressInfo	391
wds_PCSCFFQDNAddress	392
wds_PCSCFFQDNAddressList	392
wds_PCSCFIPv4ServerAddressList	393
wds_ProfileIdentifier	393
wds_profileInfo	394
wds_UMTSMInQoS	394
wdsDhcpv4HwConfig	397
wdsDhcpv4Option	397
wdsDhcpv4OptionList	398

wdsDhcpv4ProfileId	398
--	-----

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	401
common.h		401
dms.h		405
fms.h		439
loc.h		442
nas.h		453
qaGobiApiTableBandClasses.h	Network Access Service API Band Classes table	481
qaGobiApiTableCallControlReturnReasons.h	Call Control Return Reasons table	484
qaGobiApiTableCallEndReasons.h	Wireless Data Service Call End Reasons	484
qaGobiApiTableCarrierCodes.h	Carrier Codes table	500
qaGobiApiTableCodingScheme.h	Data Coding Scheme	502
qaGobiApiTableGpsCapabilityCodes.h	Position Determination Service API GPS Capability Codes	504
qaGobiApiTablePowerModes.h	Device Management Service API Power Modes table	505
qaGobiApiTableRadiolInterfaces.h	Network Access Service API Radio Interfaces table	506
qaGobiApiTableRegionCodes.h	Region Codes table	506
qaGobiApiTableServiceOptions.h	Voice Service Options	507
qaGobiApiTableSupServiceInfoClasses.h	Voice Supplementary Service Information Classes	509
qaGobiApiTableSwiAudio.h	Swi Audio related tables	509
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h	Update Complete Status table	510
qaGobiApiTableVoiceCallEndReasons.h	Voice Service Call and supplementary services end reasons	511
qmerrno.h		518
qos.h		525

sms.h	533
SwiDataTypes.h	
SWI data types	542
swiloc.h	543
swioma.h	545
SWIWWANCMAPI.h	554
uim.h	554
wds.h	563

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[<small>MAX_DESCRIPTION_LENGTH</small>]</i>	<ul style="list-style-type: none"> Application identifier value.

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

8.1.2 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_PrImageId](#) [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_PriImageId[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 <code>sMSEtwSPlmnInfo</code> 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 <code>FMSImageElement</code>

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

8.19.2 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::rssi

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>precedenceClass</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>peakThroughputClass</i>	<ul style="list-style-type: none"> • Peak throughput class
<i>meanThroughputClass</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` * `pPDType`
- `uint8_t` * `pPdpHdrCompType`
- `uint8_t` * `pPdpDataCompType`
- `uint8_t` * `pAPNName`
- `uint16_t` * `pAPNnameSize`
- `uint32_t` * `pPriDNSIPv4AddPref`
- `uint32_t` * `pSecDNSIPv4AddPref`
- `LibPackUMTSQoS` * `pUMTSReqQoS`
- `LibPackUMTSQoS` * `pUMTSMInQoS`
- `LibPackGPRSRequestedQoS` * `pGPRSRequestedQoS`
- `LibPackGPRSRequestedQoS` * `pGPRSMinimumQoS`
- `uint8_t` * `pUsername`
- `uint16_t` * `pUsernameSize`
- `uint8_t` * `pPassword`
- `uint16_t` * `pPasswordSize`
- `uint8_t` * `pAuthenticationPref`
- `uint32_t` * `pIPv4AddrPref`
- `uint8_t` * `pPcscfAddrUsingPCO`
- `uint8_t` * `pPdpAccessConFlag`
- `uint8_t` * `pPcscfAddrUsingDhcp`
- `uint8_t` * `pImCnFlag`
- `LibPackTFTIDParams` * `pTFTID1Params`
- `LibPackTFTIDParams` * `pTFTID2Params`
- `uint8_t` * `pPdpContext`
- `uint8_t` * `pSecondaryFlag`
- `uint8_t` * `pPrimaryID`
- `uint16_t` * `pIPv6AddPref`
- `LibPackUMTSReqQoSSigInd` * `pUMTSReqQoSSigInd`
- `LibPackUMTSReqQoSSigInd` * `pUMTSMInQoSsigInd`
- `uint16_t` * `pPriDNSIPv6addpref`
- `uint16_t` * `pSecDNSIPv6addpref`
- `uint8_t` * `pAddrAllocPref`
- `LibPackQoSClassID` * `pQoSClassID`
- `uint8_t` * `pAPNDisabledFlag`
- `uint32_t` * `pPDNInactivTimeout`
- `uint8_t` * `pAPNClass`

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 pProfileName field. Size of this parameter is 2 uint8_ts.
<i>pPDPTYPE</i>	<ul style="list-style-type: none"> Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile <ul style="list-style-type: none"> 0x00 - PDP-IP (IPv4) 0x01 - PDP-PPP 0x02 - PDP-IPV6 0x03 - PDP-IPV4V6
<i>pPdpHdrComp-Type</i>	<ul style="list-style-type: none"> PDP header compression type <ul style="list-style-type: none"> 0 - PDP header compression is OFF 1 - Manufacturer preferred compression 2 - PDP header compression based on RFC 1144 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095
<i>pPdpDataComp-Type</i>	<ul style="list-style-type: none"> PDP data compression type <ul style="list-style-type: none"> 0 - PDP data compression is OFF 1 - Manufacturer preferred compression 2 - V.42BIS data compression 3 - V.44 data compression
<i>pAPNName</i>	<ul style="list-style-type: none"> Access point name
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 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>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 SLQSGetProfile-Settings().
<i>pPDNInactiv-Timeout</i>	<ul style="list-style-type: none"> Optional 4 uint8_ts indicating the duration of inactivity timer in seconds If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected Default value of zero indicates infinite value This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pAPNClass</i>	<ul style="list-style-type: none"> Optional 1 uint8_t numeric identifier representing the APN in profile Can be set and queried but is not used by the modem This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

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::pIPv6AddrPref`
- 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::pPriDNSIPv4AddrPref`
- 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::pSecDNSIPv4AddrPref`
- 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 * [pIpcpAckTimeout](#)
- uint16_t * [pAuthTimeout](#)
- uint8_t * [pLcpCreqRetryCount](#)
- uint8_t * [pIpcpCreqRetryCount](#)
- 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 * [pPDNIinactivTimeout3GPP2](#)
- 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>pIpcpAckTimeout</i>	<ul style="list-style-type: none"> • IPCP ACK Timeout <ul style="list-style-type: none"> – Value of IPCP ACK Timeout in milliseconds
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> • AUTH Timeout <ul style="list-style-type: none"> – Value of Authentication Timeout in milliseconds
<i>pLcpCreqRetry-Count</i>	<ul style="list-style-type: none"> • LCP Configuration Request Retry Count
<i>pIpcpCreqRetry-Count</i>	<ul style="list-style-type: none"> • IPCP Configuration Request Retry Count
<i>pAuthRetry-Count</i>	<ul style="list-style-type: none"> • Authentication Retry Count value
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> • Authentication Protocol <ul style="list-style-type: none"> – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP
<i>pUserId</i>	<ul style="list-style-type: none"> • User ID to be used during data network authentication • maximum length allowed is 127 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 SLQSGetProfile-Settings().
<i>pAPNEnabled3-GPP2</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled • If disabled, the profile can not be used for making data calls • Values: <ul style="list-style-type: none"> – 0 - Disabled – 1 - Enabled(default value) • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

<i>pPDNInactiv- Timeout3GPP2</i>	<ul style="list-style-type: none"> • Optional 4 uint8_ts indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pAPNClass3GP- P2</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

8.24.2 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 * 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 * plmCnFlag
- 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

8.25.1 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 pAPNName field. Size of this parameter is 2 bytes.
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> Primary DNS IPv4 Address Preference
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> Secondary DNS IPv4 Address Preference
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> UMTS Requested QoS
<i>pUMTSMinQoS</i>	<ul style="list-style-type: none"> UMTS Minimum QoS
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> GPRS Minimum QoS
<i>pUsername</i>	<ul style="list-style-type: none"> User name
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes.
<i>pPassword</i>	<ul style="list-style-type: none"> Password
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes.
<i>pAuthentication-Pref</i>	<ul style="list-style-type: none"> Authentication Preference <ul style="list-style-type: none"> Bit map that indicates the authentication algorithm preference <ul style="list-style-type: none"> Bit 0 - PAP preference <ul style="list-style-type: none"> 0 - PAP is never performed 1 - PAP may be performed Bit 1 - CHAP preference <ul style="list-style-type: none"> 0 - CHAP is never performed 1 - CHAP may be performed If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> IPv4 Address Preference

<i>pPcscfAddr-UsingPCO</i>	<ul style="list-style-type: none"> • P-CSCF Address using PCO Flag <ul style="list-style-type: none"> – 1 - (TRUE) implies request PCSCF address using PCO – 0 - (FALSE) implies do not request By default, this value is 0
<i>pPdpAccess-ConFlag</i>	<ul style="list-style-type: none"> • PDP access control flag <ul style="list-style-type: none"> – 0 - PDP access control none – 1 - PDP access control reject – 2 - PDP access control permission
<i>pPcscfAddr-UsingDhcp</i>	<ul style="list-style-type: none"> • P-CSCF address using DHCP <ul style="list-style-type: none"> – 1 - (TRUE) implies Request PCSCF address using DHCP – 0 - (FALSE) implies do not request By default, value is 0
<i>pImCnFlag</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>pAddrAllocationPref</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>pAPNDisabledFlag</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>pPDNInactivityTimeout</i>	<ul style="list-style-type: none"> Optional 4 Bytes indicating the duration of inactivity timer in seconds If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected Default value of zero indicates infinite value This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().
<i>pAPNClass</i>	<ul style="list-style-type: none"> Optional 1 uint8_t numeric identifier representing the APN in profile Can be set and queried but is not used by the modem This parameter is currently read only and can be read by using the function SLQSGetProfileSettings().

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::pIPv6AddrPref
- 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::pPcsfAddrUsingDhcp
- 8.25.2.15 `uint8_t*` LibPackprofile_3GPP::pPcsfAddrUsingPCO
- 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 *** plpcpAckTimeout
- **uint16_t *** pAuthTimeout
- **uint8_t *** pLcpCreqRetryCount
- **uint8_t *** plpcpCreqRetryCount
- **uint8_t *** pAuthRetryCount
- **uint8_t *** pAuthProtocol
- **uint8_t *** pUserId
- **uint16_t *** pUserIdSize
- **uint8_t *** pAuthPassword
- **uint16_t *** pAuthPassword_tSize
- **uint8_t *** pDataRate
- **uint32_t *** pAppType
- **uint8_t *** pDataMode
- **uint8_t *** pAppPriority
- **uint8_t *** pApnString
- **uint16_t *** pApnStringSize
- **uint8_t *** pPdnType
- **uint8_t *** plsPcscfAddressNedded
- **uint32_t *** pPrimaryV4DnsAddress
- **uint32_t *** pSecondaryV4DnsAddress
- **uint16_t *** pPriV6DnsAddress
- **uint16_t *** pSecV6DnsAddress
- **uint8_t *** pRATType
- **uint8_t *** pAPNEnabled3GPP2
- **uint32_t *** pPDNInactivTimeout3GPP2
- **uint8_t *** pAPNClass3GPP2

8.26.1 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>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 bytes; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
<i>pUserIdSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 bytes.
<i>pAuthPassword</i>	<ul style="list-style-type: none"> • Password to be used during data network authentication; • maximum length allowed is 127 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
<i>pAuthPassword-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 bytes.
<i>pDataRate</i>	<ul style="list-style-type: none"> • Data Rate Requested <ul style="list-style-type: none"> – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2
<i>pAppType</i>	<ul style="list-style-type: none"> • Application Type: <ul style="list-style-type: none"> – 0x00000001 - Default Application Type – 0x00000020 - LBS Application Type – 0x00000040 - Tethered Application Type – This parameter is not used while creating/modifying a profile
<i>pDataMode</i>	<ul style="list-style-type: none"> • Data Mode to use: <ul style="list-style-type: none"> – 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) – 1 - CDMA Only (1X only) – 2 - HDR Only (1xEV-DO only) – Default is 0
<i>pAppPriority</i>	<ul style="list-style-type: none"> • Application Priority <ul style="list-style-type: none"> – Numerical 1 uint8_t value defining the application priority; higher value implies higher priority – This parameter is not used while creating/modifying a profile
<i>pApnString</i>	<ul style="list-style-type: none"> • String representing the Access Point Name • maximum length allowed is 100 bytes • QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.

<i>pApnStringSize;</i>	<ul style="list-style-type: none"> This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 bytes.
<i>pPdnType</i>	<ul style="list-style-type: none"> Packed Data Network Type Requested: <ul style="list-style-type: none"> 0 - IPv4 PDN Type 1 - IPv6 PDN Type 2 - IPv4 or IPv6 PDN Type 3 - Unspecified PDN Type (implying no preference)
<i>plsPcscf-AddressNedded</i>	<ul style="list-style-type: none"> This boolean value is used to control if PCSCF address is requested from PDSN <ul style="list-style-type: none"> 1 -(TRUE) implies request for PCSCF value from the PDSN 0 -(FALSE) implies do not request for PCSCF value from the PDSN
<i>pPrimaryV4Dns-Address</i>	<ul style="list-style-type: none"> IPv4 Primary DNS address <ul style="list-style-type: none"> The Primary IPv4 DNS address that can be statically assigned to the UE
<i>pSecondaryV4-DnsAddress</i>	<ul style="list-style-type: none"> IPv4 Secondary DNS address <ul style="list-style-type: none"> The Secondary IPv4 DNS address that can be statically assigned to the UE
<i>pPriV6Dns-Address</i>	<ul style="list-style-type: none"> Primary IPv6 DNS address <ul style="list-style-type: none"> The Primary IPv6 DNS address that can be statically assigned to the UE
<i>pSecV6Dns-Address</i>	<ul style="list-style-type: none"> Secondary IPv6 DNS address <ul style="list-style-type: none"> The Secondary IPv6 DNS address that can be statically assigned to the UE
<i>pRATType</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating RAT Type Values: <ul style="list-style-type: none"> 1 - HRPD 2 - EHRPD 3 - HRPD_EHRPD This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pAPNEnabled3-GPP2</i>	<ul style="list-style-type: none"> Optional 1 uint8_t Flag indicating if the APN is disabled/enabled If disabled, the profile can not be used for making data calls Values: <ul style="list-style-type: none"> 0 - Disabled 1 - Enabled(default value) This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

<i>pPDNInactiv- Timeout3GPP2</i>	<ul style="list-style-type: none"> • Optional 4 Bytes indicating the duration of inactivity timer in seconds • If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected • Default value of zero indicates infinite value • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().
<i>pAPNClass3GP- P2</i>	<ul style="list-style-type: none"> • Optional 1 uint8_t numeric identifier representing the APN in profile • Can be set and queried but is not used by the modem • This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().

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

<i>gUIBitRate</i>	<ul style="list-style-type: none"> Guaranteed UL bit rate
<i>maxUIBitRate</i>	<ul style="list-style-type: none"> Maximum UL bit rate

8.27.2 Field Documentation

8.27.2.1 `uint8_t LibPackQosClassID::gDIBitRate`

8.27.2.2 `uint32_t LibPackQosClassID::gUIBitRate`

8.27.2.3 `uint32_t LibPackQosClassID::maxDIBitRate`

8.27.2.4 `uint32_t LibPackQosClassID::maxUIBitRate`

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>grntDownlinkBitrate</i>	<ul style="list-style-type: none"> • Guranteed downlink bit rate in bits/sec
<i>qosDeliveryOrder</i>	<ul style="list-style-type: none"> - Qos delivery order • 0x00 - Subscribe • 0x01 - delivery order on • 0x02 - delivery order off
<i>maxSDUSize</i>	<ul style="list-style-type: none"> • Maximum SDU size
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> - SDU error ratio • Target value for fraction of SDUs lost or detected as erroneous. • 0x00 - Subscribe • 0x01 - $1 \cdot 10^{-2}$ • 0x02 - $7 \cdot 10^{-3}$ • 0x03 - $1 \cdot 10^{-3}$ • 0x04 - $1 \cdot 10^{-4}$ • 0x05 - $1 \cdot 10^{-5}$ • 0x06 - $1 \cdot 10^{-6}$ • 0x07 - $1 \cdot 10^{-1}$
<i>resBerRatio</i>	<ul style="list-style-type: none"> - Residual bit error ratio • Target value for undetected bit error ratio in in the delivered SDUs. • 0x00 - Subscribe • 0x01 - $5 \cdot 10^{-2}$ • 0x02 - $1 \cdot 10^{-2}$ • 0x03 - $5 \cdot 10^{-3}$ • 0x04 - $4 \cdot 10^{-3}$ • 0x05 - $1 \cdot 10^{-3}$ • 0x06 - $1 \cdot 10^{-4}$ • 0x07 - $1 \cdot 10^{-5}$ • 0x08 - $1 \cdot 10^{-6}$ • 0x09 - $1 \cdot 10^{-8}$
<i>deliveryErrSDU</i>	<ul style="list-style-type: none"> - Delivery of erroneous SDUs • Indicates whether SDUs detected as erroneous shall be delivered or not. • 0x00 - Subscribe • 0x01 - $5 \cdot 10^{-2}$ • 0x02 - $1 \cdot 10^{-2}$ • 0x03 - $5 \cdot 10^{-3}$ • 0x04 - $4 \cdot 10^{-3}$ • 0x05 - $1 \cdot 10^{-3}$ • 0x06 - $1 \cdot 10^{-4}$ • 0x07 - $1 \cdot 10^{-5}$ • 0x08 - $1 \cdot 10^{-6}$ • 0x09 - $1 \cdot 10^{-8}$

<i>transferDelay</i>	<ul style="list-style-type: none"> - Transfer delay (ms) <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>	<ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> • TRUE - Signaling indication ON • FALSE - Signaling indication OFF
---------------	---

8.30.2 Field Documentation

8.30.2.1 `uint8_t LibPackUMTSReqQoS::SigInd`

8.30.2.2 `LibPackUMTSReqQoS LibPackUMTSReqQoS::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"> gnssSvId deleteSvInfoMask
<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 GLONASS SVDIR – QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete GLONASS SVSTEER – QMI_LOC_MASK_DELETE_GLO_TIME (0x00000040) - Mask to delete GLONASS time – QMI_LOC_MASK_DELETE_GLO_ALM_CORR (0x00000080) - Mask to delete GLONASS almanac correlation – QMI_LOC_MASK_DELETE_SBAS_SVDIR (0x00000100) - Mask to delete SBAS SVDIR – QMI_LOC_MASK_DELETE_SBAS_SVSTEER (0x00000200) - Mask to delete SBAS SVSTEER – QMI_LOC_MASK_DELETE_POSITION (0x00000400) - Mask to delete position estimate – QMI_LOC_MASK_DELETE_TIME (0x00000800) - Mask to delete time estimate – QMI_LOC_MASK_DELETE_IONO (0x00001000) - Mask to delete IONO – QMI_LOC_MASK_DELETE_UTC (0x00002000) - Mask to delete UTC estimate – QMI_LOC_MASK_DELETE_HEALTH (0x00004000) - Mask to delete SV health record – QMI_LOC_MASK_DELETE_SADATA (0x00008000) - Mask to delete SADATA – QMI_LOC_MASK_DELETE_RTI (0x00010000) - Mask to delete RTI – QMI_LOC_MASK_DELETE_SV_NO_EXIST (0x00020000) - Mask to delete SV_NO_EXIST – QMI_LOC_MASK_DELETE_FREQ_BIAS_EST (0x00040000) - Mask to delete frequency bias estimate – QMI_LOC_MASK_DELETE_BDS_SVDIR (0x00080000) - Mask to delete BDS SVDIR – QMI_LOC_MASK_DELETE_BDS_SVSTEER (0x00100000) - Mask to delete BDS SVSTEER – QMI_LOC_MASK_DELETE_BDS_TIME (0x00200000) - Mask to delete BDS time – QMI_LOC_MASK_DELETE_BDS_ALM_CORR (0x00400000) - Mask to delete BDS almanac correlation – QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GPS (0x00800000) - Mask to delete GNSS SV blacklist GPS – QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GLO (0x01000000) - Mask to delete GNSS SV blacklist GLO – QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_BDS (0x02000000) - Mask to delete GNSS SV blacklist BDS
-------------	---

8.35.2 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>appVersion- Length</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) • $PDOP = \text{square root of } (\text{Square of HDOP} + \text{Square of VDOP}^2)$
-------------	---

<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 <code>loc_SV</code> belongs • Valid values: <ul style="list-style-type: none"> – <code>eQMI_LOC_SV_SYSTEM_GPS</code> (1) - GPS satellite – <code>eQMI_LOC_SV_SYSTEM_GALILEO</code> (2) - GALILEO satellite – <code>eQMI_LOC_SV_SYSTEM_SBAS</code> (3) - SBAS satellite – <code>eQMI_LOC_SV_SYSTEM_COMPASS</code> (4) - COMPASS satellite – <code>eQMI_LOC_SV_SYSTEM_GLONASS</code> (5) - GLONASS satellite – <code>eQMI_LOC_SV_SYSTEM_BDS</code> (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"> – <code>0x01</code> - <code>DELETE_EPHEMERIS</code> – <code>0x02</code> - <code>DELETE_ALMANAC</code>

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 lteSSInfo Struct Reference

Data Fields

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

8.43.1 Detailed Description

Parameters

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

8.43.2 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::rssi`

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

8.51.2 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_CDMA RSSI Thresh 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_CDMA RSSI Thresh::CDMARSSIThreshListLen

8.52.2.2 `int16_t *` nas_CDMA RSSI Thresh::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>ccsSupportedValid</i>	<ul style="list-style-type: none"> Indicates whether the supported concurrent service is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>ccsSupported</i>	<ul style="list-style-type: none"> Whether concurrent service is supported. Only applicable for CDMA. <ul style="list-style-type: none"> 0x00 - Not supported 0x01 - Supported 0xFF - Not Available
<i>cdmaSysIdValid</i>	<ul style="list-style-type: none"> Indicates whether the CDMA system ID is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>systemID</i>	<ul style="list-style-type: none"> System ID. <ul style="list-style-type: none"> 0xFFFF - Not Available
<i>networkID</i>	<ul style="list-style-type: none"> Network ID. <ul style="list-style-type: none"> 0xFFFF - Not Available
<i>bsInfoValid</i>	<ul style="list-style-type: none"> Indicates whether the base station information is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>baseLat</i>	<ul style="list-style-type: none"> Base station latitude in units of 0.25 sec. Expressed as a two's complement signed number with positive numbers signifying North latitudes. <ul style="list-style-type: none"> 0xFFFFFFFF - Not Available
<i>baseLong</i>	<ul style="list-style-type: none"> Base station longitude in units of 0.25 sec. Expressed as a two's complement signed number with positive numbers signifying East latitudes. <ul style="list-style-type: none"> 0xFFFFFFFF - Not Available
<i>packetZoneValid</i>	<ul style="list-style-type: none"> Indicates whether the packet zone is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available

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

8.53.2 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_CDMA SysInfo::networkIdValid`
- 8.53.2.16 `uint16_t nas_CDMA SysInfo::packetZone`
- 8.53.2.17 `uint8_t nas_CDMA SysInfo::packetZoneValid`
- 8.53.2.18 `uint8_t nas_CDMA SysInfo::pRevInUse`
- 8.53.2.19 `uint8_t nas_CDMA SysInfo::pRevInUseValid`
- 8.53.2.20 `nas_sysInfoCommon nas_CDMA SysInfo::sysInfoCDMA`
- 8.53.2.21 `uint16_t nas_CDMA SysInfo::systemID`

8.54 nas_CDMA SysInfoExt 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_CDMA SysInfoExt::imsi_11_12`
- 8.54.2.2 `uint16_t nas_CDMA SysInfoExt::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>rssi</i>	<ul style="list-style-type: none"> Current RSSI in 1/10 dBm as measured by L1. Range: -120.0 dBm to 0.
<i>srxlev</i>	<ul style="list-style-type: none"> Cell selection Rx level (Srxlev) value. Range: -128 to 128. This field is only valid when ue_in_idle is TRUE.

8.55.2 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::rssi`

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 [radioIrf](#)

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>radioIrf</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

8.62.2 Field Documentation

8.62.2.1 uint16_t nas_errorRateListElement::errorRate

8.62.2.2 uint8_t nas_errorRateListElement::radioIrf

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

<i>rxLev</i>	<ul style="list-style-type: none"> • Serving Cell Rx measurement. • Values range between 0 and 63. • Mapped to a measured signal level: <ul style="list-style-type: none"> – Rxlev 0 is a signal strength less than -110 dBm – Rxlev 1 is -110 dBm to -109 dBm – Rxlev 2 is -109 dBm to -108 dBm – ... – Rxlev 62 is -49 dBm to -48 dBm – Rxlev 63 is greater than -48 dBm – 0xFFFF - Not Available
<i>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_GSMsysInfo::rejCause`

8.68.2.14 `uint8_t nas_GSMsysInfo::rejectSrvDomain`

8.68.2.15 `nas_sysInfoCommon nas_GSMsysInfo::sysInfoGSM`

8.69 nas_HDRECIOTresh Struct Reference

Data Fields

- `uint8_t` [HDRECIOTreshListLen](#)
- `int16_t *` [pHDRECIOTreshList](#)

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 *` [pHDRIOTreshList](#)

8.70.1 Detailed Description

This structure contains HDR IO threshold related parameters.

Parameters

<i>HDRIOTresh- ListLen</i>	<ul style="list-style-type: none"> Length of the HDR IO threshold list parameter to follow
<i>pHDRIOTresh- 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_HDRIOTresh::HDRIOTreshListLen`8.70.2.2 `int16_t* nas_HDRIOTresh::pHDRIOTreshList`

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>hdrActiveProt-Valid</i>	<ul style="list-style-type: none"> Indicates whether the HDR active protocol revision information is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available
<i>hdrActiveProt</i>	<ul style="list-style-type: none"> HDR active protocol revision information . Only applicable for HDR. <ul style="list-style-type: none"> 0x00 - None 0x02 - HDR Rel 0 0x03 - HDR Rel A 0x04 - HDR Rel B 0xFF - Not Available
<i>is856SysIdValid</i>	<ul style="list-style-type: none"> Indicates whether the IS-856 system ID is valid. <ul style="list-style-type: none"> 0x00 - Invalid 0x01 - Valid 0xFF - Not Available

<i>is856SysId[SLQ-S_SYSTEM_ID_SIZE]</i>	<ul style="list-style-type: none"> • IS-856 system ID. • Only applicable for HDR.
---	---

8.73.2 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 thresh_serving_low, 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 ue_in_idle 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_IteGsmCellInfo 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 <i>n</i> set to 1 means a neighbor with NCC <i>n</i> must be included in the report. This flag is synonymous with a blacklist in other RATs. • This field is only valid when <code>ue_in_idle</code> is TRUE.
<i>cells_len</i>	<ul style="list-style-type: none"> • Provides the number of set of gsm cells.
<i>GsmCellInfo[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[NA-S_MAX_DESC-RIPTION_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 <i>ue_in_idle</i> is TRUE. <ul style="list-style-type: none"> – 0xFF - Not Available
<i>threshServing-Low</i>	<ul style="list-style-type: none"> • Serving cell low threshold. • Range: 0 to 31. • This field is only valid when <i>ue_in_idle</i> is TRUE. <ul style="list-style-type: none"> – 0xFF - Not Available

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

8.78.2 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 nas_LteGsmCellInfo 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_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • See nas_LteWcdmaCellInfo for more information.

8.80.2 Field Documentation

8.80.2.1 `uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen`

8.80.2.2 `nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]`

8.80.2.3 `uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle`

8.81 nas_lteRsrpinformation 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_lteRsrpinformation::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_LE-NGTH]</i>	<ul style="list-style-type: none"> Mobile Country Code. MCC digits in ASCII characters
<i>MNC[PLMN_LE-NGTH]</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_IteWcdmaCellInfo Struct Reference

Data Fields

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

8.89.1 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> is TRUE.

<i>cellsLen</i>	<ul style="list-style-type: none"> Provides the number of set of WCDMA cells.
<i>WCDMACellInfo[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_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>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_PhyCaAggPcellInfo Struct Reference

Data Fields

- uint16_t [pci](#)
- uint16_t [freq](#)
- [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) dl_bw_value
- uint16_t [iLTEbandValue](#)
- uint8_t [TlvPresent](#)

8.93.1 Detailed Description

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

Parameters

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

8.93.2 Field Documentation

8.93.2.1 [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) nas_PhyCaAggPcellInfo::dl_bw_value

8.93.2.2 uint16_t nas_PhyCaAggPcellInfo::freq

8.93.2.3 uint16_t nas_PhyCaAggPcellInfo::iLTEbandValue

8.93.2.4 uint16_t nas_PhyCaAggPcellInfo::pci

8.93.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

8.94 nas_PhyCaAggScellIDIBw Struct Reference

Data Fields

- [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) dl_bw_value

- `uint8_t TlvPresent`

8.94.1 Detailed Description

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

Parameters

<i>dl_bw_value</i>	<ul style="list-style-type: none"> • Downlink Bandwidth Values. • See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information.
--------------------	---

8.94.2 Field Documentation

8.94.2.1 `NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellIDBw::dl_bw_value`

8.94.2.2 `uint8_t nas_PhyCaAggScellIDBw::TlvPresent`

8.95 nas_PhyCaAggScellIndex Struct Reference

Data Fields

- `uint8_t scell_idx`
- `uint8_t TlvPresent`

8.95.1 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.

8.95.2 Field Documentation

8.95.2.1 `uint8_t nas_PhyCaAggScellIndex::scell_idx`

8.95.2.2 `uint8_t nas_PhyCaAggScellIndex::TlvPresent`

8.96 nas_PhyCaAggScellIndType Struct Reference

Data Fields

- `uint16_t pci`
- `uint16_t freq`

- [NAS_LTE_CPHY_SCELL_STATE_LITE](#) `scell_state`
- `uint8_t` `TlvPresent`

8.96.1 Detailed Description

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

Parameters

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

8.96.2 Field Documentation

8.96.2.1 `uint16_t` `nas_PhyCaAggScellIndType::freq`

8.96.2.2 `uint16_t` `nas_PhyCaAggScellIndType::pci`

8.96.2.3 `NAS_LTE_CPHY_SCELL_STATE_LITE` `nas_PhyCaAggScellIndType::scell_state`

8.96.2.4 `uint8_t` `nas_PhyCaAggScellIndType::TlvPresent`

8.97 nas_PhyCaAggScellInfo Struct Reference

Data Fields

- `uint16_t` `pci`
- `uint16_t` `freq`
- `NAS_LTE_CPHY_CA_BW_NRB_LITE` `dl_bw_value`
- `uint16_t` `iLTEbandValue`
- `NAS_LTE_CPHY_SCELL_STATE_LITE` `scell_state`
- `uint8_t` `TlvPresent`

8.97.1 Detailed Description

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

Parameters

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

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

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

8.97.2 Field Documentation

8.97.2.1 **NAS_LTE_CPHY_CA_BW_NRB_LITE** nas_PhyCaAggScellInfo::dl_bw_value

8.97.2.2 uint16_t nas_PhyCaAggScellInfo::freq

8.97.2.3 uint16_t nas_PhyCaAggScellInfo::ltebandValue

8.97.2.4 uint16_t nas_PhyCaAggScellInfo::pci

8.97.2.5 **NAS_LTE_CPHY_SCELL_STATE_LITE** nas_PhyCaAggScellInfo::scell_state

8.97.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

8.98 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

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

8.98.1 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)

8.98.2 Field Documentation

8.98.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.98.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.98.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

8.99 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]

8.99.1 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

8.99.2 Field Documentation

8.99.2.1 char nas_QmiNas3GppNetworkInfo::Description[255]

8.99.2.2 uint32_t nas_QmiNas3GppNetworkInfo::Forbidden

8.99.2.3 uint32_t nas_QmiNas3GppNetworkInfo::InUse

8.99.2.4 uint16_t nas_QmiNas3GppNetworkInfo::MCC

8.99.2.5 uint16_t nas_QmiNas3GppNetworkInfo::MNC

8.99.2.6 uint32_t nas_QmiNas3GppNetworkInfo::Preferred

8.99.2.7 uint32_t nas_QmiNas3GppNetworkInfo::Roaming

8.100 nas_QmiNas3GppNetworkRAT Struct Reference

Data Fields

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

8.100.1 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

8.100.2 Field Documentation

8.100.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.100.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.100.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.101 nas_QmisNasPcsDigit Struct Reference

Data Fields

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

8.101.1 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

8.101.2 Field Documentation

8.101.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

8.101.2.2 uint16_t nas_QmisNasPcsDigit::MCC

8.101.2.3 uint16_t nas_QmisNasPcsDigit::MNC

8.102 nas_RejectReasonTlv Struct Reference

Data Fields

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

8.102.1 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

8.102.2 Field Documentation

8.102.2.1 uint32_t nas_RejectReasonTlv::rejectCause

8.102.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

8.102.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

8.103 nas_RFInfoTlv Struct Reference

Data Fields

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

8.103.1 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

8.103.2 Field Documentation

8.103.2.1 uint32_t nas_RFInfoTlv::activeBandClass[255]

8.103.2.2 `uint32_t nas_RFInfoTlv::activeChannel[255]`

8.103.2.3 `uint32_t nas_RFInfoTlv::radiolInterface[255]`

8.103.2.4 `uint8_t nas_RFInfoTlv::radiolInterfaceSize`

8.103.2.5 `uint8_t nas_RFInfoTlv::TlvPresent`

8.104 nas_roamIndList Struct Reference

Data Fields

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

8.104.1 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>radiolInterface</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_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
<i>roamIndicator</i>	<ul style="list-style-type: none"> • Roaming Indicator • Values: <ul style="list-style-type: none"> – 0x00 - Roaming – 0x01 - Home

8.104.2 Field Documentation

8.104.2.1 `uint8_t nas_roamIndList::numInstances`

8.104.2.2 `uint8_t nas_roamIndList::radiolInterface[32]`

8.104.2.3 uint8_t nas_roamIndList::roamIndicator[32]

8.105 nas_rsrqInformation Struct Reference

Data Fields

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

8.105.1 Detailed Description

This structure contains the RSRQ Information

Parameters

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

8.105.2 Field Documentation

8.105.2.1 uint8_t nas_rsrqInformation::radiolf

8.105.2.2 int8_t nas_rsrqInformation::rsrq

8.106 nas_rxSignalStrengthListElement Struct Reference

Data Fields

- int16_t [rxSignalStrength](#)
- uint8_t [radiolf](#)

8.106.1 Detailed Description

This structure contains the Received Signal Strength Information

Parameters

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

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

Note

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

8.106.2 Field Documentation

8.106.2.1 `uint8_t nas_rxSignalStrengthListElement::radioIf`

8.106.2.2 `int16_t nas_rxSignalStrengthListElement::rxSignalStrength`

8.107 nas_servSystem Struct Reference**Data Fields**

- `uint8_t regState`
- `uint8_t csAttachState`
- `uint8_t psAttachState`
- `uint8_t selNetwork`
- `uint8_t numRadiolInterfaces`
- `uint8_t radioInterface` [32]

8.107.1 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

8.107.2 Field Documentation

8.107.2.1 `uint8_t nas_servSystem::csAttachState`

8.107.2.2 `uint8_t nas_servSystem::numRadioInterfaces`

8.107.2.3 `uint8_t nas_servSystem::psAttachState`

8.107.2.4 `uint8_t nas_servSystem::radioInterface[32]`

8.107.2.5 `uint8_t nas_servSystem::regState`

8.107.2.6 `uint8_t nas_servSystem::selNetwork`

8.108 nas_SignalStrengthTlv Struct Reference

Data Fields

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

8.108.1 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

8.108.2 Field Documentation

8.108.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

8.108.2.2 int8_t nas_SignalStrengthTlv::signalStrength

8.108.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

8.109 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](#)

8.109.1 Detailed Description

Parameters

<i>rxSignalStrengthDelta</i>	RSSI delta(in dBm) at which an event report indication
<i>ecioDelta</i>	ecio delta
<i>ioDelta</i>	io delta
<i>sinrDelta</i>	sinr delta
<i>rsrqDelta</i>	rsrq delta
<i>ecioThresholdListLen</i>	
<i>ecioThresholdList</i>	

<i>sinrThreshold-ListLen</i>	
<i>sinrThreshold-List</i>	
<i>lteSnrDelta</i>	lte snr delta
<i>lteRsrpDelta</i>	lte rsrp delta

8.109.2 Field Documentation

- 8.109.2.1 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta`
- 8.109.2.2 `int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]`
- 8.109.2.3 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen`
- 8.109.2.4 `uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta`
- 8.109.2.5 `uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta`
- 8.109.2.6 `uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta`
- 8.109.2.7 `uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta`
- 8.109.2.8 `uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta`
- 8.109.2.9 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta`
- 8.109.2.10 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]`
- 8.109.2.11 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen`

8.110 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](#)

8.110.1 Detailed Description

Parameters

<i>rxSignal- StrengthInfo</i>	signal strength info list
<i>ecioInfo</i>	ecio info list
<i>io</i>	received IO in dBm; IO is only applicable for 1xEV-DO
<i>sinr</i>	SINR level
<i>errorRateInfo</i>	error rate info

<i>rsrqInfo</i>	rsrq info
<i>lteSnrinfo</i>	lte Snr information
<i>lteRsrpinfo</i>	lte rsrp info

8.110.2 Field Documentation

8.110.2.1 **nas_ecioListElement** nas_SLQSSignalStrengthsInformation::eciInfo

8.110.2.2 **nas_errorRateListElement** nas_SLQSSignalStrengthsInformation::errorRateInfo

8.110.2.3 **uint32_t** nas_SLQSSignalStrengthsInformation::io

8.110.2.4 **nas_lteRsrpinformation** nas_SLQSSignalStrengthsInformation::lteRsrpinfo

8.110.2.5 **nas_lteSnrinformation** nas_SLQSSignalStrengthsInformation::lteSnrinfo

8.110.2.6 **nas_rsrqInformation** nas_SLQSSignalStrengthsInformation::rsrqInfo

8.110.2.7 **nas_rxSignalStrengthListElement** nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo

8.110.2.8 **uint8_t** nas_SLQSSignalStrengthsInformation::sinr

8.111 nas_SLQSSignalStrengthsTlv Struct Reference

Data Fields

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

8.111.1 Detailed Description

Parameters

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

8.111.2 Field Documentation

8.111.2.1 **nas_SLQSSignalStrengthsInformation** nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

8.111.2.2 **uint8_t** nas_SLQSSignalStrengthsTlv::TlvPresent

8.112 nas_SrvStatusInfo Struct Reference

Data Fields

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

8.112.1 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

8.112.2 Field Documentation

8.112.2.1 `uint8_t nas_SrvStatusInfo::isPrefDataPath`

8.112.2.2 `uint8_t nas_SrvStatusInfo::srvStatus`

8.113 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`

8.113.1 Detailed Description

Structure for storing the System Information common to CDMA, HDR, GSM, WCDMA and LTE networks.

Parameters

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

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

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

8.113.2 Field Documentation

8.113.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

8.113.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

8.113.2.3 uint8_t nas_sysInfoCommon::roamStatus

8.113.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

8.113.2.5 uint8_t nas_sysInfoCommon::srvCapability

8.113.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

8.113.2.7 uint8_t nas_sysInfoCommon::srvDomain

8.113.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

8.114 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

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

8.114.1 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.

8.114.2 Field Documentation

8.114.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.114.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.115 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

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

8.115.1 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).

8.115.2 Field Documentation

8.115.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.115.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

8.116 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

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

8.116.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

Parameters

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

8.116.2 Field Documentation

8.116.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.116.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

8.117 nas_TDSCDMASINRThresh Struct Reference

Data Fields

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

8.117.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

Parameters

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

8.117.2 Field Documentation

8.117.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.117.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.118 nas_timeInfo Struct Reference

Data Fields

- uint16_t [year](#)
- uint8_t [month](#)
- uint8_t [day](#)
- uint8_t [hour](#)
- uint8_t [minute](#)
- uint8_t [second](#)
- uint8_t [dayOfWeek](#)
- int8_t [timeZone](#)
- uint8_t [dayLtSavingAdj](#)
- uint8_t [radioInterface](#)
- uint8_t [TlvPresent](#)

8.118.1 Detailed Description

This structure contains the parameters for Network Time.

Parameters

<i>year</i>	<ul style="list-style-type: none"> • Year
<i>month</i>	<ul style="list-style-type: none"> • Month • 1 is January and 12 is December
<i>day</i>	<ul style="list-style-type: none"> • Day • Range - 1 to 31
<i>hour</i>	<ul style="list-style-type: none"> • Hour • Range - 0 to 59
<i>minute</i>	<ul style="list-style-type: none"> • Minute • Range - 0 to 59
<i>second</i>	<ul style="list-style-type: none"> • Second • Range - 0 to 59
<i>dayOfWeek</i>	<ul style="list-style-type: none"> • Day of the week • 0 is Monday and 6 is Sunday

<i>timeZone</i>	<ul style="list-style-type: none"> • Offset from Universal time • The difference between local time and Universal time, in increments of 15 min • Signed Value
<i>dayLtSavingAdj</i>	<ul style="list-style-type: none"> • Daylight saving adjustment in hours • Possible values - 0, 1, and 2. • This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVD0
<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio interface from which the information comes • Values <ul style="list-style-type: none"> – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO) – 0x04 - NAS_RADIO_IF_GSM - GSM – 0x05 - NAS_RADIO_IF_UMTS - UMTS – 0x08 - NAS_RADIO_IF_LTE - LTE – 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.118.2 Field Documentation

8.118.2.1 uint8_t nas_timeInfo::day

8.118.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

8.118.2.3 uint8_t nas_timeInfo::dayOfWeek

8.118.2.4 uint8_t nas_timeInfo::hour

8.118.2.5 uint8_t nas_timeInfo::minute

8.118.2.6 uint8_t nas_timeInfo::month

8.118.2.7 uint8_t nas_timeInfo::radioInterface

8.118.2.8 uint8_t nas_timeInfo::second

8.118.2.9 int8_t nas_timeInfo::timeZone

8.118.2.10 uint8_t nas_timeInfo::TlvPresent

8.118.2.11 uint16_t nas_timeInfo::year

8.119 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]

8.119.1 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</i> [<i>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</i> [N- AS_MAX_DES- CRIPTION_LE- NGTH]	<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</i> [N- AS_MAX_DES- CRIPTION_LE- NGTH]	<ul style="list-style-type: none"> • See nas_geranInstInfo for more information.

8.119.2 Field Documentation

8.119.2.1 `uint16_t nas_UMTSInfo::cellID`

8.119.2.2 `int16_t nas_UMTSInfo::ecio`

8.119.2.3 `uint8_t nas_UMTSInfo::geranInst`

8.119.2.4 `nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]`

8.119.2.5 `uint16_t nas_UMTSInfo::lac`

8.119.2.6 `uint8_t nas_UMTSInfo::plmn[3]`

8.119.2.7 `uint16_t nas_UMTSInfo::psc`

8.119.2.8 `int16_t nas_UMTSInfo::rscp`

8.119.2.9 `uint16_t nas_UMTSInfo::uarfcn`

8.119.2.10 `uint8_t nas_UMTSInfo::umtsInst`

8.119.2.11 `nas_UMTSInstInfo nas_UMTSInfo::UMTSInstInfo[255]`

8.120 nas_UMTSInstInfo Struct Reference

Data Fields

- `uint16_t umtsUarfcn`
- `uint16_t umtsPsc`
- `int16_t umtsRscp`
- `int16_t umtsEcio`

8.120.1 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).

8.120.2 Field Documentation

8.120.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.120.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.120.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.120.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.121 nas_umtsLTENbrCell Struct Reference

Data Fields

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

8.121.1 Detailed Description

This structure contains information about the UMTS LTE neighbour Cell.

Parameters

<i>earfcn</i>	<ul style="list-style-type: none"> • E-UTRA absolute RF channel number of the detected cell.
<i>pci</i>	<ul style="list-style-type: none"> • Physical cell ID of the detected cell. • Range is defined in 3GPP TS 36.211
<i>rsrp</i>	<ul style="list-style-type: none"> • Current received signal strength indication (in dBm) of the detected cell.
<i>rsrq</i>	<ul style="list-style-type: none"> • Current reference signal received quality (in dB) of the detected cell.

<i>srxlev</i>	<ul style="list-style-type: none"> Cell selection Rx level (Srxlev) value of the detected cell in linear scale. This field is only valid when wcdma_rrc_state is not NAS_WCDMA_RRC_STATE_CEL_FACH or NAS_WCDMA_RRC_STATE_CELL_DCH.
<i>cellsTDD</i>	<ul style="list-style-type: none"> TRUE if the cell is TDD; FALSE if the cell is FDD.

8.121.2 Field Documentation

8.121.2.1 uint8_t nas_umtsLTENbrCell::cellsTDD

8.121.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.121.2.3 uint16_t nas_umtsLTENbrCell::pci

8.121.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.121.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.121.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.122 nas_UniversalTime Struct Reference

Data Fields

- uint16_t [year](#)
- uint8_t [month](#)
- uint8_t [day](#)
- uint8_t [hour](#)
- uint8_t [minute](#)
- uint8_t [second](#)
- uint8_t [dayOfWeek](#)

8.122.1 Detailed Description

This structure contains the parameters for Universal Time Information.

Parameters

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

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

8.122.2 Field Documentation

8.122.2.1 `uint8_t nas_UniversalTime::day`

8.122.2.2 `uint8_t nas_UniversalTime::dayOfWeek`

8.122.2.3 `uint8_t nas_UniversalTime::hour`

8.122.2.4 `uint8_t nas_UniversalTime::minute`

8.122.2.5 `uint8_t nas_UniversalTime::month`

8.122.2.6 `uint8_t nas_UniversalTime::second`

8.122.2.7 `uint16_t nas_UniversalTime::year`

8.123 `nas_wcdmaCellInfo` Struct Reference

Data Fields

- `uint16_t psc`
- `int16_t cpich_rscp`
- `int16_t cpich_ecno`
- `int16_t srxlev`

8.123.1 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.

8.123.2 Field Documentation

8.123.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.123.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.123.2.3 uint16_t nas_wcdmaCellInfo::psc

8.123.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.124 nas_WCDMAECIOThresh Struct Reference

Data Fields

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

8.124.1 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)

8.124.2 Field Documentation

8.124.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.124.2.2 `uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen`

8.125 `nas_WCDMAInfoLTENeighborCell` Struct Reference

Data Fields

- `uint32_t wcdmaRRCTestate`
- `uint8_t umtsLTENbrCellLen`
- `nas_umtsLTENbrCell UMTSLTENbrCell` [255]

8.125.1 Detailed Description

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

Parameters

<i>wcdmaRRCTestate</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 * WCDMA RRC State is IDLE – 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH * WCDMA RRC state is CELL_PCH – 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH * WCDMA RRC state is URA_PCH – 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH * WCDMA RRC state is CELL_FACH – 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH * WCDMA RRC state is CELL_DCH
<i>umtsLTENbrCellLen</i>	<ul style="list-style-type: none"> • Number of sets of UMTS LTE Neighbors.
<i>UMTSLTENbrCell</i>	<ul style="list-style-type: none"> • See <code>nas_umtsLTENbrCell</code> for more information.

8.125.2 Field Documentation

8.125.2.1 `nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell`[255]

8.125.2.2 `uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen`

8.125.2.3 `uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCTestate`

8.126 `nas_WCDMARSSIThresh` Struct Reference

Data Fields

- `uint8_t WCDMARSSIThreshListLen`
- `int16_t * pWCDMARSSIThreshList`

8.126.1 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)

8.126.2 Field Documentation

8.126.2.1 `int16_t` `nas_WCDMARSSIThresh::pWCDMARSSIThreshList`

8.126.2.2 `uint8_t` `nas_WCDMARSSIThresh::WCDMARSSIThreshListLen`

8.127 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`

8.127.1 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>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_LE-NGTH]</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>hsCallStatusValid</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
------------	--

8.127.2 Field Documentation

- 8.127.2.1 `uint32_t nas_WCDMASysInfo::cellId`
- 8.127.2.2 `uint8_t nas_WCDMASysInfo::cellIdValid`
- 8.127.2.3 `uint8_t nas_WCDMASysInfo::hsCallStatus`
- 8.127.2.4 `uint8_t nas_WCDMASysInfo::hsCallStatusValid`
- 8.127.2.5 `uint8_t nas_WCDMASysInfo::hsInd`
- 8.127.2.6 `uint8_t nas_WCDMASysInfo::hsIndValid`
- 8.127.2.7 `uint16_t nas_WCDMASysInfo::lac`
- 8.127.2.8 `uint8_t nas_WCDMASysInfo::lacValid`
- 8.127.2.9 `uint8_t nas_WCDMASysInfo::MCC[3]`
- 8.127.2.10 `uint8_t nas_WCDMASysInfo::MNC[3]`
- 8.127.2.11 `uint8_t nas_WCDMASysInfo::networkIdValid`
- 8.127.2.12 `uint16_t nas_WCDMASysInfo::psc`
- 8.127.2.13 `uint8_t nas_WCDMASysInfo::pscValid`
- 8.127.2.14 `uint8_t nas_WCDMASysInfo::regRejectInfoValid`
- 8.127.2.15 `uint8_t nas_WCDMASysInfo::rejCause`
- 8.127.2.16 `uint8_t nas_WCDMASysInfo::rejectSrvDomain`
- 8.127.2.17 `nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA`

8.128 NASBandPreferenceTlv Struct Reference

Data Fields

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

8.128.1 Field Documentation

- 8.128.1.1 `uint64_t NASBandPreferenceTlv::band_pref`

8.128.1.2 `uint8_t` NASBandPreferenceTlv::TlvPresent

8.129 NASEmergencyModeTlv Struct Reference

Data Fields

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

8.129.1 Field Documentation

8.129.1.1 `uint8_t` NASEmergencyModeTlv::EmerMode

8.129.1.2 `uint8_t` NASEmergencyModeTlv::TlvPresent

8.130 NasGetLTECphyCalInfo Struct Reference

Data Fields

- [NASPhyCaAggScellIndType](#) `PhyCaAggScellIndType`
- [NASPhyCaAggScellDIBw](#) `PhyCaAggScellDIBw`
- [NASPhyCaAggScellInfo](#) `PhyCaAggScellInfo`
- [NASPhyCaAggPcellInfo](#) `PhyCaAggPcellInfo`
- [NASPhyCaAggScellIndex](#) `PhyCaAggScellIndex`

8.130.1 Field Documentation

8.130.1.1 `NASPhyCaAggPcellInfo` NasGetLTECphyCalInfo::PhyCaAggPcellInfo

8.130.1.2 `NASPhyCaAggScellDIBw` NasGetLTECphyCalInfo::PhyCaAggScellDIBw

8.130.1.3 `NASPhyCaAggScellIndex` NasGetLTECphyCalInfo::PhyCaAggScellIndex

8.130.1.4 `NASPhyCaAggScellIndType` NasGetLTECphyCalInfo::PhyCaAggScellIndType

8.130.1.5 `NASPhyCaAggScellInfo` NasGetLTECphyCalInfo::PhyCaAggScellInfo

8.131 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

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

8.131.1 Field Documentation

8.131.1.1 `uint32_t` NASGWAcqOrderPrefTlv::GWAcqOrderPref

8.131.1.2 `uint8_t` NASGWAcqOrderPrefTlv::TlvPresent

8.132 NASLTEBandPreferenceTlv Struct Reference

Data Fields

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

8.132.1 Field Documentation

8.132.1.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

8.132.1.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

8.133 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- uint8_t [nas_release](#)
- uint8_t [nas_major](#)
- uint8_t [nas_minor](#)

8.133.1 Field Documentation

8.133.1.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

8.133.1.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

8.133.1.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

8.133.1.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

8.134 NASModePreferenceTlv Struct Reference

Data Fields

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

8.134.1 Field Documentation

8.134.1.1 uint16_t NASModePreferenceTlv::ModePref

8.134.1.2 uint8_t NASModePreferenceTlv::TlvPresent

8.135 NASNetSelPreferenceTlv Struct Reference

Data Fields

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

8.135.1 Field Documentation

8.135.1.1 `uint8_t` NASNetSelPreferenceTlv::NetSelPref

8.135.1.2 `uint8_t` NASNetSelPreferenceTlv::TlvPresent

8.136 NASOTAMessageTlv Struct Reference

Data Fields

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

8.136.1 Field Documentation

8.136.1.1 `uint8_t` NASOTAMessageTlv::data_buf[2048]

8.136.1.2 `uint16_t` NASOTAMessageTlv::data_len

8.136.1.3 `uint32_t` NASOTAMessageTlv::message_type

8.136.1.4 `uint8_t` NASOTAMessageTlv::TlvPresent

8.137 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](#)

8.137.1 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>cell_state</i>	<ul style="list-style-type: none"> • Cell state Values. • See NAS_LTE_CPHY_CELL_STATE for more information.
<i>TlvPresent</i>	<ul style="list-style-type: none"> • Tlv Present.

8.137.2 Field Documentation

8.137.2.1 LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

8.137.2.2 uint32_t NASPhyCaAggPcellInfo::freq

8.137.2.3 uint32_t NASPhyCaAggPcellInfo::lTEbandValue

8.137.2.4 uint32_t NASPhyCaAggPcellInfo::pci

8.137.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

8.138 NASPhyCaAggScellIDIBw Struct Reference

Data Fields

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

8.138.1 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.
--------------------	--

8.138.2 Field Documentation

8.138.2.1 LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellIDIBw::dl_bw_value

8.138.2.2 uint8_t NASPhyCaAggScellIDIBw::TlvPresent

8.139 NASPhyCaAggScellIndex Struct Reference

Data Fields

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

8.139.1 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.

8.139.2 Field Documentation

8.139.2.1 `uint8_t NASPhyCaAggScellIndex::scell_idx`

8.139.2.2 `uint8_t NASPhyCaAggScellIndex::TlvPresent`

8.140 NASPhyCaAggScellIndType Struct Reference

Data Fields

- `uint32_t pci`
- `uint32_t freq`
- `LIBPACK_NAS_LTE_CPHY_SCELL_STATE scell_state`
- `uint8_t TlvPresent`

8.140.1 Detailed Description

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

Parameters

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

8.140.2 Field Documentation

8.140.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.140.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.140.2.3 LIBPACK_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.140.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.141 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](#)

8.141.1 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.

8.141.2 Field Documentation

8.141.2.1 LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

8.141.2.2 uint32_t NASPhyCaAggScellInfo::freq

8.141.2.3 uint32_t NASPhyCaAggScellInfo::ltebandValue

8.141.2.4 uint32_t NASPhyCaAggScellInfo::pci

8.141.2.5 LIBPACK_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

8.141.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

8.142 NASPRLPreferenceTlv Struct Reference

Data Fields

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

8.142.1 Field Documentation

8.142.1.1 uint16_t NASPRLPreferenceTlv::PRLPref

8.142.1.2 uint8_t NASPRLPreferenceTlv::TlvPresent

8.143 NASQmiCbkJnasSwtOTAMessageInd Struct Reference

Data Fields

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

8.143.1 Field Documentation

8.143.1.1 [NASLteNasReleaseInfoTlv](#) NASQmiCbkJnasSwtOTAMessageInd::nasRelInfoTlv

8.143.1.2 [NASOTAMessageTlv](#) NASQmiCbkJnasSwtOTAMessageInd::otaMsgTlv

8.143.1.3 [NASTimeInfoTlv](#) NASQmiCbkJnasSwtOTAMessageInd::timeTlv

8.144 NASQmiCbkJnasSystemSelPrefInd Struct Reference

Data Fields

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

8.144.1 Field Documentation

- 8.144.1.1 **NASBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::BPTlv
- 8.144.1.2 **NASEmergencyModeTlv** NASQmiCbkNasSystemSelPrefInd::EMTlv
- 8.144.1.3 **NASGWAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::GWAOPTlv
- 8.144.1.4 **NASLTEBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::LBPTlv
- 8.144.1.5 **NASModePreferenceTlv** NASQmiCbkNasSystemSelPrefInd::MPTlv
- 8.144.1.6 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv
- 8.144.1.7 **NASPRLPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv
- 8.144.1.8 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv
- 8.144.1.9 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.145 NASRoamPreferenceTlv Struct Reference

Data Fields

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

8.145.1 Field Documentation

- 8.145.1.1 uint16_t NASRoamPreferenceTlv::RoamPref
- 8.145.1.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.146 NASServDomainPrefTlv Struct Reference

Data Fields

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

8.146.1 Field Documentation

- 8.146.1.1 uint32_t NASServDomainPrefTlv::SrvDomainPref
- 8.146.1.2 uint8_t NASServDomainPrefTlv::TlvPresent

8.147 NASServingSystemInfo Struct Reference

Data Fields

- uint8_t [registrationState](#)
- uint8_t [csAttachState](#)

- uint8_t [psAttachState](#)
- uint8_t [selectedNetwork](#)
- uint8_t [radioInterfaceNo](#)
- uint8_t [radioInterfaceList](#) [255]
- uint8_t [hdrPersonality](#)

8.147.1 Detailed Description

This structure will hold the serving system parameters information

Parameters

<i>registrationState</i>	- Registration state of the mobile <ul style="list-style-type: none"> • 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service • 1 - QMI_NAS_REGISTERED Registered with a network • 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service • 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network • 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown
<i>csAttachState</i>	- Circuit Switch domain attach state of the mobile <ul style="list-style-type: none"> • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
<i>psAttachState</i>	- Packet domain attach state of the mobile <ul style="list-style-type: none"> • 0 - Unknown or not applicable • 1 - Attached • 2 - Detached
<i>selectedNetwork</i>	- Type of selected radio access network <ul style="list-style-type: none"> • 0x00 - Unknown • 0x01 - 3GPP2 network • 0x02 - 3GPP network
<i>radioInterfaceNo</i>	- Number of radio interfaces currently in use; this indicates how many radio_if identifiers follow this field
<i>radioInterface-List</i>	- Radio interface currently in use (each is 1 byte) <ul style="list-style-type: none"> • 0x00 - None (no service) • 0x01 - cdma2000 1X • 0x02 - cdma2000 HRPD (1xEV-DO) • 0x03 - AMPS • 0x04 - GSM • 0x05 - UMTS • 0x08 - LTE
<i>hdrPersonality</i>	- HDR personality information (valid only for EVDO) <ul style="list-style-type: none"> • 0x00 - Unknown • 0x01 - HRPD • 0x02 - eHRPD

Note: None

8.147.2 Field Documentation

8.147.2.1 uint8_t NAServingSystemInfo::csAttachState

8.147.2.2 uint8_t NAServingSystemInfo::hdrPersonality

8.147.2.3 uint8_t NAServingSystemInfo::psAttachState

8.147.2.4 uint8_t NAServingSystemInfo::radioInterfaceList[255]

8.147.2.5 uint8_t NAServingSystemInfo::radioInterfaceNo

8.147.2.6 uint8_t NAServingSystemInfo::registrationState

8.147.2.7 uint8_t NAServingSystemInfo::selectedNetwork

8.148 NASTimeInfoTlv Struct Reference

Data Fields

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

8.148.1 Field Documentation

8.148.1.1 uint64_t NASTimeInfoTlv::time

8.148.1.2 uint8_t NASTimeInfoTlv::TlvPresent

8.149 newMTMessageTlv Struct Reference

Data Fields

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

8.149.1 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

8.149.2 Field Documentation

8.149.2.1 `sMSMTMessageInfo` `newMTMessageTlv::MTMessageInfo`

8.149.2.2 `uint8_t` `newMTMessageTlv::TlvPresent`

8.150 pack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

- `uint8_t` `cust_id` [64+1]
- `uint8_t` `list_type`
- `uint16_t` `Tlvresult`

8.150.1 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

8.150.2 Field Documentation

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

8.150.2.2 `uint8_t` `pack_dms_GetCustFeaturesV2_t::list_type`

8.150.2.3 `uint16_t` `pack_dms_GetCustFeaturesV2_t::Tlvresult`

8.151 pack_dms_SetCrashAction_t Struct Reference

Data Fields

- `uint8_t` `crashAction`

8.151.1 Detailed Description

Modem action in case of a crash

Parameters

<i>crashAction</i>	<ul style="list-style-type: none"> • 0 - USB Memory Download. Modem will reset after a crash and will stay in USB download mode with only DM port enumerated. • 1 - Reset. Modem will reset and come back in ONLINE mode. Minimal crash data will be available and can be extracted with at!gcdump? AT command • 2 - No Action
--------------------	---

8.151.2 Field Documentation

8.151.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

8.152 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](#)

8.152.1 Field Documentation

8.152.1.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

8.152.1.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

8.152.1.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

8.152.1.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

8.152.1.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

8.152.1.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

8.152.1.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

8.152.1.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

8.152.1.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

8.153 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](#)

8.153.1 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

8.153.2 Field Documentation

8.153.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.153.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.153.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.153.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.154 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t [mode](#)

8.154.1 Field Documentation

8.154.1.1 uint8_t pack_dms_SetEventReport_t::mode

8.155 pack_dms_SetPower_t Struct Reference

Data Fields

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

8.155.1 Field Documentation

8.155.1.1 uint32_t pack_dms_SetPower_t::mode

8.155.1.2 uint16_t pack_dms_SetPower_t::Tlvresult

8.156 pack_dms_SetUSBComp_t Struct Reference

Data Fields

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

8.156.1 Field Documentation

8.156.1.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

8.156.1.2 uint8_t pack_dms_SetUSBComp_t::USBComp

8.157 pack_dms_SLQSDmsSwilIndicationRegister_t Struct Reference

Data Fields

- uint8_t [resetInfoInd](#)

8.157.1 Detailed Description

Parameters

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

8.157.2 Field Documentation

8.157.2.1 uint8_t pack_dms_SLQSDmsSwilIndicationRegister_t::resetInfoInd

8.158 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

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

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

8.158.2 Field Documentation

8.158.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.158.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.159 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.159.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

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

8.159.2 Field Documentation

8.159.2.1 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

8.160 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.160.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

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

8.160.2 Field Documentation

8.160.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

8.161 pack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.161.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

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

8.161.2 Field Documentation

8.161.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

8.162 pack_fms_SetImagesPreference_t Struct Reference

Data Fields

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

8.162.1 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 - Focrce Download.
<i>modemindex</i>	<ul style="list-style-type: none"> • Modem Index.
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Unpack result

8.162.2 Field Documentation

8.162.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

8.162.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

8.162.2.3 uint8_t pack_fms_SetImagesPreference_t::modemIndex

8.162.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList

8.162.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

8.163 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](#)

8.163.1 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.

8.163.2 Field Documentation

8.163.2.1 [loc_BdsSVInfo](#)* [pack_loc_Delete_Assist_Data_t](#)::[pBdsSVInfo](#)

8.163.2.2 [loc_CellDb](#)* [pack_loc_Delete_Assist_Data_t](#)::[pCellDb](#)

8.163.2.3 `loc_ClkInfo*` `pack_loc_Delete_Assist_Data_t::pClkInfo`

8.163.2.4 `loc_GnssData*` `pack_loc_Delete_Assist_Data_t::pGnssData`

8.163.2.5 `loc_SVInfo*` `pack_loc_Delete_Assist_Data_t::pSVInfo`

8.163.2.6 `uint16_t` `pack_loc_Delete_Assist_Data_t::Tlvresult`

8.164 `pack_loc_EventRegister_t` Struct Reference

Data Fields

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

8.164.1 Detailed Description

This structure contains the Parameter for RegisterEvents

Parameters

<i>eventRegister</i>	<ul style="list-style-type: none"> • Specifies the events that the control point is interested in receiving. -Values <ul style="list-style-type: none"> – 0x00000001 - to receive position report event indications – 0x00000002 - to receive satellite report event indications. These reports are sent at a 1 Hz rate. – 0x00000004 - to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate. – 0x00000008 - to receive NI Notify/Verify request event indications – 0x00000010 - to receive time injection request event indications. – 0x00000020 - to receive predicted orbits request event indications. – 0x00000040 - to receive position injection request event indications. – 0x00000080 - to receive engine state report event indications. – 0x00000100 - to receive fix session status report event indications. – 0x00000200 - to receive Wi-Fi position request event indications. – 0x00000400 - to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.). – 0x00000800 - to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock. – 0x00001000 - to receive Stationary Position Indicator (SPI) streaming report indications. – 0x00002000 - to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server. – 0x00004000 - to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited. – 0x00008000 - to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, e.g., if GPS is turned off or if the network is unavailable. – 0x00010000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence. – 0x00020000 - to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports. – 0x00040000 - to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data. – 0x00080000 - to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session. – 0x00100000 - to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing. – 0x00200000 - to receive Wi-Fi Access Point (AP) data inject request event indications. – 0x00400000 - to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification. – 0x00800000 - to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.). – 0x01000000 - to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). – 0x02000000 - to receive satellite position reports as polynomials. Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG.
----------------------	--

Note

Multiple events can be registered by OR the individual masks and sending them in this TLV. All unused bits in this mask must be set to 0.

Parameters

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

8.164.2 Field Documentation

8.164.2.1 `uint64_t pack_loc_EventRegister_t::eventRegister`

8.164.2.2 `uint16_t pack_loc_EventRegister_t::Tlvresult`

8.165 pack_loc_SetExtPowerState_t Struct Reference**Data Fields**

- `uint32_t extPowerState`
- `uint16_t Tlvresult`

8.165.1 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.

8.165.2 Field Documentation

8.165.2.1 `uint32_t pack_loc_SetExtPowerState_t::extPowerState`

8.165.2.2 `uint16_t pack_loc_SetExtPowerState_t::Tlvresult`

8.166 pack_loc_SetOperationMode_t Struct Reference**Data Fields**

- `uint32_t mode`
- `uint16_t Tlvresult`

8.166.1 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.

8.166.2 Field Documentation

8.166.2.1 uint32_t pack_loc_SetOperationMode_t::mode

8.166.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

8.167 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

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

8.167.1 Detailed Description

This structure contains Set Operation Mode pack

Parameters

<i>xid</i>	<ul style="list-style-type: none"> Identifies the transaction. The transaction ID is returned in the Get Best Available Position indication.
<i>Tlvresult</i>	<ul style="list-style-type: none"> Pack result.

8.167.2 Field Documentation

8.167.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.167.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

8.168 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](#)

8.168.1 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

8.168.2 Field Documentation

8.168.2.1 `loc_LocApplicationInfo*` `pack_loc_Start_t::pApplicationInfo`

8.168.2.2 `uint32_t*` `pack_loc_Start_t::pConfigAltitudeAssumed`

8.168.2.3 `uint32_t*` `pack_loc_Start_t::pHorizontalAccuracyLvl`

8.168.2.4 `uint32_t*` `pack_loc_Start_t::pIntermediateReportState`

8.168.2.5 `uint32_t*` `pack_loc_Start_t::pMinIntervalTime`

8.168.2.6 `uint32_t*` `pack_loc_Start_t::pRecurrenceType`

8.168.2.7 `uint8_t` `pack_loc_Start_t::SessionId`

8.168.2.8 `uint16_t` `pack_loc_Start_t::Tlvresult`

8.169 pack_loc_Stop_t Struct Reference

Data Fields

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

8.169.1 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.

8.169.2 Field Documentation

8.169.2.1 `uint8_t` `pack_loc_Stop_t::SessionId`

8.169.2.2 `uint16_t` `pack_loc_Stop_t::Tlvresult`

8.170 pack_nas_SetACCOLC_t Struct Reference

Data Fields

- `int8_t` [spc](#) [6]
- `uint8_t` [accolc](#)

8.170.1 Detailed Description

Parameters

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

8.170.2 Field Documentation

8.170.2.1 `uint8_t pack_nas_SetACCOLC_t::accolc`

8.170.2.2 `int8_t pack_nas_SetACCOLC_t::spc[6]`

8.171 `pack_nas_SetNetworkPreference_t` Struct Reference

Data Fields

- `uint32_t TechnologyPref`
- `uint32_t Duration`
- `uint16_t Tlvresult`

8.171.1 Detailed Description

Parameters

<i>TechnologyPref[IN]</i>	<ul style="list-style-type: none"> • Bitmask representing the radio technology preference set. • No bits set indicates to the device to automatically determine the technology to use • Values: <ul style="list-style-type: none"> – Bit 0 - Technology is 3GPP2 – Bit 1 - Technology is 3GPP • Any combination of the following may be returned: <ul style="list-style-type: none"> – Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP – Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP – Bit 4 - HDR – Bit 5 - LTE – Bits 6 to 15 - Reserved
<i>Duration[IN]</i>	<ul style="list-style-type: none"> • Duration of active preference <ul style="list-style-type: none"> – 0 - Permanent – 1 - Power cycle – 2 - Until the end of the next call or a power cycle – 3 - Until the end of the next call, a specified time, or a power cycle – 4 to 6 - Until the end of the next call
<i>Tlvresult</i>	<ul style="list-style-type: none"> • pack result

8.171.2 Field Documentation

8.171.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

8.171.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

8.171.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

8.172 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

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

8.172.1 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.

8.172.2 Field Documentation

8.172.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.172.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.172.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.173 pack_nas_SLQSIInitiateNetworkRegistration_t Struct Reference

Data Fields

- uint32_t [regAction](#)
- [nas_MNRInfo](#) * [pMNRInfo](#)

- uint32_t * [pChangeDuration](#)
- uint8_t * [pMncPcsDigitStatus](#)

8.173.1 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.

8.173.2 Field Documentation

8.173.2.1 uint32_t* [pack_nas_SLQSIInitiateNetworkRegistration_t::pChangeDuration](#)

8.173.2.2 uint8_t* [pack_nas_SLQSIInitiateNetworkRegistration_t::pMncPcsDigitStatus](#)

8.173.2.3 [nas_MNRInfo](#)* [pack_nas_SLQSIInitiateNetworkRegistration_t::pMNRInfo](#)

8.173.2.4 uint32_t [pack_nas_SLQSIInitiateNetworkRegistration_t::regAction](#)

8.174 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference

Data Fields

- [nas_CDMARSSIThresh](#) * [pCDMARSSIThresh](#)
- uint16_t * [pCDMARSSIDelta](#)
- [nas_CDMAECIOThresh](#) * [pCDMAECIOThresh](#)
- uint16_t * [pCDMAECIODelta](#)
- [nas_HDRRSSIThresh](#) * [pHDRRSSIThresh](#)
- uint16_t * [pHDRRSSIDelta](#)
- [nas_HDRECIOThresh](#) * [pHDRECIOThresh](#)

- uint16_t * pHDRECIODelta
- nas_HDRSINRThreshold * pHDRSINRThresh
- uint16_t * pHDRSINRDelta
- nas_HDRIOTresh * pHDRIOTresh
- uint16_t * pHDRIODelta
- nas_GSMRSSIThresh * pGSMRSSIThresh
- uint16_t * pGSMRSSIDelta
- nas_WCDMARSSIThresh * pWCDMARSSIThresh
- uint16_t * pWCDMARSSIDelta
- nas_WCDMAECIOThresh * pWCDMAECIOThresh
- uint16_t * pWCDMAECIODelta
- nas_LTERSSIThresh * pLTERSSIThresh
- uint16_t * pLTERSSIDelta
- nas_LTESNRThreshold * pLTESNRThresh
- uint16_t * pLTESNRDelta
- nas_LTERSRQThresh * pLTERSRQThresh
- uint16_t * pLTERSRQDelta
- nas_LTERSRPThresh * pLTERSRPThresh
- uint16_t * pLTERSRPDelta
- nas_LTESigRptConfig * pLTESigRptConfig
- nas_TDSCDMARSCPTThresh * pTDSCDMARSCPTThresh
- uint16_t * pTDSCDMARSCPDelta
- nas_TDSCDMARSSIThresh * pTDSCDMARSSIThresh
- float * pTDSCDMARSSIDelta
- nas_TDSCDMAECIOThresh * pTDSCDMAECIOThresh
- float * pTDSCDMAECIODelta
- nas_TDSCDMASINRThresh * pTDSCDMASINRThresh
- float * pTDSCDMASINRDelta

8.174.1 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>pHDRECIOTresh</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>pHDRSINRTresh</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-Tresh</i>	<ul style="list-style-type: none"> GSM RSSI Threshold List See GSMRSSITresh 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-Tresh</i>	<ul style="list-style-type: none"> WCDMA RSSI Threshold List See WCDMARSSITresh 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-Tresh</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-Tresh</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.

8.174.2 Field Documentation

- 8.174.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta
- 8.174.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh
- 8.174.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta
- 8.174.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh
- 8.174.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta
- 8.174.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh
- 8.174.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta
- 8.174.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh
- 8.174.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta
- 8.174.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh
- 8.174.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIDelta
- 8.174.2.12 nas_HDRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIThresh
- 8.174.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRDelta
- 8.174.2.14 nas_HRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRThresh
- 8.174.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta
- 8.174.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh
- 8.174.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta
- 8.174.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh
- 8.174.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta
- 8.174.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh
- 8.174.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig
- 8.174.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta
- 8.174.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh
- 8.174.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta
- 8.174.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh
- 8.174.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta
- 8.174.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh
- 8.174.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

8.174.2.29 nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh

8.174.2.30 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

8.174.2.31 nas_TDSCDMASINRThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh

8.174.2.32 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIODelta

8.174.2.33 nas_WCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh

8.174.2.34 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta

8.174.2.35 nas_WCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh

8.175 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](#)

8.175.1 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>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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

8.175.2 Field Documentation

8.175.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.175.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.175.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.175.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.175.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.175.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.175.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.175.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.175.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.175.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.175.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.175.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.175.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.176 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](#)

8.176.1 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

8.176.2 Field Documentation

- 8.176.2.1 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsDI
- 8.176.2.2 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsUI
- 8.176.2.3 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmDI
- 8.176.2.4 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmUI
- 8.176.2.5 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmDI
- 8.176.2.6 uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmUI
- 8.176.2.7 uint8_t* pack_nas_SLQSNasSwiOTAMessageCallback_t::pRankIndicatorInd

8.177 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

Data Fields

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

8.177.1 Detailed Description

Parameters

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

8.177.2 Field Documentation

- 8.177.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable
- 8.177.2.2 [nas_SLQSSignalStrengthsIndReq](#)* [pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq](#)

8.178 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](#)

8.178.1 Detailed Description

Contain the system selection preferences.

Parameters

<i>pEmerMode</i>	<ul style="list-style-type: none">• Optional parameter specifying the emergency Mode• Values:<ul style="list-style-type: none">– 0 - OFF (normal)– 1 - ON (Emergency)
<i>pModePref</i>	<ul style="list-style-type: none">• Optional parameter• Bit Mask indicating the radio technology mode preference• Bit values:<ul style="list-style-type: none">– Bit 0 - cdma2000 1x– Bit 1 - cdma2000 HRPD(1xEV-DO)– Bit 2 - GSM– Bit 3 - UMTS– Bit 4 - LTE

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

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

<i>pNetSelPref</i>	<ul style="list-style-type: none"> - netSelectionPref • Optional parameter for specifying Network Selection Preference • Modem selects networks based on this parameter(if present). • see 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

8.178.2 Field Documentation

8.178.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.178.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

8.178.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

8.178.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

8.178.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

8.178.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

- 8.178.2.7 `uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref`
- 8.178.2.8 `uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat`
- 8.178.2.9 `uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref`
- 8.178.2.10 `struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref`
- 8.178.2.11 `uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref`
- 8.178.2.12 `unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT`
- 8.178.2.13 `uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref`
- 8.178.2.14 `uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref`
- 8.178.2.15 `uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction`
- 8.178.2.16 `uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdsdmaBandPref`

8.179 `pack_qmi_t` Struct Reference

Data Fields

- `uint16_t` [xid](#)
- `int` [timeout](#)
- `uint16_t` [msgid](#)
- `uint8_t` [svc](#)

8.179.1 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

8.179.2 Field Documentation

- 8.179.2.1 `uint16_t` `pack_qmi_t::msgid`
- 8.179.2.2 `uint8_t` `pack_qmi_t::svc`
- 8.179.2.3 `int` `pack_qmi_t::timeout`
- 8.179.2.4 `uint16_t` `pack_qmi_t::xid`

8.180 `pack_qos_SLQSQosSwiReadApnExtraParams_t` Struct Reference

Data Fields

- uint32_t [apnId](#)

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

8.180.2 Field Documentation

8.180.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.181 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.181.1 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
-------------------	--

8.181.2 Field Documentation

8.181.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.182 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t [enable](#)

8.182.1 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
--------------------	---

8.182.2 Field Documentation

8.182.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.183 pack_sms_SendSMS_t Struct Reference

Data Fields

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

8.183.1 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

8.183.2 Field Documentation

8.183.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.183.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.183.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.183.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.184 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum [eqmiCbkJSetStatus status](#)

8.184.1 Detailed Description

Parameters

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

8.184.2 Field Documentation

8.184.2.1 enum eqmiCbkJSetStatus pack_sms_SetNewSMSCallback_t::status

8.185 pack_sms_SLQSDDeleteSMS_t Struct Reference

Data Fields

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

8.185.1 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)

8.185.2 Field Documentation

8.185.2.1 uint32_t* pack_sms_SLQSDDeleteSMS_t::pMessageIndex

8.185.2.2 uint8_t* pack_sms_SLQSDDeleteSMS_t::pMessageMode

8.185.2.3 uint32_t* pack_sms_SLQSDDeleteSMS_t::pMessageTag

8.185.2.4 uint32_t pack_sms_SLQSDDeleteSMS_t::storageType

8.186 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

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

8.186.1 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)

8.186.2 Field Documentation

8.186.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.186.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

8.186.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

8.187 pack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

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

8.187.1 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)

8.187.2 Field Documentation

8.187.2.1 uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode

8.187.2.2 uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag

8.187.2.3 uint32_t pack_sms_SLQSGetSMSList_t::storageType

8.188 pack_sms_SLQSMModifySMSStatus_t Struct Reference

Data Fields

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

8.188.1 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)
---------------------	--

8.188.2 Field Documentation

8.188.2.1 uint32_t pack_sms_SLQSMModifySMSStatus_t::messageIndex

8.188.2.2 uint32_t pack_sms_SLQSMModifySMSStatus_t::messageTag

8.188.2.3 uint8_t* pack_sms_SLQSMModifySMSStatus_t::pMessageMode

8.188.2.4 uint32_t pack_sms_SLQSMModifySMSStatus_t::storageType

8.189 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](#)

8.189.1 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

8.189.2 Field Documentation

8.189.2.1 uint32_t pack_swilloc_SwiLocSetAutoStart_t::fix_rate

8.189.2.2 uint8_t pack_swilloc_SwiLocSetAutoStart_t::fix_type

8.189.2.3 uint8_t pack_swilloc_SwiLocSetAutoStart_t::function

8.189.2.4 uint32_t pack_swilloc_SwiLocSetAutoStart_t::max_dist

8.189.2.5 uint8_t pack_swilloc_SwiLocSetAutoStart_t::max_time

8.189.2.6 int pack_swilloc_SwiLocSetAutoStart_t::set_fix_rate

8.189.2.7 int pack_swilloc_SwiLocSetAutoStart_t::set_fix_type

8.189.2.8 int pack_swilloc_SwiLocSetAutoStart_t::set_function

8.189.2.9 int pack_swilloc_SwiLocSetAutoStart_t::set_max_dist

8.189.2.10 int pack_swilloc_SwiLocSetAutoStart_t::set_max_time

8.190 pack_swiloma_SLQSOMADMCancelSession_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.190.1 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
------------------------	---

8.190.2 Field Documentation

8.190.2.1 uint32_t pack_swioama_SLQSOMADMCancelSession_t::sessionType

8.191 pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t [SessionType](#)

8.191.1 Detailed Description

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

Parameters

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

8.191.2 Field Documentation

8.191.2.1 uint32_t pack_swioama_SLQSOMADMGetSessionInfo_t::SessionType

8.192 pack_swioama_SLQSOMADMSendSelection_t Struct Reference

Data Fields

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

8.192.1 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.

8.192.2 Field Documentation

8.192.2.1 uint32_t* pack_swima_SLQSOMADMSendSelection_t::pDeferTime

8.192.2.2 uint32_t* pack_swima_SLQSOMADMSendSelection_t::pRejectReason

8.192.2.3 uint32_t pack_swima_SLQSOMADMSendSelection_t::selection

8.193 pack_swima_SLQSOMADMSetSettings_t Struct Reference

Data Fields

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

8.193.1 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[IN]</i>	<ul style="list-style-type: none"> • Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled Accept – 0x02 - Enabled Reject
<i>pFwAutoCheck[IN]</i>	<ul style="list-style-type: none"> • Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> – 0x00 - Disabled – 0x01 - Enabled

8.193.2 Field Documentation

8.193.2.1 `uint8_t pack_swioama_SLQSOMADMSetSettings_t::FOTAdownload`

8.193.2.2 `uint8_t pack_swioama_SLQSOMADMSetSettings_t::FOTAUpdate`

8.193.2.3 `uint8_t* pack_swioama_SLQSOMADMSetSettings_t::pAutosdm`

8.193.2.4 `uint8_t* pack_swioama_SLQSOMADMSetSettings_t::pFwAutoCheck`

8.194 `pack_swioama_SLQSOMADMStartSession_t` Struct Reference

Data Fields

- `uint32_t` [sessionType](#)

8.194.1 Detailed Description

Structure that contains the session type for OMA start session command

Parameters

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

8.194.2 Field Documentation

8.194.2.1 `uint32_t pack_swioama_SLQSOMADMStartSession_t::sessionType`

8.195 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](#)

8.195.1 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.

8.195.2 Field Documentation

8.195.2.1 [uim_changeUIMPIN](#) [pack_uim_ChangePin_t::changePIN](#)

8.195.2.2 [uim_encryptedPIN1](#) [pack_uim_ChangePin_t::EncryptedPIN1](#)

8.195.2.3 [uint32_t*](#) [pack_uim_ChangePin_t::pIndicationToken](#)

8.195.2.4 [uint8_t*](#) [pack_uim_ChangePin_t::pKeyReferenceID](#)

8.195.2.5 [uim_sessionInformation](#) [pack_uim_ChangePin_t::sessionInfo](#)

8.195.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

8.196 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](#)

8.196.1 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.

8.196.2 Field Documentation

8.196.2.1 uim_fileInfo pack_uim_ReadTransparent_t::fileIndex

8.196.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

8.196.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

8.196.2.4 uim_readTransparentInfo pack_uim_ReadTransparent_t::readTransparent

8.196.2.5 uim_sessionInformation pack_uim_ReadTransparent_t::sessionInfo

8.196.2.6 uint16_t pack_uim_ReadTransparent_t::Tlvresult

8.197 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

8.197.1 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.

8.197.2 Field Documentation

8.197.2.1 uim_encryptedPIN1 pack_uim_SetPinProtection_t::EncryptedPIN1

8.197.2.2 uint32_t* pack_uim_SetPinProtection_t::pIndicationToken

8.197.2.3 `uim_setPINProtection` `pack_uim_SetPinProtection_t::pinProtection`

8.197.2.4 `uint8_t*` `pack_uim_SetPinProtection_t::pKeyReferenceID`

8.197.2.5 `uim_sessionInformation` `pack_uim_SetPinProtection_t::sessionInfo`

8.197.2.6 `uint16_t` `pack_uim_SetPinProtection_t::Tlvresult`

8.198 `pack_uim_SLQSUIEventRegister_t` Struct Reference

Data Fields

- `uint32_t` [eventMask](#)

8.198.1 Detailed Description

Parameters

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

8.198.2 Field Documentation

8.198.2.1 `uint32_t` `pack_uim_SLQSUIEventRegister_t::eventMask`

8.199 `pack_uim_SLQSUISSwitchSlot_t` Struct Reference

Data Fields

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

8.199.1 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
-----------------------	--

8.199.2 Field Documentation

8.199.2.1 `uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot`

8.199.2.2 `uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot`

8.200 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`

8.200.1 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
-----------------------------------	---

8.200.2 Field Documentation

8.200.2.1 `uim_encryptedPIN1` `pack_uim_UnblockPin_t::EncryptedPIN1`

8.200.2.2 `uint32_t*` `pack_uim_UnblockPin_t::pIndicationToken`

8.200.2.3 `uim_unblockUIMPIN` `pack_uim_UnblockPin_t::pinProtection`

8.200.2.4 `uint8_t*` `pack_uim_UnblockPin_t::pKeyReferenceID`

8.200.2.5 `uim_sessionInformation` `pack_uim_UnblockPin_t::sessionInfo`

8.200.2.6 `uint16_t` `pack_uim_UnblockPin_t::Tlvresult`

8.201 `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`

8.201.1 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 <code>UIMSessionInformation</code> for more information.
<i>verifyPIN</i>	<ul style="list-style-type: none"> • See <code>verifyUIMPIN</code> for more information.
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See <code>encryptedPIN1</code> 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.

8.201.2 Field Documentation

8.201.2.1 uim_encryptedPIN1* pack_uim_VerifyPin_t::pEncryptedPIN1

8.201.2.2 uint32_t* pack_uim_VerifyPin_t::pIndicationToken

8.201.2.3 uint8_t* pack_uim_VerifyPin_t::pKeyReferenceID

8.201.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

8.201.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

8.201.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

8.202 pack_wds_GetDefaultProfile_t Struct Reference**Data Fields**

- uint32_t [profiletype](#)

8.202.1 Detailed Description**Parameters**

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

8.202.2 Field Documentation

8.202.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.203 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

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

8.203.1 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

8.203.2 Field Documentation

8.203.2.1 [uint8_t pack_wds_GetDefaultProfileNum_t::family](#)

8.203.2.2 [uint8_t pack_wds_GetDefaultProfileNum_t::type](#)

8.204 [pack_wds_GetDormancyState_t](#) Struct Reference

8.205 [pack_wds_GetLastMobileIPError_t](#) Struct Reference

8.206 [pack_wds_GetMobileIP_t](#) Struct Reference

8.207 [pack_wds_GetMobileIPProfile_t](#) Struct Reference

Data Fields

- [uint8_t index](#)

8.207.1 Detailed Description

Parameters

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

8.207.2 Field Documentation

8.207.2.1 [uint8_t pack_wds_GetMobileIPProfile_t::index](#)

8.208 [pack_wds_GetPacketStatus_t](#) Struct Reference

Data Fields

- uint32_t [statmask](#)

8.208.1 Detailed Description

Parameters

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

8.208.2 Field Documentation

8.208.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.209 pack_wds_GetSessionDuration_t Struct Reference

8.210 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- [rmTrasnferStaticsReq](#) RmTrasnferStaticsReq

8.210.1 Detailed Description

Parameters

rmTrasnfer-StaticsReq	RM Transfer Statistics Indicator
---------------------------------------	----------------------------------

8.210.2 Field Documentation

8.210.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

8.211 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](#)

8.211.1 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

8.211.2 Field Documentation

8.211.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

8.211.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

8.211.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

8.211.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

8.211.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

8.211.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

8.211.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

8.211.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

8.211.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

8.211.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

8.212 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

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

8.212.1 Field Documentation

8.212.1.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

8.212.1.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

8.212.1.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

8.213 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](#)

8.213.1 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

8.213.2 Field Documentation

8.213.2.1 uint8_t pack_wds_SetMobileIPProfile_t::index

8.213.2.2 uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI

8.213.2.3 uint32_t* pack_wds_SetMobileIPProfile_t::pAddress

8.213.2.4 uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled

8.213.2.5 uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI

8.213.2.6 int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA

8.213.2.7 int8_t* pack_wds_SetMobileIPProfile_t::pMNHA

8.213.2.8 int8_t* pack_wds_SetMobileIPProfile_t::pNAI

8.213.2.9 uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA

8.213.2.10 uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling

8.213.2.11 uint32_t* pack_wds_SetMobilePPProfile_t::pSecondaryHA

8.213.2.12 int8_t pack_wds_SetMobilePPProfile_t::spc[10]

8.214 pack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

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

8.214.1 Detailed Description

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
<i>ProfileType</i>	<ul style="list-style-type: none"> • Identifies the technology type of the profile <ul style="list-style-type: none"> – 0x00 - 3GPP – 0x01 - 3GPP2 – 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

8.214.2 Field Documentation

8.214.2.1 [wds_profileInfo](#)* [pack_wds_SLQSCreateProfile_t::pCurProfile](#)

8.214.2.2 uint8_t* [pack_wds_SLQSCreateProfile_t::pProfileId](#)

8.214.2.3 uint8_t* [pack_wds_SLQSCreateProfile_t::pProfileType](#)

8.215 pack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

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

8.215.1 Detailed Description

Parameters

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

8.215.2 Field Documentation

8.215.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.215.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.216 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.217 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.218 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](#)

8.218.1 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

8.218.2 Field Documentation

8.218.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.218.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.218.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.218.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.218.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.218.2.6 `transferStatInd*` `pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd`

8.219 `pack_wds_SLQSGetProfileSettings_t` Struct Reference

Data Fields

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

8.219.1 Detailed Description

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
<i>ProfileType</i>	<ul style="list-style-type: none"> • Identifies the technology type of the profile <ul style="list-style-type: none"> – 0x00 - 3GPP – 0x01 - 3GPP2

Note

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

Returns

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_xxx` error value otherwise

See Also

See [qmerrno.h](#) for `eQCWWAN_xxx` error values

8.219.2 Field Documentation

8.219.2.1 `uint8_t` `pack_wds_SLQSGetProfileSettings_t::ProfileId`

8.219.2.2 `uint8_t` `pack_wds_SLQSGetProfileSettings_t::ProfileType`

8.220 `pack_wds_SLQSGetRuntimeSettings_t` Struct Reference

Data Fields

- `uint32_t *` [pReqSettings](#)

8.220.1 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
---------------------	--

8.220.2 Field Documentation

8.220.2.1 `uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings`

8.221 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- `uint8_t * pProfileId`
- `uint8_t * pProfileType`
- `wds_profileInfo curProfile`

8.221.1 Detailed Description

Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> • 1 to 16 for 3GPP profile (EM/MC73xx or earlier) • 1 to 24 for 3GPP profile (EM/MC74xx onwards) • 101 to 106 for 3GPP2 profile
------------------	---

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

8.221.2 Field Documentation

8.221.2.1 **wds_profileInfo** pack_wds_SLQSMModifyProfile_t::curProfile

8.221.2.2 **uint8_t*** pack_wds_SLQSMModifyProfile_t::pProfileId

8.221.2.3 **uint8_t*** pack_wds_SLQSMModifyProfile_t::pProfileType

8.222 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](#)

8.222.1 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

8.222.2 Field Documentation

8.222.2.1 uint8_t pack_wds_SLQSSet3GPPConfigItem_t::_3gppRelease

8.222.2.2 uint8_t pack_wds_SLQSSet3GPPConfigItem_t::defaultPDNEnabled

8.222.2.3 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileList[24]

8.222.2.4 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.222.2.5 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::profileList[5]

8.223 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint8_t [IPFamilyPreference](#)

8.223.1 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
----------------------------	---

8.223.2 Field Documentation

8.223.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

8.224 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](#)

8.224.1 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

8.224.2 Field Documentation

8.224.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

8.224.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

8.224.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

8.224.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

8.224.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

8.224.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

8.224.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

8.225 pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

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

8.225.1 Detailed Description

Parameters

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

8.225.2 Field Documentation

8.225.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSetDHCPv4ClientConfig_t::pProfileId

8.226 pack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

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

8.226.1 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

8.226.2 Field Documentation

8.226.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

8.226.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

8.226.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

8.226.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

8.226.2.5 `uint8_t*` `pack_wds_SLQSStartDataSession_t::pTech`

8.226.2.6 `char*` `pack_wds_SLQSStartDataSession_t::pUser`

8.227 `pack_wds_SLQSStopDataSession_t` Struct Reference

Data Fields

- `uint32_t` * [psid](#)

8.227.1 Detailed Description

Parameters

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

8.227.2 Field Documentation

8.227.2.1 `uint32_t*` `pack_wds_SLQSStopDataSession_t::psid`

8.228 `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t` Struct Reference

Data Fields

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

8.228.1 Detailed Description

Parameters

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

8.228.2 Field Documentation

8.228.2.1 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.228.2.2 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType`

8.229 `PackCreateProfileOut` Struct Reference

Data Fields

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

8.229.1 Field Documentation

8.229.1.1 `uint16_t` `PackCreateProfileOut::ExtErrorCode`

8.229.1.2 uint8_t PackCreateProfileOut::ProfileIndex

8.229.1.3 uint8_t PackCreateProfileOut::ProfileType

8.230 packgetDyingGaspCfg Struct Reference

Data Fields

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

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

8.230.2 Field Documentation

8.230.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.230.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.231 packgetDyingGaspStatistics Struct Reference

Data Fields

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

8.231.1 Detailed Description

Parameters

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

8.231.2 Field Documentation

8.231.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.231.2.2 uint32_t* packetDyingGaspStatistics::pTimeStamp

8.232 qmiSmsMessageList Struct Reference

Data Fields

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

8.232.1 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

8.232.2 Field Documentation

8.232.2.1 uint32_t qmiSmsMessageList::messageIndex

8.232.2.2 uint32_t qmiSmsMessageList::messageTag

8.233 qmiWSDDataBearerTechnology Struct Reference

Data Fields

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

8.233.1 Detailed Description

Parameters

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

8.233.2 Field Documentation

8.233.2.1 uint8_t qmiWSDDataBearerTechnology::currentNetwork

8.233.2.2 uint32_t qmiWSDDataBearerTechnology::ratMask

8.233.2.3 uint32_t qmiWSDDataBearerTechnology::soMask

8.234 RFBandInfoElements Struct Reference

Data Fields

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

8.234.1 Detailed Description

Parameters

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

8.234.2 Field Documentation

8.234.2.1 uint16_t RFBandInfoElements::activeBandClass

8.234.2.2 uint16_t RFBandInfoElements::activeChannel

8.234.2.3 uint8_t RFBandInfoElements::radioInterface

8.235 rmTrasnferStaticsReq Struct Reference

Data Fields

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

8.235.1 Detailed Description

Parameters

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

8.235.2 Field Documentation

8.235.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

8.235.2.2 uint32_t rmTrasnferStaticsReq::ulMask

8.236 slot_t Struct Reference

Data Fields

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

8.236.1 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.

8.236.2 Field Documentation

8.236.2.1 `uint8_t slot_t::bICCID[255]`

8.236.2.2 `uint8_t slot_t::bICCIDLength`

8.236.2.3 `uint8_t slot_t::bLogicalSlot`

8.236.2.4 `uint32_t slot_t::uPhyCardStatus`

8.236.2.5 `uint32_t slot_t::uPhySlotStatus`

8.237 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]

8.237.1 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_APPLICATIONS]	<ul style="list-style-type: none"> See appStats for more information.
---	--

8.237.2 Field Documentation

8.237.2.1 `appStats slotInf::AppStatus`[10]

8.237.2.2 `uint8_t slotInf::cardState`

8.237.2.3 `uint8_t slotInf::errorState`

8.237.2.4 `uint8_t slotInf::numApp`

8.237.2.5 `uint8_t slotInf::upinRetries`

8.237.2.6 `uint8_t slotInf::upinState`

8.237.2.7 `uint8_t slotInf::upukRetries`

8.238 `slots_t` Struct Reference

Data Fields

- [slot_t uimSlotStatus](#) [255]

8.238.1 Field Documentation

8.238.1.1 `slot_t slots_t::uimSlotStatus`[255]

8.239 `sMSCAddress` Struct Reference

Data Fields

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

8.239.1 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

8.239.2 Field Documentation

8.239.2.1 uint8_t sMSCAddress::data[256]

8.239.2.2 uint8_t sMSCAddress::length

8.240 sMSCAddressTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [sMSCAddressInfo](#) [SMSCInfo](#)

8.240.1 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

8.240.2 Field Documentation

8.240.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

8.240.2.2 uint8_t sMSCAddressTlv::TlvPresent

8.241 sMSEtwsMessage Struct Reference

Data Fields

- uint8_t [notificationType](#)
- uint16_t [length](#)
- uint8_t [data](#) [1254]

8.241.1 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

8.241.2 Field Documentation

8.241.2.1 uint8_t sMSEtwsMessage::data[1254]

8.241.2.2 uint16_t sMSEtwsMessage::length

8.241.2.3 uint8_t sMSEtwsMessage::notificationType

8.242 sMSEtwsMessageTlv Struct Reference

Data Fields

- uint8_t [TlvPresent](#)
- [sMSEtwsMessageInfo](#) [EtwsMessageInfo](#)

8.242.1 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 sMSEtwsMessageInfo for more information

8.242.2 Field Documentation

8.242.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

8.242.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

8.243 sMSEtwsPlmn Struct Reference

Data Fields

- uint16_t [mobileCountryCode](#)
- uint16_t [mobileNetworkCode](#)

8.243.1 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

8.243.2 Field Documentation

8.243.2.1 uint16_t sMSEtwsPlmn::mobileCountryCode

8.243.2.2 uint16_t sMSEtwsPlmn::mobileNetworkCode

8.244 sMSMessageMode Struct Reference

Data Fields

- uint8_t [messageMode](#)

8.244.1 Detailed Description

Parameters

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

8.244.2 Field Documentation

8.244.2.1 uint8_t sMSMessageMode::messageMode

8.245 sMSMTMessage Struct Reference

Data Fields

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

8.245.1 Detailed Description

Parameters

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

8.245.2 Field Documentation

8.245.2.1 uint32_t sMSMTMessage::messageIndex

8.245.2.2 uint32_t sMSMTMessage::storageType

8.246 sMSOnIMS Struct Reference

Data Fields

- uint8_t [smsOnIMS](#)

8.246.1 Detailed Description

Parameters

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

8.246.2 Field Documentation

8.246.2.1 `uint8_t sMSONIMS::smsOnIMS`

8.247 sMSONIMSTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSONIMSInfo IMSInfo`

8.247.1 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 <code>sMSONIMSInfo</code> for more information

8.247.2 Field Documentation

8.247.2.1 `sMSONIMSInfo sMSONIMSTlv::IMSInfo`

8.247.2.2 `uint8_t sMSONIMSTlv::TlvPresent`

8.248 sMSTransferRouteMTMessage Struct Reference

Data Fields

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

8.248.1 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

8.248.2 Field Documentation

8.248.2.1 uint8_t sMSTransferRouteMTMessage::ackIndicator

8.248.2.2 uint8_t sMSTransferRouteMTMessage::data[256]

8.248.2.3 uint8_t sMSTransferRouteMTMessage::format

8.248.2.4 uint16_t sMSTransferRouteMTMessage::length

8.248.2.5 uint32_t sMSTransferRouteMTMessage::transactionID

8.249 tdscdmaSigInfoExt Struct Reference

Data Fields

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

8.249.1 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.

8.249.2 Field Documentation

8.249.2.1 float tdscdmaSigInfoExt::ecio

8.249.2.2 float tdscdmaSigInfoExt::rscp

8.249.2.3 float tdscdmaSigInfoExt::rssi

8.249.2.4 float tdscdmaSigInfoExt::sinr

8.250 transferRouteMessageTlv Struct Reference

Data Fields

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

8.250.1 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

8.250.2 Field Documentation

8.250.2.1 [uint8_t transferRouteMessageTlv::TlvPresent](#)

8.250.2.2 [sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo](#)

8.251 transferStatInd Struct Reference

Data Fields

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

8.251.1 Detailed Description

Parameters

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

8.251.2 Field Documentation

8.251.2.1 [uint32_t transferStatInd::StatsMask](#)

8.251.2.2 [uint8_t transferStatInd::StatsPeriod](#)

8.252 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](#)

8.252.1 Detailed Description

This structure contains Application Status Information loaded on the card.

Parameters

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

<i>persoFeature</i>	<ul style="list-style-type: none"> • Indicates the personalization feature. • This applies only when a personalization code is required to deactivate or unblock personalization. <ul style="list-style-type: none"> – 0 - GW network personalization – 1 - GW network subset personalization – 2 - GW service provider personalization – 3 - GW corporate personalization – 4 - GW UIM personalization – 5 - 1X network type 1 personalization – 6 - 1X network type 2 personalization – 7 - 1X HRPD personalization – 8 - 1X service provider personalization – 9 - 1X corporate personalization – 10 - 1X RUIM personalization – 11 - Unknown
<i>persoRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to disable the personalization.
<i>persoUnblock-Retries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock the personalization.
<i>aidLength</i>	<ul style="list-style-type: none"> • Number of sets of the following elements. i.e. aidVal • If zero(0) then no aidVal information exists.
<i>aidVal[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> • Application identifier value.
<i>univPin</i>	<ul style="list-style-type: none"> • Indicates whether UPIN replaces PIN1. <ul style="list-style-type: none"> – 0 - PIN1 is used – 1 - UPIN replaces PIN1
<i>pin1State</i>	<ul style="list-style-type: none"> • Indicates the state of PIN1. <ul style="list-style-type: none"> – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked
<i>pin1Retries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to verify PIN1.
<i>puk1Retries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock PIN1.

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

8.252.2 Field Documentation

8.252.2.1 `uint8_t uim_appStatus::aidLength`

8.252.2.2 `uint8_t uim_appStatus::aidVal[255]`

8.252.2.3 `uint8_t uim_appStatus::appState`

8.252.2.4 `uint8_t uim_appStatus::appType`

8.252.2.5 `uint8_t uim_appStatus::persoFeature`

8.252.2.6 `uint8_t uim_appStatus::persoRetries`

8.252.2.7 `uint8_t uim_appStatus::persoState`

8.252.2.8 `uint8_t uim_appStatus::persoUnblockRetries`

8.252.2.9 `uint8_t uim_appStatus::pin1Retries`

8.252.2.10 `uint8_t uim_appStatus::pin1State`

8.252.2.11 `uint8_t uim_appStatus::pin2Retries`

8.252.2.12 `uint8_t uim_appStatus::pin2State`

8.252.2.13 `uint8_t uim_appStatus::puk1Retries`

8.252.2.14 `uint8_t uim_appStatus::puk2Retries`

8.252.2.15 `uint8_t uim_appStatus::univPin`

8.253 uim_cardResult Struct Reference

Data Fields

- `uint8_t` [sw1](#)

- `uint8_t sw2`

8.253.1 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.

8.253.2 Field Documentation

8.253.2.1 `uint8_t uim_cardResult::sw1`

8.253.2.2 `uint8_t uim_cardResult::sw2`

8.254 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]

8.254.1 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.

8.254.2 Field Documentation

8.254.2.1 `uint16_t uim_cardStatus::index1xPri`

8.254.2.2 `uint16_t uim_cardStatus::index1xSec`

8.254.2.3 `uint16_t uim_cardStatus::indexGwPri`

8.254.2.4 `uint16_t uim_cardStatus::indexGwSec`

8.254.2.5 `uint8_t uim_cardStatus::numSlot`

8.254.2.6 `uim_slotInfo uim_cardStatus::SlotInfo[5]`

8.255 uim_changeUIMPIN Struct Reference

Data Fields

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

8.255.1 Detailed Description

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

Parameters

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

8.255.2 Field Documentation

8.255.2.1 `uint8_t uim_changeUIMPIN::oldPINLen`8.255.2.2 `uint8_t uim_changeUIMPIN::oldPINVal[255]`8.255.2.3 `uint8_t uim_changeUIMPIN::pinID`8.255.2.4 `uint8_t uim_changeUIMPIN::pinLen`8.255.2.5 `uint8_t uim_changeUIMPIN::pinVal[255]`8.256 `uim_encryptedPIN1` Struct Reference

Data Fields

- `uint8_t pin1Len`
- `uint8_t pin1Val [255]`

8.256.1 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.

8.256.2 Field Documentation

8.256.2.1 `uint8_t uim_encryptedPIN1::pin1Len`

8.256.2.2 `uint8_t uim_encryptedPIN1::pin1Val[255]`

8.257 uim_fileInfo Struct Reference**Data Fields**

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

8.257.1 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).

8.257.2 Field Documentation

8.257.2.1 `uint16_t uim_fileInfo::fileID`

8.257.2.2 `uint16_t uim_fileInfo::path[255]`

8.257.2.3 `uint8_t uim_fileInfo::pathLen`

8.258 uim_hotSwapStatus Struct Reference

Data Fields

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

8.258.1 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

8.258.2 Field Documentation

8.258.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.258.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.259 uim_readResult Struct Reference

Data Fields

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

8.259.1 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.

8.259.2 Field Documentation

8.259.2.1 uint8_t uim_readResult::content[255]

8.259.2.2 uint16_t uim_readResult::contentLen

8.260 uim_readTransparentInfo Struct Reference

Data Fields

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

8.260.1 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.

8.260.2 Field Documentation

8.260.2.1 uint16_t uim_readTransparentInfo::length

8.260.2.2 uint16_t uim_readTransparentInfo::offset

8.261 uim_remainingRetries Struct Reference

Data Fields

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

8.261.1 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.

8.261.2 Field Documentation

8.261.2.1 uint8_t uim_remainingRetries::unlockLeft

8.261.2.2 uint8_t uim_remainingRetries::verifyLeft

8.262 uim_sessionInformation Struct Reference**Data Fields**

- uint8_t [sessionType](#)
- uint8_t [aidLength](#)
- uint8_t [aid](#) [255]

8.262.1 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.

8.262.2 Field Documentation

8.262.2.1 uint8_t uim_sessionInformation::aid[255]

8.262.2.2 uint8_t uim_sessionInformation::aidLength

8.262.2.3 uint8_t uim_sessionInformation::sessionType

8.263 uim_setPINProtection Struct Reference

Data Fields

- uint8_t [pinID](#)
- uint8_t [pinOperation](#)
- uint8_t [pinLength](#)
- uint8_t [pinValue](#) [255]

8.263.1 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.

8.263.2 Field Documentation

8.263.2.1 uint8_t uim_setPINProtection::pinID

8.263.2.2 uint8_t uim_setPINProtection::pinLength

8.263.2.3 uint8_t uim_setPINProtection::pinOperation

8.263.2.4 uint8_t uim_setPINProtection::pinValue[255]

8.264 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]

8.264.1 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.

8.264.2 Field Documentation

8.264.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.264.2.2 `uint8_t uim_slotInfo::cardState`

8.264.2.3 `uint8_t uim_slotInfo::errorState`

8.264.2.4 `uint8_t uim_slotInfo::numApp`

8.264.2.5 `uint8_t uim_slotInfo::upinRetries`

8.264.2.6 `uint8_t uim_slotInfo::upinState`

8.264.2.7 `uint8_t uim_slotInfo::upukRetries`

8.265 uim_UIMSessionInformation Struct Reference

Data Fields

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

8.265.1 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.

8.265.2 Field Documentation

8.265.2.1 `uint8_t uim_UIMSessionInformation::aid[255]`

8.265.2.2 `uint8_t uim_UIMSessionInformation::aidLength`

8.265.2.3 `uint8_t uim_UIMSessionInformation::sessionType`

8.266 uim_unblockUIMPIN Struct Reference

Data Fields

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

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

8.266.2 Field Documentation

8.266.2.1 uint8_t uim_unblockUIMPIN::newPINLen

8.266.2.2 uint8_t uim_unblockUIMPIN::newPINVal[255]

8.266.2.3 uint8_t uim_unblockUIMPIN::pinID

8.266.2.4 uint8_t uim_unblockUIMPIN::pukLen

8.266.2.5 uint8_t uim_unblockUIMPIN::pukVal[255]

8.267 uim_verifyUIMPIN Struct Reference

Data Fields

- uint8_t [pinID](#)
- uint8_t [pinLen](#)
- uint8_t [pinVal](#) [255]

8.267.1 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.

8.267.2 Field Documentation

8.267.2.1 uint8_t uim_verifyUIMPIN::pinID

8.267.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.267.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.268 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- uint8_t [state](#)

8.268.1 Detailed Description

Parameters

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

8.268.2 Field Documentation

8.268.2.1 uint8_t unpack_dms_GetActivationState_t::state

8.269 unpack_dms_GetBandCapability_t Struct Reference

Data Fields

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

8.269.1 Field Documentation

8.269.1.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

8.269.1.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.270 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

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

8.270.1 Field Documentation

8.270.1.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.270.1.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.271 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](#)

8.271.1 Field Documentation

8.271.1.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.271.1.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.271.1.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.271.1.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.271.1.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.271.1.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.271.1.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

8.271.1.8 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

8.271.1.9 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

8.271.1.10 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

8.272 unpack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

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

8.272.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

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

8.272.2 Field Documentation

8.272.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

8.272.2.2 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

8.273 unpack_dms_GetDeviceCap_t Struct Reference

Data Fields

- uint32_t [MaxTXChannelRate](#)
- uint32_t [MaxRXChannelRate](#)
- uint32_t [DataServiceCapability](#)
- uint32_t [SimCapability](#)
- uint32_t [RadiolfacesSize](#)
- uint8_t [Radiolfaces](#) [64]
- uint16_t [Tlvresult](#)

8.273.1 Field Documentation

8.273.1.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

8.273.1.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

8.273.1.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

8.273.1.4 uint8_t unpack_dms_GetDeviceCap_t::Radiolfaces[64]

8.273.1.5 uint32_t unpack_dms_GetDeviceCap_t::RadiolfacesSize

8.273.1.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.273.1.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.274 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]

8.274.1 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>radiofacesSize</i>	radio interface length
<i>Radiofaces</i>	radio interfaces

8.274.2 Field Documentation

8.274.2.1 `uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability`

8.274.2.2 `uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate`

8.274.2.3 `uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate`

8.274.2.4 `uint8_t unpack_dms_GetDeviceCapabilities_t::Radiofaces[255]`

8.274.2.5 `uint32_t unpack_dms_GetDeviceCapabilities_t::radiofacesSize`

8.274.2.6 `uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability`

8.275 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

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

8.275.1 Field Documentation

8.275.1.1 `char unpack_dms_GetDeviceHardwareRev_t::String[255]`

8.275.1.2 `uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize`

8.275.1.3 `uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult`

8.276 unpack_dms_GetDeviceMfr_t Struct Reference

Data Fields

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

8.276.1 Field Documentation

8.276.1.1 `char unpack_dms_GetDeviceMfr_t::String[255]`

8.276.1.2 `uint8_t unpack_dms_GetDeviceMfr_t::stringSize`

8.276.1.3 `uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult`

8.277 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](#)

8.277.1 Field Documentation

8.277.1.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

8.277.1.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

8.277.1.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

8.277.1.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

8.277.1.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

8.277.1.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

8.277.1.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

8.277.1.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

8.277.1.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

8.278 unpack_dms_GetFirmwareInfo_t Struct Reference

Data Fields

- char [modelid_str](#) [20]
- char [bootversion_str](#) [85]
- char [appversion_str](#) [85]
- char [sku_str](#) [15]
- char [packageid_str](#) [85]
- char [carrier_str](#) [20]
- char [priversion_str](#) [16]
- char [cur_carr_name](#) [17]
- char [cur_carr_rev](#) [13]
- uint16_t [Tlvresult](#)

8.278.1 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.

8.278.2 Field Documentation

8.278.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

8.278.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

8.278.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

8.278.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

8.278.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

8.278.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

8.278.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

8.278.2.8 char unpack_dms_GetFirmwareInfo_t::priversion_str[16]

8.278.2.9 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

8.278.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.279 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

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

8.279.1 Field Documentation

8.279.1.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.279.1.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.279.1.3 `char unpack_dms_GetFirmwareRevision_t::PRIString[255]`

8.279.1.4 `uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult`

8.280 `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`

8.280.1 Detailed Description

Parameters

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

8.280.2 Field Documentation

8.280.2.1 `uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize`

8.280.2.2 `char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]`

8.280.2.3 `uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize`

8.280.2.4 `char unpack_dms_GetFirmwareRevisions_t::BootString[255]`

8.280.2.5 `uint8_t unpack_dms_GetFirmwareRevisions_t::priSize`

8.280.2.6 `char unpack_dms_GetFirmwareRevisions_t::PRIString[255]`

8.280.2.7 `uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult`

8.281 `unpack_dms_GetFSN_t` Struct Reference

Data Fields

- `char String [255]`
- `uint16_t Tlvresult`

8.281.1 Field Documentation

8.281.1.1 `char unpack_dms_GetFSN_t::String[255]`

8.281.1.2 `uint16_t unpack_dms_GetFSN_t::Tlvresult`

8.282 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char [hwVer](#) [255]

8.282.1 Detailed Description

Parameters

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

8.282.2 Field Documentation

8.282.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.283 unpack_dms_GetIMSI_t Struct Reference

Data Fields

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

8.283.1 Field Documentation

8.283.1.1 char unpack_dms_GetIMSI_t::imsi[255]

8.283.1.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.284 unpack_dms_GetModelID_t Struct Reference

Data Fields

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

8.284.1 Detailed Description

Parameters

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

8.284.2 Field Documentation

8.284.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.284.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.285 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

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

8.285.1 Detailed Description

Parameters

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

Note

The source of the timestamp provided specifies how the timestamp was determined. The first network time that is available will be returned. If no network time is available, the timestamp is taken from the 32 kHz slow-clock of the device.

8.285.2 Field Documentation

8.285.2.1 uint16_t `unpack_dms_GetNetworkTime_t::source`

8.285.2.2 uint64_t `unpack_dms_GetNetworkTime_t::timestamp`

8.285.2.3 uint16_t `unpack_dms_GetNetworkTime_t::Tlvresult`

8.286 `unpack_dms_GetPower_t` Struct Reference

Data Fields

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

8.286.1 Detailed Description

Parameters

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

8.286.2 Field Documentation

8.286.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

8.286.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

8.286.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

8.286.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

8.287 unpack_dms_GetPRLVersion_t Struct Reference

Data Fields

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

8.287.1 Field Documentation

8.287.1.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.287.1.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.287.1.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.288 unpack_dms_GetSerialNumbers_t Struct Reference

Data Fields

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

8.288.1 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

8.288.2 Field Documentation

8.288.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

8.288.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

8.288.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

8.288.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

8.289 unpack_dms_GetUSBComp_t Struct Reference

Data Fields

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

8.289.1 Field Documentation

8.289.1.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.289.1.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

8.289.1.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.289.1.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

8.290 unpack_dms_GetVoiceNumber_t Struct Reference

Data Fields

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

8.290.1 Field Documentation

8.290.1.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.290.1.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.290.1.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.290.1.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.290.1.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.291 unpack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t [notused](#)

8.291.1 Detailed Description

Modem response. Not used

8.291.2 Field Documentation

8.291.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

8.292 unpack_dms_SetCustFeature_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.292.1 Field Documentation

8.292.1.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

8.293 unpack_dms_SetCustFeaturesV2_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.293.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

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

8.293.2 Field Documentation

8.293.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

8.294 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

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

8.294.1 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

8.294.2 Field Documentation

8.294.2.1 `dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv`

8.294.2.2 `dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv`

8.294.2.3 `uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult`

8.295 `unpack_dms_SetEventReport_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.295.1 Field Documentation

8.295.1.1 `uint16_t unpack_dms_SetEventReport_t::Tlvresult`

8.296 `unpack_dms_SetFirmwarePreference_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.296.1 Field Documentation

8.296.1.1 `uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult`

8.297 `unpack_dms_SetPower_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.297.1 Field Documentation

8.297.1.1 `uint16_t unpack_dms_SetPower_t::Tlvresult`

8.298 `unpack_dms_SetUSBComp_t` Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.298.1 Field Documentation

8.298.1.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

8.299 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

Data Fields

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

8.299.1 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

8.299.2 Field Documentation

8.299.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

8.299.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

8.299.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

8.300 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

Data Fields

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

8.300.1 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

8.300.2 Field Documentation

8.300.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.300.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.300.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.301 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.301.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

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

8.301.2 Field Documentation

8.301.2.1 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::Tlvresult

8.302 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](#)

8.302.1 Detailed Description

This structure contains the Band Capabilities response.

Please check is_<Param_Name>_Available field for presence of optional parameters

Parameters

<i>bandCapability[OUT]</i>	<p>Bitmask of bands supported by the device</p> <ul style="list-style-type: none"> • Bit 0 - Band class 0, A-system • Bit 1 - Band class 0, B-system • Bit 2 - Band class 1, all blocks • Bit 3 - Band class 2 • Bit 4 - Band class 3, A-system • Bit 5 - Band class 4, all blocks • Bit 6 - Band class 5, all blocks • Bit 7 - GSM DCS band (1800) • Bit 8 - GSM Extended GSM (E-GSM) band (900) • Bit 9 - GSM Primary GSM (P-GSM) band (900) • Bit 10 - Band class 6 • Bit 11 - Band class 7 • Bit 12 - Band class 8 • Bit 13 - Band class 9 • Bit 14 - Band class 10 • Bit 15 - Band class 11 • Bit 16 - GSM 450 band • Bit 17 - GSM 480 band • Bit 18 - GSM 750 band • Bit 19 - GSM 850 band • Bit 20 - GSM railways GSM band (900) • Bit 21 - GSM PCS band (1900) • Bit 22 - WCDMA (Europe, Japan, and China) 2100 band • Bit 23 - WCDMA US PCS 1900 band • Bit 24 - WCDMA (Europe and China) DCS 1800 band • Bit 25 - WCDMA US 1700 band • Bit 26 - WCDMA US 850 band • Bit 27 - WCDMA Japan 800 band • Bit 28 - Band class 12 • Bit 29 - Band class 14 • Bit 30 - Reserved • Bit 31 - Band class 15 • Bits 32 through 47 - Reserved • Bit 48 - WCDMA Europe 2600 band • Bit 49 - WCDMA Europe and Japan 900 band • Bit 50 - WCDMA Japan 1700 band • Bits 51 through 55 - Reserved • Bit 56 - Band class 16 • Bit 57 - Band class 17 • Bit 58 - Band class 18 • Bit 59 - Band class 19
----------------------------	---

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

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

8.302.2 Field Documentation

8.302.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

8.302.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

8.302.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

8.302.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

8.302.2.5 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

8.303 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.303.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

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

8.303.2 Field Documentation

8.303.2.1 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult

8.304 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

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

8.304.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

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

8.304.2 Field Documentation

8.304.2.1 `packgetDyingGaspCfg*` `unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg`

8.304.2.2 `uint16_t` `unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult`

8.305 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

Data Fields

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

8.305.1 Detailed Description

This structure contains Get Dying GASP Statistics.

Parameters

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

8.305.2 Field Documentation

8.305.2.1 `packgetDyingGaspStatistics*` `unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics`

8.305.2.2 `uint16_t` `unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

8.306 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]

8.306.1 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

8.306.2 Field Documentation

8.306.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.306.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.306.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.306.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.306.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.306.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.307 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference

Data Fields

- uint32_t [ResCode](#)
- uint8_t [imgType](#)
- uint32_t [refData](#)
- uint8_t [refString](#) [15]
- uint8_t [logString](#) [255]
- uint16_t [Tlvresult](#)

8.307.1 Detailed Description

This structure is used to store Firmware Update Status

Parameters

<i>ResCode</i>	<ul style="list-style-type: none"> FW Update Result Code Values: <ul style="list-style-type: none"> 0x00000001 - Successful 0xFFFFFFFF - Unknown (due to power off reset after firmware update) 0x100000nn - File update errors while nn will be the exact error number: <ul style="list-style-type: none"> * 00 - General error 0x200000nn - NVUP update errors while nn will be the exact error number: <ul style="list-style-type: none"> * 00 - General error 0x40000nnn - FOTA update agent errors while nnn will be the exact error number: <ul style="list-style-type: none"> * 000 ~ 0FF - Insignia defined error code * 100 ~ 1FF - Sierra defined error code * See qaGobiApiTableFwDldErrorCodes.h for more detailed information 0x800000nn - FDT/SSDP reported errors while nn will be the exact error number <ul style="list-style-type: none"> * See qaGobiApiTableFwDldErrorCodes.h for more detailed information
<i>imgType</i>	<ul style="list-style-type: none"> Optional parameter Firmware image type that failed the update
<i>refData</i>	<ul style="list-style-type: none"> Optional parameter Failed image reference data This is normally the offset of the image that caused the failure
<i>refString</i>	<ul style="list-style-type: none"> Optional parameter Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.
<i>logString</i>	<ul style="list-style-type: none"> Optional parameter Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result

8.307.2 Field Documentation

8.307.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

8.307.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

8.307.2.3 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

8.307.2.4 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

8.307.2.5 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

8.307.2.6 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult

8.308 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.308.1 Detailed Description

This structure contains set Dying GASP Config 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_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

8.309 unpack_dms_UIMGetlCCID_t Struct Reference

Data Fields

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

8.309.1 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.

8.309.2 Field Documentation

8.309.2.1 uint8_t unpack_dms_UIMGetlCCID_t::String[255]

8.309.2.2 uint8_t unpack_dms_UIMGetlCCID_t::stringSize

8.309.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.310 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

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

8.310.1 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

8.310.2 Field Documentation

8.310.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

8.310.2.2 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

8.310.2.3 uint16_t unpack_fms_GetImagesPreference_t::Tlvresult

8.311 unpack_fms_GetStoredImages_t Struct Reference

Data Fields

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

8.311.1 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

8.311.2 Field Documentation

8.311.2.1 **FMSImageList** `unpack_fms_GetStoredImages_t::imageList`

8.311.2.2 `uint32_t` `unpack_fms_GetStoredImages_t::imagelistSize`

8.311.2.3 `uint16_t` `unpack_fms_GetStoredImages_t::Tlvresult`

8.312 `unpack_fms_SetImagesPreference_t` Struct Reference

Data Fields

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

8.312.1 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

8.312.2 Field Documentation

8.312.2.1 `uint8_t` `unpack_fms_SetImagesPreference_t::ImageTypes`[255]

8.312.2.2 `uint32_t` `unpack_fms_SetImagesPreference_t::ImageTypesSize`

8.312.2.3 `uint16_t` `unpack_fms_SetImagesPreference_t::Tlvresult`

8.313 `unpack_loc_BestAvailPos_Ind_t` Struct Reference

Data Fields

- uint32_t [status](#)
- uint32_t * [pXid](#)
- uint64_t * [pLatitude](#)
- uint64_t * [pLongitude](#)
- uint32_t * [pHorUncCircular](#)
- uint32_t * [pAltitudeWrtEllipsoid](#)
- uint32_t * [pVertUnc](#)
- uint64_t * [pTimestampUtc](#)
- uint32_t * [pTimeUnc](#)
- uint32_t * [pHorUncEllipseSemiMinor](#)
- uint32_t * [pHorUncEllipseSemiMajor](#)
- uint32_t * [pHorUncEllipseOrientAzimuth](#)
- uint8_t * [pHorCirConf](#)
- uint8_t * [pHorEllpConf](#)
- uint32_t * [pHorReliability](#)
- uint32_t * [pSpeedHorizontal](#)
- uint32_t * [pSpeedUnc](#)
- uint32_t * [pAltitudeWrtMeanSeaLevel](#)
- uint8_t * [pVertConfidence](#)
- uint32_t * [pVertReliability](#)
- uint32_t * [pSpeedVertical](#)
- uint32_t * [pSpeedVerticalUnc](#)
- uint32_t * [pHeading](#)
- uint32_t * [pHeadingUnc](#)
- uint32_t * [pMagneticDeviation](#)
- uint32_t * [pTechnologyMask](#)
- [loc_precisionDilution](#) * [pPrecisionDilution](#)
- [loc_gpsTime](#) * [pGpsTime](#)
- uint32_t * [pTimeSrc](#)
- [loc_sensorDataUsage](#) * [pSensorDataUsage](#)
- [loc_svUsedforFix](#) * [pSvUsedforFix](#)
- uint16_t [Tlvresult](#)

8.313.1 Detailed Description

This structure contains Best Available Position

Parameters

<i>status</i>	<ul style="list-style-type: none"> Valid values: <ul style="list-style-type: none"> eQMI_LOC_SUCCESS (0) - Request was completed successfully eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline eQMI_LOC_TIMEOUT (6) - Request failed because it timed out eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure
<i>xid</i>	Transaction ID that was specified in the Get Best Available Position request.
<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>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> Altitude With Respect to WGS84 Ellipsoid. Units - Meters Range -500 to 15883
<i>pVertUnc</i>	<ul style="list-style-type: none"> Vertical uncertainty. Units - Meters

<i>pTimestampUtc</i>	<ul style="list-style-type: none"> • UTC timestamp • Units - Milliseconds since Jan. 1, 1970
<i>pTimeUnc</i>	<ul style="list-style-type: none"> • Time uncertainty. • Units - Milliseconds
<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>pHorCirConf</i>	<ul style="list-style-type: none"> • Horizontal circular uncertainty confidence • Units: Precent • Range: 0 to 99
<i>pHorEllpConf</i>	<ul style="list-style-type: none"> • Horizontal elliptical uncertainty confidence • Units: Precent • Range: 0 to 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-MeanSeaLevel</i>	<ul style="list-style-type: none"> • Altitude With Respect to Sea Level. • 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>pSpeedVertical-Unc</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>-pGpsTime</i>	<ul style="list-style-type: none"> • See loc_gpsTime for more information

<i>pTimeSrc</i>	<ul style="list-style-type: none"> • Values <ul style="list-style-type: none"> – 0 - Invalid time. – 1 - Time is set by the 1X system. – 2 - Time is set by WCDMA/GSM time tagging. – 3 - Time is set by an external injection. – 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite. – 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites. – 6 - Both time of the week and the GPS week number are known. – 7 - Time is set by the position engine after the fix is obtained – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.
<i>-pSensorData-Usage</i>	<ul style="list-style-type: none"> • See loc_sensorDataUsage for more information
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> • See loc_svUsedforFix for more information

8.313.2 Field Documentation

8.313.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

8.313.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

8.313.2.3 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

8.313.2.4 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

8.313.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

8.313.2.6 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

8.313.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

8.313.2.8 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

8.313.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

8.313.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

8.313.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

- 8.313.2.12 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor`
- 8.313.2.13 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude`
- 8.313.2.14 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude`
- 8.313.2.15 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation`
- 8.313.2.16 `loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution`
- 8.313.2.17 `loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage`
- 8.313.2.18 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal`
- 8.313.2.19 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc`
- 8.313.2.20 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical`
- 8.313.2.21 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc`
- 8.313.2.22 `loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix`
- 8.313.2.23 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask`
- 8.313.2.24 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc`
- 8.313.2.25 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc`
- 8.313.2.26 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc`
- 8.313.2.27 `uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence`
- 8.313.2.28 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability`
- 8.313.2.29 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc`
- 8.313.2.30 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid`
- 8.313.2.31 `uint32_t unpack_loc_BestAvailPos_Ind_t::status`
- 8.313.2.32 `uint16_t unpack_loc_BestAvailPos_Ind_t::Tlvresult`

8.314 `unpack_loc_Delete_Assist_Data_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.314.1 Detailed Description

This structure contains LOC delete assist data unpack

Parameters

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

8.314.2 Field Documentation

8.314.2.1 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

8.315 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

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

8.315.1 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 on2 - Location engine is off
<i>Tlvresult</i>	<ul style="list-style-type: none">unpack result

8.315.2 Field Documentation

8.315.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.315.2.2 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

8.316 unpack_loc_EventRegister_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.316.1 Detailed Description

This structure contains Event Register unpack

Parameters

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

8.316.2 Field Documentation

8.316.2.1 uint16_t unpack_loc_EventRegister_t::Tlvresult

8.317 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](#)

8.317.1 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 – 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large. – 9 - Time is set after decoding GLO satellites. – 10- Time is set after transforming the GPS to GLO time – 11- Time is set by the sleep time tag provided by the WCDMA network. – 12- Time is set by the sleep time tag provided by the GSM network – 13- Source of the time is unknown – 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state – 15- Time is set after decoding QZSS satellites. – 16- Time is set after decoding BDS satellites.
<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

8.317.2 Field Documentation

8.317.2.1 `uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed`

8.317.2.2 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid`

8.317.2.3 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel`

8.317.2.4 `uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId`

8.317.2.5 `loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime`

- 8.317.2.6 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading
- 8.317.2.7 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc
- 8.317.2.8 uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence
- 8.317.2.9 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability
- 8.317.2.10 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular
- 8.317.2.11 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth
- 8.317.2.12 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor
- 8.317.2.13 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor
- 8.317.2.14 uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude
- 8.317.2.15 uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds
- 8.317.2.16 uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude
- 8.317.2.17 uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation
- 8.317.2.18 loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution
- 8.317.2.19 loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage
- 8.317.2.20 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal
- 8.317.2.21 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc
- 8.317.2.22 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical
- 8.317.2.23 loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix
- 8.317.2.24 uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask
- 8.317.2.25 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc
- 8.317.2.26 uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc
- 8.317.2.27 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc
- 8.317.2.28 uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence
- 8.317.2.29 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability
- 8.317.2.30 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc
- 8.317.2.31 uint8_t unpack_loc_PositionRpt_Ind_t::sessionId
- 8.317.2.32 uint32_t unpack_loc_PositionRpt_Ind_t::sessionIdStatus
- 8.317.2.33 uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult

8.318 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

Data Fields

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

8.318.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

Parameters

<i>status</i>	<ul style="list-style-type: none"> • Valid values <ul style="list-style-type: none"> – 0 - Request was completed successfully – 1 - Request failed because of a general failure. – 2 - Request failed because it is not supported. – 3 - Request failed because it contained invalid parameters – 4 - Request failed because the engine is busy – 5 - Request failed because the phone is offline – 6 - Request failed because it timed out – 7 - Request failed because an undefined configuration was requested – 8 - engine could not allocate sufficient memory – 9 - Request failed because the maximum number of Geofences are already programmed – 10 -Location service failed because of an XTRA version-based file format check failure
<i>Tlvresult</i>	<ul style="list-style-type: none"> • unpack result

8.318.2 Field Documentation

8.318.2.1 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

8.318.2.2 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

8.319 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.319.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

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

8.319.2 Field Documentation

8.319.2.1 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

8.320 unpack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.320.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

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

8.320.2 Field Documentation

8.320.2.1 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

8.321 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.321.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

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

8.321.2 Field Documentation

8.321.2.1 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.322 unpack_loc_Start_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.322.1 Detailed Description

This structure contains Start LOC unpack

Parameters

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

8.322.2 Field Documentation

8.322.2.1 `uint16_t unpack_loc_Start_t::Tlvresult`

8.323 `unpack_loc_Stop_t` Struct Reference

Data Fields

- `uint16_t` [Tlvresult](#)

8.323.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

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

8.323.2 Field Documentation

8.323.2.1 `uint16_t unpack_loc_Stop_t::Tlvresult`

8.324 `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](#)

8.324.1 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

8.324.2 Field Documentation

8.324.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.324.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.324.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.324.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.324.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.324.2.6 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

8.324.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

8.324.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

8.324.2.9 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

8.324.2.10 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

8.324.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

8.325 unpack_nas_GetHomeNetwork_t Struct Reference

Data Fields

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

8.325.1 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
------------	--

8.325.2 Field Documentation

8.325.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

8.325.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

8.325.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

8.325.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

8.325.2.5 uint16_t unpack_nas_GetHomeNetwork_t::sid

8.326 unpack_nas_GetNetworkPreference_t Struct Reference

Data Fields

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

8.326.1 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
------------------	---

8.326.2 Field Documentation

8.326.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.326.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.326.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.326.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.327 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

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

8.327.1 Detailed Description

Parameters

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

8.327.2 Field Documentation

8.327.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

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

8.328 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]

8.328.1 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

8.328.2 Field Documentation

8.328.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.328.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.328.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.328.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.328.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

8.328.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

8.328.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

8.328.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

8.328.2.9 uint8_t unpack_nas_GetServingNetwork_t::Radiolfaces[255]

8.328.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadiolfacesSize

8.328.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

8.328.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

8.328.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

8.329 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

Data Fields

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

8.329.1 Detailed Description

Parameters

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

8.329.2 Field Documentation

8.329.2.1 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]`

8.329.2.2 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen`

8.330 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

- `uint32_t len`
- signed char `rssi` [8]
- `uint32_t radio` [8]

8.330.1 Detailed Description

Parameters

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

8.330.2 Field Documentation

8.330.2.1 `uint32_t unpack_nas_GetSignalStrengths_t::len`

8.330.2.2 `uint32_t unpack_nas_GetSignalStrengths_t::radio[8]`

8.330.2.3 signed char `unpack_nas_GetSignalStrengths_t::rssi[8]`

8.331 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`

8.331.1 Detailed Description

Parameters

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

8.331.2 Field Documentation

8.331.2.1 `nas_QmiNas3GppNetworkInfo*` `unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances`

8.331.2.2 `uint8_t*` `unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize`

8.331.2.3 `nas_QmisNasPcsDigit*` `unpack_nas_PerformNetworkScan_t::pPCSInstance`

8.331.2.4 `uint8_t*` `unpack_nas_PerformNetworkScan_t::pPCSInstanceSize`

8.331.2.5 `nas_QmiNas3GppNetworkRAT*` `unpack_nas_PerformNetworkScan_t::pRATInstance`

8.331.2.6 `uint8_t*` `unpack_nas_PerformNetworkScan_t::pRATInstanceSize`

8.331.2.7 `uint32_t*` `unpack_nas_PerformNetworkScan_t::pScanResult`

8.332 `unpack_nas_SetDataCapabilitiesCallback_ind_t` Struct Reference

Data Fields

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

8.332.1 Detailed Description

Parameters

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

8.332.2 Field Documentation

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

8.332.2.2 `uint8_t` `unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize`

8.333 `unpack_nas_SetEventReportInd_t` Struct Reference

Data Fields

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

8.333.1 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

8.333.2 Field Documentation

8.333.2.1 nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv

8.333.2.2 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

8.333.2.3 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

8.333.2.4 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

8.334 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference

Data Fields

- [nas_PhyCaAggScellIndType](#) sPhyCaAggScellIndType
- [nas_PhyCaAggScellIDIBw](#) sPhyCaAggScellIDIBw
- [nas_PhyCaAggScellInfo](#) sPhyCaAggScellInfo
- [nas_PhyCaAggPcellInfo](#) sPhyCaAggPcellInfo
- [nas_PhyCaAggScellIndex](#) sPhyCaAggScellIndex

8.334.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

Parameters

<i>pPhyCaAgg-ScellIndType</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellIndType for more information.
<i>sPhyCaAgg-ScellIDIBw</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellIDIBw for more information.
<i>sPhyCaAgg-ScellInfo</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellInfo for more information.
<i>sPhyCaAgg-PcellInfo</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggPcellInfo for more information.
<i>sPhyCaAgg-ScellIndex</i>	<ul style="list-style-type: none"> • See nas_PhyCaAggScellIndex for more information.

8.334.2 Field Documentation

8.334.2.1 nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggPcellInfo

8.334.2.2 nas_PhyCaAggScellIDIBw unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIDIBw

8.334.2.3 nas_PhyCaAggScellIndex unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndex

8.334.2.4 nas_PhyCaAggScellIndType unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndType

8.334.2.5 nas_PhyCaAggScellInfo unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellInfo

8.335 unpack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.335.1 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

8.335.2 Field Documentation

8.335.2.1 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

8.336 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- uint8_t [roaming](#)

8.336.1 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
----------------	--

8.336.2 Field Documentation

8.336.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.337 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

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

8.337.1 Detailed Description

Parameters

<i>SSInfo</i>	<ul style="list-style-type: none">• Serving system parameters information<ul style="list-style-type: none">– See NAServingSystemInfo for more details
<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack result

8.337.2 Field Documentation

8.337.2.1 [NAServingSystemInfo](#) unpack_nas_SetServingSystemCallback_ind_t::SSInfo

8.337.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

8.338 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- [NasGetLTECphyCAInfo](#) LTECphyCAInfo

- uint16_t [Tlvresult](#)

8.338.1 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

8.338.2 Field Documentation

8.338.2.1 [NasGetLTECphyCAInfo](#) [unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCAInfo](#)

8.338.2.2 [uint16_t](#) [unpack_nas_SlqsGetLTECphyCAInfo_t::Tlvresult](#)

8.339 [unpack_nas_SLQSGetNetworkTime_t](#) Struct Reference

Data Fields

- [nas_timeInfo](#) * [p3GPP2TimeInfo](#)
- [nas_timeInfo](#) * [p3GPPTimeInfo](#)

8.339.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

Parameters

<i>p3GPP2Time-Info</i>	[Optional] <ul style="list-style-type: none"> • See nas_timeInfo for more information
<i>p3GPPTimeInfo</i>	[Optional] <ul style="list-style-type: none"> • See nas_timeInfo for more information

8.339.2 Field Documentation

8.339.2.1 [nas_timeInfo*](#) [unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo](#)

8.339.2.2 [nas_timeInfo*](#) [unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo](#)

8.340 [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]

8.340.1 Field Documentation

- 8.340.1.1 char [unpack_nas_SLQSGetPLMNName_t::longName](#)[255]
- 8.340.1.2 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameCI](#)
- 8.340.1.3 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameEn](#)
- 8.340.1.4 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameLen](#)
- 8.340.1.5 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameSB](#)
- 8.340.1.6 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortName](#)[255]
- 8.340.1.7 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameCI](#)
- 8.340.1.8 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameEn](#)
- 8.340.1.9 char [unpack_nas_SLQSGetPLMNName_t::shortNameLen](#)
- 8.340.1.10 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameSB](#)
- 8.340.1.11 char [unpack_nas_SLQSGetPLMNName_t::spn](#)[255]
- 8.340.1.12 uint8_t [unpack_nas_SLQSGetPLMNName_t::spnEncoding](#)
- 8.340.1.13 uint8_t [unpack_nas_SLQSGetPLMNName_t::spnLength](#)

8.341 unpack_nas_SLQSGetservingSystem_t Struct Reference

Data Fields

- [nas_servSystem](#) [ServingSystem](#)
- uint8_t [RoamIndicatorVal](#)
- [nas_dataSrvCapabilities](#) [DataSrvCapabilities](#)
- [nas_currentPLMN](#) [CurrentPLMN](#)
- uint16_t [SystemID](#)
- uint16_t [NetworkID](#)
- uint16_t [BasestationID](#)
- uint32_t [BasestationLatitude](#)
- uint32_t [BasestationLongitude](#)
- [nas_roamIndList](#) [RoamingIndicatorList](#)

- uint8_t [DefaultRoamInd](#)
- nas_qaQmi3Gpp2TimeZone [Gpp2TimeZone](#)
- uint8_t [CDMA_P_Rev](#)
- uint8_t [GppTimeZone](#)
- uint8_t [GppNetworkDSTAdjustment](#)
- uint16_t [Lac](#)
- uint32_t [CellID](#)
- uint8_t [ConcSvcInfo](#)
- uint8_t [PRLInd](#)
- uint8_t [DTMInd](#)
- nas_detailSvcInfo [DetailedSvcInfo](#)
- nas_CDMA SysInfoExt [CDMA SystemInfoExt](#)
- uint8_t [HdrPersonality](#)
- uint16_t [TrackAreaCode](#)
- nas_callBarStatus [CallBarStatus](#)

8.341.1 Detailed Description

Parameters

<i>ServingSystem</i>	serving system info
<i>RoamIndicator-Val</i>	roaming indicator value
<i>DataSrv-Capabilities</i>	data servcie capabilities
<i>CurrentPLMN</i>	current PLMN info
<i>SystemID</i>	system id
<i>NetworkID</i>	network id
<i>BasestationID</i>	base station id
<i>Basestation-Latitude</i>	base station latitude
<i>Basestation-Longitude</i>	base station longitude
<i>Roaming-IndicatorList</i>	roaming indicator list
<i>DefaultRoamInd</i>	default roaming indicator
<i>3Gpp2TimeZone</i>	3Gpp2 time zone
<i>pCDMA_P_Rev</i>	cdma P_Rev in use
<i>3GppTimeZone</i>	3Gpp time zone
<i>GppNetworkDSTAdjustment</i>	3GPP network daylight saving adjustment
<i>Lac</i>	location area code
<i>CellID</i>	3GPP cell id
<i>ConcSvcInfo</i>	3GPP2 concurrent servcie info
<i>PRLInd</i>	3GPP2 PRL indicator
<i>DTMInd</i>	DTM indicator(GSM)
<i>DetailedSvcInfo</i>	detail servcie info
<i>CDMA System-InfoExt</i>	extra cdma system info
<i>HdrPersonality</i>	hdr personality
<i>TrackAreaCode</i>	track area code
<i>CallBarStatus</i>	call barring status

8.341.2 Field Documentation

- 8.341.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID
- 8.341.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude
- 8.341.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude
- 8.341.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus
- 8.341.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev
- 8.341.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt
- 8.341.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID
- 8.341.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo
- 8.341.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN
- 8.341.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities
- 8.341.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd
- 8.341.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo
- 8.341.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd
- 8.341.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone
- 8.341.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment
- 8.341.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone
- 8.341.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality
- 8.341.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac
- 8.341.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID
- 8.341.2.20 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd
- 8.341.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal
- 8.341.2.22 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList
- 8.341.2.23 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem
- 8.341.2.24 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID
- 8.341.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

8.342 unpack_nas_SLQSGetSignalStrength_t Struct Reference

Data Fields

- uint16_t [signalStrengthReqMask](#)
- uint16_t [rxSignalStrengthListLen](#)

- [nas_rxSignalStrengthListElement](#) rxSignalStrengthList [18]
- [uint16_t](#) ecioListLen
- [nas_ecioListElement](#) ecioList [18]
- [int32_t](#) lo
- [uint8_t](#) sinr
- [uint16_t](#) errorRateListLen
- [nas_errorRateListElement](#) errorRateList [18]
- [nas_rsrqInformation](#) rsrqInfo
- [int16_t](#) ltesnr
- [int16_t](#) ltersrp

8.342.1 Detailed Description

Parameters

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

8.342.2 Field Documentation

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

8.342.2.2 [uint16_t](#) unpack_nas_SLQSGetSignalStrength_t::ecioListLen

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

8.342.2.4 [uint16_t](#) unpack_nas_SLQSGetSignalStrength_t::errorRateListLen

8.342.2.5 [int32_t](#) unpack_nas_SLQSGetSignalStrength_t::lo

8.342.2.6 [int16_t](#) unpack_nas_SLQSGetSignalStrength_t::ltersrp

8.342.2.7 [int16_t](#) unpack_nas_SLQSGetSignalStrength_t::ltesnr

8.342.2.8 [nas_rsrqInformation](#) unpack_nas_SLQSGetSignalStrength_t::rsrqInfo

8.342.2.9 [nas_rxSignalStrengthListElement](#) unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]

8.342.2.10 [uint16_t](#) unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen

8.342.2.11 [uint16_t](#) unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask

8.342.2.12 [uint8_t](#) unpack_nas_SLQSGetSignalStrength_t::sinr

8.343 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](#)

8.343.1 Detailed Description

Parameters

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

8.343.2 Field Documentation

- 8.343.2.1 nas_AddCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo
- 8.343.2.2 nas_AddSysInfo* unpack_nas_SLQSGetSysInfo_t::pAddGSM SysInfo
- 8.343.2.3 uint16_t* unpack_nas_SLQSGetSysInfo_t::pAddHDR SysInfo
- 8.343.2.4 uint16_t* unpack_nas_SLQSGetSysInfo_t::pAddLTE SysInfo
- 8.343.2.5 nas_AddSysInfo* unpack_nas_SLQSGetSysInfo_t::pAddWCDMA SysInfo
- 8.343.2.6 nas_SrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo
- 8.343.2.7 nas_CDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo
- 8.343.2.8 nas_CallBarringSysInfo* unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo
- 8.343.2.9 uint8_t* unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo
- 8.343.2.10 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pGSM SrvStatusInfo
- 8.343.2.11 nas_GSM SysInfo* unpack_nas_SLQSGetSysInfo_t::pGSM SysInfo
- 8.343.2.12 nas_SrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pHDR SrvStatusInfo
- 8.343.2.13 nas_HDR SysInfo* unpack_nas_SLQSGetSysInfo_t::pHDR SysInfo
- 8.343.2.14 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pLTE SrvStatusInfo
- 8.343.2.15 nas_LTE SysInfo* unpack_nas_SLQSGetSysInfo_t::pLTE SysInfo
- 8.343.2.16 uint8_t* unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo
- 8.343.2.17 nas_CallBarringSysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo
- 8.343.2.18 uint8_t* unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo
- 8.343.2.19 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMA SrvStatusInfo
- 8.343.2.20 nas_WCDMA SysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMA SysInfo

8.344 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](#)

8.344.1 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 – Bit 59 - Band Class 19 – Bit 60 to 64 - Reserved
------------------	--

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

<i>pNetSelPref</i>	<ul style="list-style-type: none"> Optional parameter indicating network selection preference Values: <ul style="list-style-type: none"> 0x00 - Automatic network selection 0x01 - Manual network selection.
<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>pGWAcqOrderPref</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

8.344.2 Field Documentation

8.344.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

8.344.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

8.344.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

8.344.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

8.344.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

8.344.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

8.344.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

8.344.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

8.344.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

8.345 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](#)

8.345.1 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>pWCDMAInfoLTENeighborCell</i>	<ul style="list-style-type: none"> • See nas_WCDMAInfoLTENeighborCell for more information.

8.345.2 Field Documentation

8.345.2.1 [nas_CDMAInfo](#)* [unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo](#)

8.345.2.2 [nas_GERANInfo](#)* [unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo](#)

8.345.2.3 [nas_LTEInfoInterfreq](#)* [unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq](#)

8.345.2.4 [nas_LTEInfoIntrafreq](#)* [unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq](#)

8.345.2.5 [nas_LTEInfoNeighboringGSM](#)* [unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM](#)

8.345.2.6 [nas_LTEInfoNeighboringWCDMA](#)* [unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA](#)

8.345.2.7 uint32_t* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

8.345.2.8 nas_UMTSInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

8.345.2.9 nas_WCDMAInfoLTENeighborCell* unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

8.346 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

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

8.346.1 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

8.346.2 Field Documentation

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

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

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

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

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

8.347 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

Data Fields

- [nas_UniversalTime](#) [universalTime](#)
- [uint8_t](#) * [pTimeZone](#)
- [uint8_t](#) * [pDayltSavAdj](#)
- [uint8_t](#) * [pRadioInterface](#)

8.347.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

Parameters

<i>universalTime</i>	<ul style="list-style-type: none"> • See nas_UniversalTime for more information.
<i>pTimeZone</i>	<ul style="list-style-type: none"> • Time Zone. • Offset from Universal time, i.e., the difference between local time and Universal time, in increments of 15 min (signed value).
<i>pDayltSavAdj</i>	<ul style="list-style-type: none"> • Daylight Saving Adjustment. • Daylight saving adjustment in hr. <ul style="list-style-type: none"> – Possible values: 0, 1, and 2.
<i>pRadioInterface</i>	<ul style="list-style-type: none"> • Radio interface from which the information comes • Values <ul style="list-style-type: none"> – 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - NAS_RADIO_IF_CDMA_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

8.347.2 Field Documentation

8.347.2.1 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

8.347.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

8.347.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

8.347.2.4 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

8.348 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference

Data Fields

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

8.348.1 Detailed Description

Parameters

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

8.348.2 Field Documentation

8.348.2.1 **cdmaSSInfo*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo8.348.2.2 **int8_t*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo8.348.2.3 **hdrSSInfo*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo8.348.2.4 **lteSSInfo*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo8.348.2.5 **int8_t*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp8.348.2.6 **tdscdmaSigInfoExt*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt8.348.2.7 **cdmaSSInfo*** unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

8.349 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

Data Fields

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

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

8.349.2 Field Documentation

8.349.2.1 **nas_CommInfo** unpack_nas_SLQSNasSwiModemStatus_t::commonInfo8.349.2.2 **nas_LTEInfo*** unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo

8.350 unpack_nas_SLQSNasSwtOTAMessageCallback_ind_t Struct Reference

Data Fields

- [NASQmiCbkNasSwtOTAMessageInd](#) Info
- uint16_t [Tlvresult](#)

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

8.350.2 Field Documentation

8.350.2.1 [NASQmiCbkNasSwtOTAMessageInd](#) unpack_nas_SLQSNasSwtOTAMessageCallback_ind_t::Info

8.350.2.2 uint16_t unpack_nas_SLQSNasSwtOTAMessageCallback_ind_t::Tlvresult

8.351 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

Data Fields

- [NASQmiCbkNasSystemSelPrefInd](#) Info
- uint16_t [Tlvresult](#)

8.351.1 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

8.351.2 Field Documentation

8.351.2.1 [NASQmiCbkNasSystemSelPrefInd](#) unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

8.351.2.2 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult

8.352 unpack_nas_SLQSSwtGetLteCQI_t Struct Reference

Data Fields

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

8.352.1 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

8.352.2 Field Documentation

8.352.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

8.352.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

8.352.2.3 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

8.352.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

8.353 unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference

Data Fields

- nas_SrvStatusInfo * [pCDMASrvStatusInfo](#)
- nas_SrvStatusInfo * [pHDRSrvStatusInfo](#)
- nas_GSMSrvStatusInfo * [pGSMSrvStatusInfo](#)
- nas_GSMSrvStatusInfo * [pWCDMASrvStatusInfo](#)
- nas_GSMSrvStatusInfo * [pLTESrvStatusInfo](#)
- nas_CDMASysInfo * [pCDMASysInfo](#)
- nas_HDRSysInfo * [pHDRSysInfo](#)
- nas_GSMSysInfo * [pGSMSysInfo](#)
- nas_WCDMASysInfo * [pWCDMASysInfo](#)

- [nas_LTESysInfo](#) * [pLTESysInfo](#)
- [nas_AddCDMASysInfo](#) * [pAddCDMASysInfo](#)
- [uint16_t](#) * [pAddHDRSysInfo](#)
- [nas_AddSysInfo](#) * [pAddGSM SysInfo](#)
- [nas_AddSysInfo](#) * [pAddWCDMASysInfo](#)
- [uint16_t](#) * [pAddLTESysInfo](#)
- [nas_CallBarringSysInfo](#) * [pGSMCallBarringSysInfo](#)
- [nas_CallBarringSysInfo](#) * [pWCDMACallBarringSysInfo](#)
- [uint8_t](#) * [pLTEVoiceSupportSysInfo](#)
- [uint8_t](#) * [pGSMCipherDomainSysInfo](#)
- [uint8_t](#) * [pWCDMACipherDomainSysInfo](#)
- [uint8_t](#) * [pSysInfoNoChange](#)

8.353.1 Detailed Description

Parameters

<i>pCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> • See SrvStatusInfo for more information.
<i>pHDRSrvStatusInfo</i>	<ul style="list-style-type: none"> • See SrvStatusInfo for more information.
<i>pGSMSrvStatusInfo</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pWCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> • See GSMSrvStatusInfo for more information.
<i>pLTESrvStatusInfo</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>pWCDMASysInfo</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>pAddCDMASysInfo</i>	<ul style="list-style-type: none"> • See AddCDMASysInfo for more information.
<i>pAddHDRSysInfo</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

8.353.2 Field Documentation

8.353.2.1 **nas_AddCDMASysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

8.353.2.2 **nas_AddSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

8.353.2.3 **uint16_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

8.353.2.4 **uint16_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

- 8.353.2.5 `nas_AddSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo`
- 8.353.2.6 `nas_SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo`
- 8.353.2.7 `nas_CDMASysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo`
- 8.353.2.8 `nas_CallBarringSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo`
- 8.353.2.9 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo`
- 8.353.2.10 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo`
- 8.353.2.11 `nas_GSMSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo`
- 8.353.2.12 `nas_SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pHDSrvStatusInfo`
- 8.353.2.13 `nas_HDRSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSysInfo`
- 8.353.2.14 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo`
- 8.353.2.15 `nas_LTESysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo`
- 8.353.2.16 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo`
- 8.353.2.17 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange`
- 8.353.2.18 `nas_CallBarringSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo`
- 8.353.2.19 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo`
- 8.353.2.20 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo`
- 8.353.2.21 `nas_WCDMASysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo`

8.354 `unpack_omaDmConfigTlv_t` Struct Reference

Data Fields

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

8.354.1 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>	<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>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)

8.354.2 Field Documentation

8.354.2.1 `uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]`

8.354.2.2 `uint16_t unpack_omaDmConfigTlv_t::alertmsglength`

8.354.2.3 `uint8_t unpack_omaDmConfigTlv_t::state`

8.354.2.4 `uint8_t unpack_omaDmConfigTlv_t::userInputReq`

8.354.2.5 `uint16_t unpack_omaDmConfigTlv_t::userInputTimeout`

8.355 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`

8.355.1 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

8.355.2 Field Documentation

- 8.355.2.1 `uint8_t unpack_omaDmFotaTlv_t::description[256]`
- 8.355.2.2 `uint16_t unpack_omaDmFotaTlv_t::descriptionlength`
- 8.355.2.3 `uint32_t unpack_omaDmFotaTlv_t::fwdloadsize`
- 8.355.2.4 `uint32_t unpack_omaDmFotaTlv_t::fwloadComplete`
- 8.355.2.5 `uint16_t unpack_omaDmFotaTlv_t::namelength`
- 8.355.2.6 `uint8_t unpack_omaDmFotaTlv_t::package_name[256]`
- 8.355.2.7 `uint8_t unpack_omaDmFotaTlv_t::sessionType`
- 8.355.2.8 `uint8_t unpack_omaDmFotaTlv_t::severity`
- 8.355.2.9 `uint8_t unpack_omaDmFotaTlv_t::state`
- 8.355.2.10 `uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus`
- 8.355.2.11 `uint8_t unpack_omaDmFotaTlv_t::userInputReq`
- 8.355.2.12 `uint16_t unpack_omaDmFotaTlv_t::userInputTimeout`
- 8.355.2.13 `uint8_t unpack_omaDmFotaTlv_t::version[256]`
- 8.355.2.14 `uint16_t unpack_omaDmFotaTlv_t::versionlength`

8.356 unpack_omaDmNotificationsTlv_t Struct Reference

Data Fields

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

8.356.1 Field Documentation

- 8.356.1.1 `uint8_t unpack_omaDmNotificationsTlv_t::notification`
- 8.356.1.2 `uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus`

8.357 unpack_qmi_t Struct Reference

Data Fields

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

8.357.1 Detailed Description

qmi response context

Parameters

out	<i>type</i>	message type
out	<i>msgid</i>	message id
out	<i>xid</i>	transaction id

8.357.2 Field Documentation

8.357.2.1 uint16_t unpack_qmi_t::msgid

8.357.2.2 enum msgtype unpack_qmi_t::type

8.357.2.3 uint16_t unpack_qmi_t::xid

8.358 unpack_qos_dataRate_t Struct Reference

Data Fields

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

8.358.1 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)

8.358.2 Field Documentation

8.358.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

8.358.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

8.359 unpack_qos_IPv4Addr_t Struct Reference

Data Fields

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

8.359.1 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

8.359.2 Field Documentation

8.359.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

8.359.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

8.360 unpack_qos_IPv6Addr_t Struct Reference

Data Fields

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

8.360.1 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.

8.360.2 Field Documentation

8.360.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

8.360.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

8.361 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

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

8.361.1 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) 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

8.361.2 Field Documentation

8.361.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.361.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.362 unpack_qos_pktErrRate_t Struct Reference

Data Fields

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

8.362.1 Detailed Description

This structure contains the IP flow packet error rate

Parameters

<i>multiplier</i>	Factor m in calculating packet error rate: $E = m * 10^{**}(-p)$
<i>exponent</i>	Factor p in calculating packet error rate (see above)

8.362.2 Field Documentation

8.362.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.362.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.363 unpack_qos_Port_t Struct Reference

Data Fields

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

8.363.1 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

8.363.2 Field Documentation

8.363.2.1 uint16_t unpack_qos_Port_t::port

8.363.2.2 uint16_t unpack_qos_Port_t::range

8.364 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

8.364.1 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.

8.364.2 Field Documentation

- 8.364.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`
- 8.364.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`
- 8.364.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`
- 8.364.2.4 `uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters`
- 8.364.2.5 `uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters`
- 8.364.2.6 `unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState`
- 8.364.2.7 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]`
- 8.364.2.8 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted`
- 8.364.2.9 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]`
- 8.364.2.10 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted`

8.365 `unpack_qos_QosFlowInfoState_t` Struct Reference

Data Fields

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

8.365.1 Detailed Description

This structure contains QoS flow state

Parameters

<i>id</i>	QoS identifier
<i>isNewFlow</i>	<ul style="list-style-type: none"> • 1 – Newly added flow • 0 – Existing flow
<i>state</i>	This indicates that the flow that was added/modified/deleted: <ul style="list-style-type: none"> • 0x01 – Flow activated • 0x02 – Flow modified • 0x03 – Flow deleted • 0x04 – Flow suspended • 0x05 – Flow enabled • 0x06 – Flow disabled

8.365.2 Field Documentation

8.365.2.1 uint32_t unpack_qos_QoSFlowInfoState_t::id

8.365.2.2 uint8_t unpack_qos_QoSFlowInfoState_t::isNewFlow

8.365.2.3 uint8_t unpack_qos_QoSFlowInfoState_t::state

8.366 unpack_qos_SLQSQoSGetNetworkStatus_t Struct Reference

Data Fields

- uint8_t [NWQoSStatus](#)

8.366.1 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
--------------------	--

8.366.2 Field Documentation

8.366.2.1 uint8_t unpack_qos_SLQSQoSGetNetworkStatus_t::NWQoSStatus

8.367 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](#)

8.367.1 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

8.367.2 Field Documentation

8.367.2.1 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl`

8.367.2.2 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl_ext`

8.367.2.3 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl_ext2`

8.367.2.4 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul`

8.367.2.5 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul_ext`

8.367.2.6 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul_ext2`

8.367.2.7 uint32_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::apnId

8.368 unpack_qos_SLQSQoSswiReadDataStats_t Struct Reference

Data Fields

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

8.368.1 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

8.368.2 Field Documentation

- 8.368.2.1 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId`
- 8.368.2.2 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow`
- 8.368.2.3 `unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]`
- 8.368.2.4 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes`
- 8.368.2.5 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt`
- 8.368.2.6 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes`
- 8.368.2.7 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp`
- 8.368.2.8 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt`
- 8.368.2.9 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp`

8.369 `unpack_qos_SLQSSetQosEventCallback_ind_t` Struct Reference

Data Fields

- `uint8_t NumFlows`
- `unpack_qos_QosFlowInfo_t QosFlowInfo` [8]

8.369.1 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

8.369.2 Field Documentation

- 8.369.2.1 `uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows`
- 8.369.2.2 `unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]`

8.370 `unpack_qos_SLQSSetQosNWStatusCallback_ind_t` Struct Reference

Data Fields

- `uint8_t status`

8.370.1 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

8.370.2 Field Documentation

8.370.2.1 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

8.371 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference

Data Fields

- `uint16_t event`

8.371.1 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
--------------	---

8.371.2 Field Documentation

8.371.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

8.372 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference

Data Fields

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

8.372.1 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 • 0x16 - QMI_NETWORK_BUSY

8.372.2 Field Documentation

8.372.2.1 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event

8.372.2.2 uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id

8.372.2.3 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason

8.372.2.4 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status

8.373 unpack_qos_swiQosFilter_t Struct Reference

Data Fields

- uint8_t [index](#)
- uint8_t [version](#)
- uint8_t [is_IPv4SrcAddr_Available](#)
- [unpack_qos_IPv4Addr_t](#) [IPv4SrcAddr](#)
- uint8_t [is_IPv4DstAddr_Available](#)
- [unpack_qos_IPv4Addr_t](#) [IPv4DstAddr](#)
- uint8_t [is_NxtHdrProto_Available](#)
- uint8_t [NxtHdrProto](#)
- uint8_t [is_IPv4Tos_Available](#)
- [unpack_qos_Tos_t](#) [IPv4Tos](#)
- uint8_t [is_IPv6SrcAddr_Available](#)
- [unpack_qos_IPv6Addr_t](#) [IPv6SrcAddr](#)
- uint8_t [is_IPv6DstAddr_Available](#)
- [unpack_qos_IPv6Addr_t](#) [IPv6DstAddr](#)
- uint8_t [is_IPv6TrafCls_Available](#)
- [unpack_qos_IPv6TrafCls_t](#) [IPv6TrafCls](#)
- uint8_t [is_IPv6Label_Available](#)
- uint32_t [IPv6Label](#)
- uint8_t [is_TCPSrcPort_Available](#)
- [unpack_qos_Port_t](#) [TCPSrcPort](#)
- uint8_t [is_TCPDstPort_Available](#)
- [unpack_qos_Port_t](#) [TCPDstPort](#)
- uint8_t [is_UDPSrcPort_Available](#)
- [unpack_qos_Port_t](#) [UDPSrcPort](#)
- uint8_t [is_UDPDstPort_Available](#)
- [unpack_qos_Port_t](#) [UDPDstPort](#)
- uint8_t [is_EspSpi_Available](#)
- uint32_t [EspSpi](#)
- uint8_t [is_Precedence_Available](#)
- uint16_t [Precedence](#)
- uint8_t [is_Id_Available](#)
- uint16_t [Id](#)
- uint8_t [is_TransrcPort_Available](#)
- [unpack_qos_Port_t](#) [TransrcPort](#)
- uint8_t [is_TransdstPort_Available](#)
- [unpack_qos_Port_t](#) [TransdstPort](#)

8.373.1 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 protocolfilter 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

8.373.2 Field Documentation

8.373.2.1 `uint32_t unpack_qos_swiQosFilter_t::EspSpi`

8.373.2.2 `uint16_t unpack_qos_swiQosFilter_t::Id`

8.373.2.3 `uint8_t unpack_qos_swiQosFilter_t::index`

8.373.2.4 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr`

8.373.2.5 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr`

8.373.2.6 `unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos`

8.373.2.7 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr`

8.373.2.8 `uint32_t unpack_qos_swiQosFilter_t::IPv6Label`

8.373.2.9 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr`

8.373.2.10 `unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls`

8.373.2.11 `uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available`

8.373.2.12 `uint8_t unpack_qos_swiQosFilter_t::is_Id_Available`

8.373.2.13 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available`

8.373.2.14 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available`

8.373.2.15 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available`

8.373.2.16 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available`

8.373.2.17 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available`

8.373.2.18 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available`

8.373.2.19 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available`

8.373.2.20 `uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available`

8.373.2.21 `uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available`

8.373.2.22 `uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available`

8.373.2.23 `uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available`

8.373.2.24 `uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available`

8.373.2.25 `uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available`

8.373.2.26 `uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available`

8.373.2.27 `uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available`

8.373.2.28 `uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto`

8.373.2.29 `uint16_t unpack_qos_swiQosFilter_t::Precedence`

8.373.2.30 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort`

8.373.2.31 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort`

8.373.2.32 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort`

8.373.2.33 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort`

8.373.2.34 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort`

8.373.2.35 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort`

8.373.2.36 `uint8_t unpack_qos_swiQosFilter_t::version`

8.374 `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](#)

8.374.1 Detailed Description

This structure contains the QoS Flow Request.

Please check is_<Param_Name>_Available field for presence of optional parameters

Parameters

<i>index</i>	<ul style="list-style-type: none"> • Mandatory parameter • IP flow index • Integer that uniquely identifies each flow instance • Unique index must be assigned by the control point to every flow_spec instance
<i>ProfileId3GPP2</i>	<ul style="list-style-type: none"> • IP flow 3GPP2 profile ID • A profile ID is shorthand for a defined set of QoS flow parameters specified by the network; to be present while requesting QoS for a CDMA device
<i>val_3GPP2Pri</i>	<ul style="list-style-type: none"> • IP flow 3GPP2 flow priority • Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices
<i>TrafficClass</i>	<ul style="list-style-type: none"> • IP flow traffic class • Integer that designates the requested traffic class: <ul style="list-style-type: none"> • 0 – Conversational • 1 – Streaming • 2 – Interactive • 3 – Background
<i>DataRate</i>	<ul style="list-style-type: none"> • IP flow data rate min max • See unpack_qos_dataRate_t for more information
<i>TokenBucket</i>	<ul style="list-style-type: none"> • IP flow data rate token bucket • See unpack_qos_tokenBucket_t for more information
<i>Latency</i>	<ul style="list-style-type: none"> • IP flow latency • Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link

<i>Jitter</i>	<ul style="list-style-type: none"> • IP flow jitter • Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link
<i>PktErrRate</i>	<ul style="list-style-type: none"> • IP flow packet error rate • See unpack_qos_pktErrRate_t for more information
<i>MinPolicedPktSz</i>	<ul style="list-style-type: none"> • IP flow minimum policed packet size • Integer that defines the minimum packet size (in bytes) that will be policed for QoS guarantees; any IP packets that are smaller than the minimum specified policed size may not receive requested QoS
<i>MaxAllowedPktSz</i>	<ul style="list-style-type: none"> • IP flow maximum allowed packet size • Integer that defines the maximum packet size (in bytes) allowed in the IP flow; any IP packets greater in size than the maximum allowed packet size are not queued for transmission
<i>val_3GPPRes-ResidualBER</i>	<ul style="list-style-type: none"> • IP flow 3GPP residual bit error rate • residual_bit_error_rate • 0 = 5×10^{-2} residual BER • 1 = 1×10^{-2} residual BER • 2 = 5×10^{-3} residual BER • 3 = 4×10^{-3} residual BER • 4 = 1×10^{-3} residual BER • 5 = 1×10^{-4} residual BER • 6 = 1×10^{-5} residual BER • 7 = 1×10^{-6} residual BER • 8 = 6×10^{-8} residual BER • Integer that indicates the undetected BER for each IP flow in the delivered packets; Applies only to 3GPP networks
<i>val_3GPPTra-HdlPri</i>	<ul style="list-style-type: none"> • 3GPP traffic handling priority • 0 – Relative traffic handling priority 1 • 1 – Relative traffic handling priority 2 • 2 – Relative traffic handling priority 3 • Defines the relative priority of the flow; applies only to 3GPP networks
<i>val_3GPPImCn</i>	<ul style="list-style-type: none"> • IP flow 3GPP IM CN flag • IM CN subsystem signaling flag: • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks

<i>val_3GPPSigInd</i>	<ul style="list-style-type: none"> • IP flow 3GPP signaling indication • 0x00 – FALSE • 0x01 – TRUE • This parameter applies only to 3GPP networks
<i>LteQci</i>	<ul style="list-style-type: none"> • LTE QoS Class Identifier • QoS Class Identifier(QCI) is a required parameter to request QoS in LTE • QCI values: <ul style="list-style-type: none"> – QCI value 0 requests the network to assign the appropriate QCI value – QCI values 1-4 are associated with guaranteed bitrates – QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored

8.374.2 Field Documentation

8.374.2.1 `unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate`

8.374.2.2 `uint8_t unpack_qos_swiQosFlow_t::index`

8.374.2.3 `uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.374.2.4 `uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.374.2.5 `uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available`

8.374.2.6 `uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.374.2.7 `uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.374.2.8 `uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.374.2.9 `uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.374.2.10 `uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.374.2.11 `uint8_t unpack_qos_swiQosFlow_t::is-TokenBucket_Available`

8.374.2.12 `uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.374.2.13 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.374.2.14 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

8.374.2.15 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available`

8.374.2.16 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPSSigInd_Available`

8.374.2.17 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available`

8.374.2.18 `uint32_t unpack_qos_swiQosFlow_t::Jitter`

- 8.374.2.19 `uint32_t unpack_qos_swiQosFlow_t::Latency`
- 8.374.2.20 `uint8_t unpack_qos_swiQosFlow_t::LteQci`
- 8.374.2.21 `uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz`
- 8.374.2.22 `uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz`
- 8.374.2.23 `unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate`
- 8.374.2.24 `uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2`
- 8.374.2.25 `unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket`
- 8.374.2.26 `uint8_t unpack_qos_swiQosFlow_t::TrafficClass`
- 8.374.2.27 `uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri`
- 8.374.2.28 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn`
- 8.374.2.29 `uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER`
- 8.374.2.30 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd`
- 8.374.2.31 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri`

8.375 `unpack_qos_tokenBucket_t` Struct Reference

Data Fields

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

8.375.1 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

8.375.2 Field Documentation

- 8.375.2.1 `uint32_t unpack_qos_tokenBucket_t::bucketSz`
- 8.375.2.2 `uint32_t unpack_qos_tokenBucket_t::peakRate`
- 8.375.2.3 `uint32_t unpack_qos_tokenBucket_t::tokenRate`

8.376 unpack_qos_Tos_t Struct Reference

Data Fields

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

8.376.1 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.

8.376.2 Field Documentation

8.376.2.1 uint8_t unpack_qos_Tos_t::mask

8.376.2.2 uint8_t unpack_qos_Tos_t::val

8.377 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](#)

8.377.1 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

8.377.2 Field Documentation

8.377.2.1 `uint32_t unpack_QosFlowStat_t::bearerId`

8.377.2.2 `uint64_t unpack_QosFlowStat_t::tx_bytes`

8.377.2.3 `uint64_t unpack_QosFlowStat_t::tx_bytes_drp`

8.377.2.4 `uint32_t unpack_QosFlowStat_t::tx_pkt`

8.377.2.5 `uint32_t unpack_QosFlowStat_t::tx_pkt_drp`

8.378 `unpack_sms_SendSMS_t` Struct Reference

Data Fields

- `uint16_t` [messageID](#)
- `uint32_t` [messageFailureCode](#)

8.378.1 Detailed Description

Parameters

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

8.378.2 Field Documentation

8.378.2.1 `uint32_t unpack_sms_SendSMS_t::messageFailureCode`

8.378.2.2 `uint16_t unpack_sms_SendSMS_t::messageID`

8.379 `unpack_sms_SetNewSMSCallback_ind_t` Struct Reference

Data Fields

- `struct` [newMTMessageTlv](#) `NewMMTlv`
- `struct` [transferRouteMessageTlv](#) `TRMessageTlv`
- `struct` [messageModeTlv](#) `MMTlv`
- `struct` [sMSEtwsMessageTlv](#) `ETWSTlv`
- `struct` [eTWSPLMNInfoTlv](#) `ETWSPLMNTlv`

- struct [sMSCAddressTlv](#) [SMSTlv](#)
- struct [sMSONIMSTlv](#) [IMSTlv](#)

8.379.1 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>SMSTlv</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

8.379.2 Field Documentation

8.379.2.1 struct [eTWSPLMNInfoTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv](#)

8.379.2.2 struct [sMSEtwsMessageTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv](#)

8.379.2.3 struct [sMSONIMSTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::IMSTlv](#)

8.379.2.4 struct [messageModeTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::MMTlv](#)

8.379.2.5 struct [newMTMessageTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv](#)

8.379.2.6 struct [sMSCAddressTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::SMSTlv](#)

8.379.2.7 struct [transferRouteMessageTlv](#) [unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv](#)

8.380 unpack_sms_SetNewSMSCallback_t Struct Reference

8.381 unpack_sms_SLQSDeleteSMS_t Struct Reference

8.382 unpack_sms_SLQSGetSMS_t Struct Reference

Data Fields

- uint32_t [messageTag](#)
- uint32_t [messageFormat](#)
- uint32_t [messageSize](#)
- uint8_t [message](#) [2048]

8.382.1 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
----------------	--

8.382.2 Field Documentation

8.382.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

8.382.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

8.382.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

8.382.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

8.383 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

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

8.383.1 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

8.383.2 Field Documentation

8.383.2.1 [qmiSmsMessageList](#) [unpack_sms_SLQSGetSMSList_t::messageList](#)[255]

8.383.2.2 [uint32_t](#) [unpack_sms_SLQSGetSMSList_t::messageListSize](#)

8.384 unpack_sms_SLQSMModifySMSStatus_t Struct Reference

8.385 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

Data Fields

- [uint8_t](#) [storageType](#)
- [uint8_t](#) [messageMode](#)

8.385.1 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)

8.385.2 Field Documentation

8.385.2.1 [uint8_t](#) [unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode](#)

8.385.2.2 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType`

8.386 `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`

8.386.1 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

8.386.2 Field Documentation

8.386.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

8.386.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

8.386.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

8.386.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

8.386.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

8.386.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

8.386.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

8.386.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

8.386.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

8.386.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

8.387 unpack_swima_SLQSOMADMAAlertCallback_ind_t Struct Reference

Data Fields

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

8.387.1 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

8.387.2 Field Documentation

8.387.2.1 `uint32_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::eventType`

8.387.2.2 `unpack_omaDmConfigTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoConfig`

8.387.2.3 `unpack_omaDmFotaTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoFota`

8.387.2.4 `unpack_omaDmNotificationsTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoNotification`

8.388 `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`

8.388.1 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

8.388.2 Field Documentation

8.388.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

8.388.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

8.388.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

8.388.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

8.388.2.5 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]

8.388.2.6 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength

8.388.2.7 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount

8.388.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState

8.388.2.9 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

8.388.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity

8.388.2.11 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]

8.388.2.12 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength

8.388.2.13 uint8_t unpack_swisma_SLQSOMADMGetSessionInfo_t::Status

8.388.2.14 uint8_t unpack_swisma_SLQSOMADMGetSessionInfo_t::Time[255]

8.388.2.15 uint16_t unpack_swisma_SLQSOMADMGetSessionInfo_t::TimeLength

8.388.2.16 uint16_t unpack_swisma_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

8.389 unpack_swisma_SLQSOMADMGetSettings_t Struct Reference

Data Fields

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

8.389.1 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.

8.389.2 Field Documentation

8.389.2.1 uint8_t unpack_swima_SLQSOMADMGetSettings_t::Autosdm

8.389.2.2 uint8_t unpack_swima_SLQSOMADMGetSettings_t::FOTAdownload

8.389.2.3 uint8_t unpack_swima_SLQSOMADMGetSettings_t::FOTAupdate

8.389.2.4 uint8_t unpack_swima_SLQSOMADMGetSettings_t::FwAutoCheck

8.389.2.5 uint32_t unpack_swima_SLQSOMADMGetSettings_t::OMADMEabled

8.390 unpack_swima_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [FwAvailability](#)

8.390.1 Detailed Description

Structure that contains the responses for OMA start session command

Parameters

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

8.390.2 Field Documentation

8.390.2.1 uint32_t unpack_swima_SLQSOMADMStartSession_t::FwAvailability

8.391 unpack_uim_ChangePin_t Struct Reference

Data Fields

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

8.391.1 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.

8.391.2 Field Documentation

8.391.2.1 [uim_encryptedPIN1](#)* [unpack_uim_ChangePin_t::pEncryptedPIN1](#)

8.391.2.2 uint32_t* [unpack_uim_ChangePin_t::pIndicationToken](#)

8.391.2.3 [uim_remainingRetries](#)* [unpack_uim_ChangePin_t::pRemainingRetries](#)

8.391.2.4 uint16_t [unpack_uim_ChangePin_t::Tlvresult](#)

8.392 unpack_uim_GetCardStatus_t Struct Reference

Data Fields

- [uim_cardStatus](#) * [pCardStatus](#)
- [uim_hotSwapStatus](#) * [pHotSwapStatus](#)
- uint16_t [Tlvresult](#)

8.392.1 Detailed Description

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

Parameters

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

Note

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

8.392.2 Field Documentation

8.392.2.1 [uim_cardStatus](#)* [unpack_uim_GetCardStatus_t::pCardStatus](#)

8.392.2.2 [uim_hotSwapStatus](#)* [unpack_uim_GetCardStatus_t::pHotSwapStatus](#)

8.392.2.3 [uint16_t](#) [unpack_uim_GetCardStatus_t::Tlvresult](#)

8.393 [unpack_uim_ReadTransparent_t](#) Struct Reference

Data Fields

- [uim_cardResult](#) * [pCardResult](#)
- [uim_readResult](#) * [pReadResult](#)
- [uint32_t](#) * [pIndicationToken](#)
- [uint8_t](#) * [pEncryptedData](#)
- [uint16_t](#) [Tlvresult](#)

8.393.1 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.

8.393.2 Field Documentation

8.393.2.1 `uim_cardResult*` `unpack_uim_ReadTransparent_t::pCardResult`

8.393.2.2 `uint8_t*` `unpack_uim_ReadTransparent_t::pEncryptedData`

8.393.2.3 `uint32_t*` `unpack_uim_ReadTransparent_t::pIndicationToken`

8.393.2.4 `uim_readResult*` `unpack_uim_ReadTransparent_t::pReadResult`

8.393.2.5 `uint16_t` `unpack_uim_ReadTransparent_t::Tlvresult`

8.394 unpack_uim_SetPinProtection_t Struct Reference

Data Fields

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

8.394.1 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 <code>uim_remainingRetries</code> for more information.
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> • See <code>uim_encryptedPIN1</code> 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.

8.394.2 Field Documentation

8.394.2.1 `uim_encryptedPIN1*` `unpack_uim_SetPinProtection_t::pEncryptedPIN1`

8.394.2.2 `uint32_t*` `unpack_uim_SetPinProtection_t::pIndicationToken`

8.394.2.3 `uim_remainingRetries*` `unpack_uim_SetPinProtection_t::pRemainingRetries`

8.394.2.4 `uint16_t unpack_uim_SetPinProtection_t::Tlvresult`

8.395 `unpack_uim_SetUimSlotStatusChangeCallback_ind_t` Struct Reference

Data Fields

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

8.395.1 Detailed Description

Structure consist of card status params

Parameters

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

8.395.2 Field Documentation

8.395.2.1 `uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots`

8.395.2.2 `slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange`

8.396 `unpack_uim_SLQSUIMEventRegister_t` Struct Reference

Data Fields

- `uint32_t` [eventMask](#)

8.396.1 Detailed Description

Parameters

<i>eventMask</i>	- bit 0 - card status <ul style="list-style-type: none"> • bit 1 - SAP connection • bit 4 - physical slot status
------------------	--

8.396.2 Field Documentation

8.396.2.1 `uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask`

8.397 `unpack_uim_SLQSUIMGetSlotsStatus_t` Struct Reference

Data Fields

- `uint8_t *` [pNumberOfPhySlot](#)

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

8.397.1 Detailed Description

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

Parameters

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

8.397.2 Field Documentation

8.397.2.1 `uint8_t* unpack_uim_SLQSUIGetSlotsStatus_t::pNumberOfPhySlot`

8.397.2.2 `slots_t* unpack_uim_SLQSUIGetSlotsStatus_t::pUimSlotsStatus`

8.398 unpack_uim_SLQSUISetStatusChangeCallBack_ind_t Struct Reference

Data Fields

- [uim_cardStatus](#) * [pCardStatus](#)

8.398.1 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.
--------------------	--

8.398.2 Field Documentation

8.398.2.1 `uim_cardStatus* unpack_uim_SLQSUISetStatusChangeCallBack_ind_t::pCardStatus`

8.399 unpack_uim_UnblockPin_t Struct Reference

Data Fields

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

8.399.1 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

8.399.2 Field Documentation

8.399.2.1 `uim_encryptedPIN1*` `unpack_uim_UnblockPin_t::pEncryptedPIN1`

8.399.2.2 `uint32_t*` `unpack_uim_UnblockPin_t::pIndicationToken`

8.399.2.3 `uim_remainingRetries*` `unpack_uim_UnblockPin_t::pRemainingRetries`

8.399.2.4 `uint16_t` `unpack_uim_UnblockPin_t::Tlvresult`

8.400 `unpack_uim_VerifyPin_t` Struct Reference

Data Fields

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

8.400.1 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.

8.400.2 Field Documentation

8.400.2.1 uim_encryptedPIN1* unpack_uim_VerifyPin_t::pEncryptedPIN1

8.400.2.2 uint32_t* unpack_uim_VerifyPin_t::pIndicationToken

8.400.2.3 uim_remainingRetries* unpack_uim_VerifyPin_t::pRemainingRetries

8.400.2.4 uint16_t unpack_uim_VerifyPin_t::Tlvresult

8.401 unpack_wds_GetConnectionRate_t Struct Reference

Data Fields

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

8.401.1 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

8.401.2 Field Documentation

8.401.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

8.401.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

8.401.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

8.401.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

8.402 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]

8.402.1 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

8.402.2 Field Documentation

8.402.2.1 int8_t [unpack_wds_GetDefaultProfile_t::apnname](#)[255]

8.402.2.2 uint8_t [unpack_wds_GetDefaultProfile_t::apnsize](#)

8.402.2.3 uint32_t [unpack_wds_GetDefaultProfile_t::auth](#)

8.402.2.4 uint32_t [unpack_wds_GetDefaultProfile_t::ipaddr](#)

8.402.2.5 uint16_t [unpack_wds_GetDefaultProfile_t::ipaddrv6](#)

8.402.2.6 int8_t [unpack_wds_GetDefaultProfile_t::name](#)[255]

8.402.2.7 uint8_t [unpack_wds_GetDefaultProfile_t::namesize](#)

8.402.2.8 uint32_t [unpack_wds_GetDefaultProfile_t::pdptype](#)

8.402.2.9 uint32_t [unpack_wds_GetDefaultProfile_t::pridns](#)

8.402.2.10 uint16_t [unpack_wds_GetDefaultProfile_t::pridnsv6](#)

8.402.2.11 uint32_t [unpack_wds_GetDefaultProfile_t::secdns](#)

8.402.2.12 uint16_t [unpack_wds_GetDefaultProfile_t::secdnsv6](#)

8.402.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

8.402.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

8.403 unpack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [index](#)

8.403.1 Detailed Description

Parameters

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

8.403.2 Field Documentation

8.403.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

8.404 unpack_wds_GetDormancyState_t Struct Reference

Data Fields

- uint32_t [dormancyState](#)

8.404.1 Detailed Description

Parameters

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

8.404.2 Field Documentation

8.404.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

8.405 unpack_wds_GetLastMobileLError_t Struct Reference

Data Fields

- uint32_t [error](#)

8.405.1 Detailed Description

Parameters

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

8.405.2 Field Documentation

8.405.2.1 uint32_t unpack_wds_GetLastMobileLError_t::error

8.406 unpack_wds_GetMobileIP_t Struct Reference

Data Fields

- uint32_t [mipMode](#)

8.406.1 Detailed Description

Parameters

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

8.406.2 Field Documentation

8.406.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.407 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](#)

8.407.1 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

8.407.2 Field Documentation

8.407.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

- 8.407.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAState
- 8.407.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address
- 8.407.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled
- 8.407.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI
- 8.407.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HASState
- 8.407.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]
- 8.407.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize
- 8.407.2.9 uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA
- 8.407.2.10 uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling
- 8.407.2.11 uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA

8.408 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

8.408.1 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

8.408.2 Field Documentation

8.408.2.1 `uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount`

8.408.2.2 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount`

8.408.2.3 `uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall`

8.408.2.4 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors`

8.408.2.5 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows`

8.408.2.6 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses`

8.408.2.7 `uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount`

8.408.2.8 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount`

8.408.2.9 `uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall`

8.408.2.10 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors`

8.408.2.11 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows`

8.408.2.12 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses`

8.409 `unpack_wds_GetSessionDuration_t` Struct Reference

Data Fields

- `uint64_t` [callDuration](#)

8.409.1 Detailed Description

Parameters

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

8.409.2 Field Documentation

8.409.2.1 `uint64_t unpack_wds_GetSessionDuration_t::callDuration`

8.410 `unpack_wds_GetSessionState_t` Struct Reference

Data Fields

- `uint32_t` [connectionStatus](#)

8.410.1 Detailed Description

Parameters

connection-Status	state of the current packet data session
-----------------------------------	--

8.410.2 Field Documentation

8.410.2.1 `uint32_t unpack_wds_GetSessionState_t::connectionStatus`

8.411 unpack_wds_RMSetTransferStatistics_t Struct Reference

8.412 unpack_wds_SetMobileIPProfile_t Struct Reference

8.413 unpack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

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

8.413.1 Detailed Description

Parameters

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

8.413.2 Field Documentation

8.413.2.1 `PackCreateProfileOut*` `unpack_wds_SLQSCreateProfile_t::pCreateProfileOut`

8.413.2.2 `uint8_t*` `unpack_wds_SLQSCreateProfile_t::pProfileID`

8.413.2.3 `uint16_t` `unpack_wds_SLQSCreateProfile_t::Tlvresult`

8.414 unpack_wds_SLQSDestroyProfile_t Struct Reference

Data Fields

- `uint16_t` [extendedErrorCode](#)

8.414.1 Detailed Description

Parameters

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

8.414.2 Field Documentation

8.414.2.1 `uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode`

8.415 `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`

8.415.1 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

8.415.2 Field Documentation

8.415.2.1 `uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease`

8.415.2.2 `uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled`

8.415.2.3 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]

8.415.2.4 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen

8.415.2.5 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]

8.416 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

Data Fields

- uint8_t [prefNetwork](#)
- uint8_t [networkInfoLen](#)
- [currNetworkInfo](#) [currNetworkInfo](#) [255]

8.416.1 Detailed Description

Parameters

prefNetwork	preferred network
networkInfoLen	number of set of currNetworkInfo elements
currNetworkInfo	current network information.

8.416.2 Field Documentation

8.416.2.1 [currNetworkInfo](#) [unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo](#)[255]

8.416.2.2 uint8_t [unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen](#)

8.416.2.3 uint8_t [unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork](#)

8.417 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference

Data Fields

- uint8_t [dataBearerMask](#)
- [qmiWSDDataBearerTechnology](#) [curDataBearerTechnology](#)
- [qmiWSDDataBearerTechnology](#) [lastCallDataBearerTechnology](#)

8.417.1 Detailed Description

Parameters

dataBearerMask	bit mask indicates bearer info is for current and/or last call
curDataBearerTechnology	current data bearer technology value
lastCallDataBearerTechnology	last call data bearer technology value

8.417.2 Field Documentation

8.417.2.1 [qmiWSDDataBearerTechnology](#) [unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology](#)

8.417.2.2 `uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask`

8.417.2.3 `qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology`

8.418 `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](#)

8.418.1 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

8.418.2 Field Documentation

8.418.2.1 `uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason`

8.418.2.2 `dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate`

8.418.2.3 `connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus`

8.418.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`

8.418.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`

- 8.418.2.6 `uint8_t` `unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech`
- 8.418.2.7 `uint64_t` `unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt`
- 8.418.2.8 `uint64_t` `unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt`
- 8.418.2.9 `uint64_t` `unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive`
- 8.418.2.10 `uint64_t` `unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount`
- 8.418.2.11 `uint64_t` `unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount`

8.419 unpack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

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

8.419.1 Field Documentation

- 8.419.1.1 `UnPackGetProfileSettingOut`* `unpack_wds_SLQSGetProfileSettings_t::pProfileSettings`
- 8.419.1.2 `uint8_t` `unpack_wds_SLQSGetProfileSettings_t::ProfileType`
- 8.419.1.3 `uint16_t` `unpack_wds_SLQSGetProfileSettings_t::Tlvresult`

8.420 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](#)

8.420.1 Detailed Description

Parameters

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

8.420.2 Field Documentation

8.420.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

8.420.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

8.420.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

8.420.2.4 struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

8.420.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

8.420.2.6 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag

8.420.2.7 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference

- 8.420.2.8 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4
- 8.420.2.9 struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6AddrInfo
- 8.420.2.10 struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6GWAddrInfo
- 8.420.2.11 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu
- 8.420.2.12 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO
- 8.420.2.13 struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList
- 8.420.2.14 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPTType
- 8.420.2.15 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4
- 8.420.2.16 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]
- 8.420.2.17 struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID
- 8.420.2.18 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]
- 8.420.2.19 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4
- 8.420.2.20 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]
- 8.420.2.21 struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList
- 8.420.2.22 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4
- 8.420.2.23 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology
- 8.420.2.24 LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS
- 8.420.2.25 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]

8.421 unpack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

- uint16_t * [pExtErrorCode](#)

8.421.1 Detailed Description

Parameters

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

8.421.2 Field Documentation

- 8.421.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

8.422 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.422.1 Detailed Description

Parameters

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

8.422.2 Field Documentation

8.422.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

8.423 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](#)

8.423.1 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.

8.423.2 Field Documentation

8.423.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID

8.423.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status

8.423.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily

8.423.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd

8.423.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason

8.423.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName

8.423.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason

8.423.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

8.424 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]

8.424.1 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

8.424.2 Field Documentation

- 8.424.2.1 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail`
- 8.424.2.2 `wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]`
- 8.424.2.3 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail`
- 8.424.2.4 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail`
- 8.424.2.5 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology`
- 8.424.2.6 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail`
- 8.424.2.7 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus`
- 8.424.2.8 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail`
- 8.424.2.9 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus`
- 8.424.2.10 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen`
- 8.424.2.11 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork`
- 8.424.2.12 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask`
- 8.424.2.13 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes`
- 8.424.2.14 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts`
- 8.424.2.15 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask`
- 8.424.2.16 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes`
- 8.424.2.17 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts`
- 8.424.2.18 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail`

8.425 `unpack_wds_SLQSSGetDHCPv4ClientConfig_t` Struct Reference

Data Fields

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

8.425.1 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

8.425.2 Field Documentation

8.425.2.1 wdsDhcpv4HwConfig* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig

8.425.2.2 wdsDhcpv4OptionList* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList

8.426 unpack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

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

8.426.1 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>pVerboseFail-ReasonType</i>	<ul style="list-style-type: none"> • Parameter describing type of verbose failure reason • See qaGobiApiTableCallEndReasons.h for Call End Reason Type
<i>pVerboseFailure-Reason</i>	<ul style="list-style-type: none"> • Verbose reason explaining why call failed. Depends on verbFailReasonType parameter • See qaGobiApiTableCallEndReasons.h for Call End Reason

8.426.2 Field Documentation

8.426.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

8.426.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

8.426.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

8.426.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

8.427 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]

8.427.1 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

8.427.2 Field Documentation

8.427.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

8.427.2.2 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId

8.427.2.3 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.427.2.4 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address

8.427.2.5 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress

8.427.2.6 struct [ipv6AddressInfo](#) unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address

8.427.2.7 struct [ipv6AddressInfo](#) unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress

8.427.2.8 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address

- 8.427.2.9 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`
- 8.427.2.10 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`
- 8.427.2.11 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`
- 8.427.2.12 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`
- 8.427.2.13 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`
- 8.427.2.14 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`
- 8.427.2.15 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

8.428 UnPackGetProfileSettingOut Struct Reference

Data Fields

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

8.428.1 Field Documentation

- 8.428.1.1 **UnpackQmiProfileInfo** UnPackGetProfileSettingOut::curProfile
- 8.428.1.2 `uint16_t*` UnPackGetProfileSettingOut::pExtErrCode

8.429 unpackWdsProfileParam Union Reference

Data Fields

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

8.429.1 Field Documentation

- 8.429.1.1 **LibpackProfile3GPP** unpackWdsProfileParam::SlqsProfile3GPP
- 8.429.1.2 **LibpackProfile3GPP2** unpackWdsProfileParam::SlqsProfile3GPP2

8.430 wds_currNetworkInfo Struct Reference

Data Fields

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

8.430.1 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
---------------	--

8.430.2 Field Documentation

8.430.2.1 `uint8_t wds_currNetworkInfo::NetworkType`

8.430.2.2 `uint32_t wds_currNetworkInfo::RATMask`

8.430.2.3 `uint32_t wds_currNetworkInfo::SOMask`

8.431 wds_Domain Struct Reference

Data Fields

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

8.431.1 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)

8.431.2 Field Documentation

8.431.2.1 `uint16_t wds_Domain::domainLen`

8.431.2.2 `uint8_t wds_Domain::domainName[256]`

8.432 wds_DomainNameList Struct Reference

Data Fields

- `uint8_t numInstances`
- struct `wds_Domain domain` [10]

8.432.1 Detailed Description

This structure contains the DomainNameList Information

Parameters

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

8.432.2 Field Documentation

8.432.2.1 `struct wds_Domain wds_DomainNameList::domain[10]`

8.432.2.2 `uint8_t wds_DomainNameList::numInstances`

8.433 wds_GPRSQoS Struct Reference

Data Fields

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

8.433.1 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

8.433.2 Field Documentation

8.433.2.1 uint32_t wds_GPRSQoS::delayClass

8.433.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

8.433.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

8.433.2.4 uint32_t wds_GPRSQoS::precedenceClass

8.433.2.5 uint32_t wds_GPRSQoS::reliabilityClass

8.434 wds_IPV6AddressInfo Struct Reference

Data Fields

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

8.434.1 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.

8.434.2 Field Documentation

8.434.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

8.434.2.2 uint8_t wds_IPV6AddressInfo::IPv6PrefixLen

8.435 wds_IPV6GWAddressInfo Struct Reference

Data Fields

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

8.435.1 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.

8.435.2 Field Documentation

8.435.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

8.435.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

8.436 wds_PCSCFFQDNAddress Struct Reference

Data Fields

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

8.436.1 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)

8.436.2 Field Documentation

8.436.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

8.436.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

8.437 wds_PCSCFFQDNAddressList Struct Reference

Data Fields

- uint8_t [numInstances](#)
- struct [wds_PCSCFFQDNAddress](#) [pcsfFQDNAddress](#) [10]

8.437.1 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)

8.437.2 Field Documentation

8.437.2.1 uint8_t wds_PCSCFFQDNAddressList::numInstances

8.437.2.2 struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]

8.438 wds_PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- uint8_t [numInstances](#)
- uint32_t [pcscfIPv4Addr](#) [64]

8.438.1 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)

8.438.2 Field Documentation

8.438.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

8.438.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pcscfIPv4Addr[64]

8.439 wds_ProfileIdentifier Struct Reference

Data Fields

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

8.439.1 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

8.439.2 Field Documentation

8.439.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.439.2.2 uint8_t wds_ProfileIdentifier::profileType

8.440 wds_profileInfo Union Reference

Data Fields

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

8.440.1 Detailed Description

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

8.440.2 Field Documentation

8.440.2.1 [LibPackprofile_3GPP](#) wds_profileInfo::SlqsProfile3GPP

8.440.2.2 [LibPackprofile_3GPP2](#) wds_profileInfo::SlqsProfile3GPP2

8.441 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](#)

8.441.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> • 0x00 - Subscribed • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 0x04 - Background
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> • Maximum uplink bit rate in bits/sec
<i>maxDownlink- Bitrate</i>	<ul style="list-style-type: none"> • Maximum downlink bit rate in bits/sec
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> • Guaranteed uplink bit rate in bits/sec
<i>grntDownlink- Bitrate</i>	<ul style="list-style-type: none"> • Guaranteed downlink bit rate in bits/sec
<i>qosDelivery- Order</i>	<ul style="list-style-type: none"> - Qos delivery order • 0x00 - Subscribe • 0x01 - Delivery order on • 0x02 - Delivery order off
<i>maxSDUSize</i>	<ul style="list-style-type: none"> • Maximum SDU size
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> - SDU error ratio • Target value for fraction of SDUs lost or detected as erroneous. • 0x00 - Subscribe • 0x01 - $1 \cdot 10^{(-2)}$ • 0x02 - $7 \cdot 10^{(-3)}$ • 0x03 - $1 \cdot 10^{(-3)}$ • 0x04 - $1 \cdot 10^{(-4)}$ • 0x05 - $1 \cdot 10^{(-5)}$ • 0x06 - $1 \cdot 10^{(-6)}$ • 0x07 - $1 \cdot 10^{(-1)}$

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

8.441.2 Field Documentation

8.441.2.1 `uint8_t wds_UMTSMinQoS::deliveryErrSDU`

8.441.2.2 `uint32_t wds_UMTSMinQoS::grntDownlinkBitrate`

8.441.2.3 `uint32_t wds_UMTSMinQoS::grntUplinkBitrate`

8.441.2.4 `uint32_t wds_UMTSMinQoS::maxDownlinkBitrate`

8.441.2.5 `uint32_t wds_UMTSMinQoS::maxSDUSize`

8.441.2.6 `uint32_t wds_UMTSMinQoS::maxUplinkBitrate`

8.441.2.7 `uint8_t wds_UMTSMinQoS::qosDeliveryOrder`

8.441.2.8 `uint8_t wds_UMTSMInQoS::resBerRatio`

8.441.2.9 `uint8_t wds_UMTSMInQoS::sduErrorRatio`

8.441.2.10 `uint8_t wds_UMTSMInQoS::trafficClass`

8.441.2.11 `uint32_t wds_UMTSMInQoS::trafficPriority`

8.441.2.12 `uint32_t wds_UMTSMInQoS::transferDelay`

8.442 wdsDhcpv4HwConfig Struct Reference

Data Fields

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

8.442.1 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

8.442.2 Field Documentation

8.442.2.1 `uint8_t wdsDhcpv4HwConfig::chaddr`[16]

8.442.2.2 `uint8_t wdsDhcpv4HwConfig::chaddrLen`

8.442.2.3 `uint8_t wdsDhcpv4HwConfig::hwType`

8.443 wdsDhcpv4Option Struct Reference

Data Fields

- `uint8_t optCode`
- `uint8_t optValLen`
- `uint8_t optVal` [255]

8.443.1 Detailed Description

Parameters

<i>optCode</i>	Option code • 0 - 255
<i>optValLen</i>	Option value length • 0 - 255
<i>optVal</i>	Option Value

8.443.2 Field Documentation

8.443.2.1 `uint8_t wdsDhcpv4Option::optCode`8.443.2.2 `uint8_t wdsDhcpv4Option::optVal[255]`8.443.2.3 `uint8_t wdsDhcpv4Option::optValLen`

8.444 wdsDhcpv4OptionList Struct Reference

Data Fields

- `uint8_t numOpt`
- `wdsDhcpv4Option * pOptList`

8.444.1 Detailed Description

Parameters

<i>numOpt</i>	number of options • 0 - 255
<i>pOptList</i>	pointer to list of DHCP Options

8.444.2 Field Documentation

8.444.2.1 `uint8_t wdsDhcpv4OptionList::numOpt`8.444.2.2 `wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList`

8.445 wdsDhcpv4ProfileId Struct Reference

Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

8.445.1 Detailed Description

Parameters

<i>profileType</i>	profile type • 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)
------------------	---

8.445.2 Field Documentation

8.445.2.1 `uint8_t wdsDhcpv4ProfileId::profileId`

8.445.2.2 `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,
[eTMD](#) =0x18,
[eSWIOMA](#) =240,
[eSWILOC](#) =246 }
- enum [msgtype](#) {
[eREQ](#) =0,
[eRSP](#) =2,
[eIND](#) =4 }

Functions

- [uint16_t helper_get_xid](#) ([uint8_t](#) *qmi_resp)
- [const char *](#) [helper_get_resp_ctx](#) ([uint8_t](#) svc, [uint8_t](#) *pbuf, [uint32_t](#) len, [unpack_qmi_t](#) *pCtx)
- [unsigned](#) [unpack_result_code_only](#) ([uint8_t](#) *pMdmResp)
- [int](#) [helper_set_log_func](#) ([logger](#) func)
- [void](#) [libpack_log](#) ([uint8_t](#) lvl, [const char](#) *fmt,...)
- [int](#) [helper_set_log_lvl](#) ([uint8_t](#) lvl)
- [void](#) [fill_sdu_hdr](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf)
- [void](#) [fill_pack_ctx](#) ([pack_qmi_t](#) *pCtx, [uint8_t](#) *pReqBuf, [uint16_t](#) *pLen, [uint8_t](#) svc, [int](#) timeout)
- [char *](#) [get_version](#) ()
- [char *](#) [libpack_GetVersion](#) ()

Variables

- [logger](#) [glog](#)
- [uint8_t](#) [gloglvl](#)

9.2.1 Macro Definition Documentation

9.2.1.1 `#define DEAUULT_LOC_TIMEOUT_IN_SEC 2`

9.2.1.2 `#define MINREQBKLEN (2048)`

9.2.1.3 `#define MSGID_AND_LEN (4)`

9.2.1.4 `#define MSGID_DONT_CARE (0xffff)`

9.2.1.5 `#define SDK_VALIDATE_INPUT_PACK_PARAM(pCtx, pBuf, pLen)`

Value:

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
{ \
    libpack_log(eLOG_DEBUG, "[ pack] %s parameter NULL\n", __func__); \
    return eQCWWAN_ERR_INVALID_ARG; \
}
```

9.2.1.6 `#define SDU_HDR_LEN (3)`

9.2.1.7 `#define UNUSEDPARAM(x) (void)x`

9.2.2 Typedef Documentation

9.2.2.1 `typedef void(* logger)(uint8_t lvl, const char *buff)`

9.2.3 Enumeration Type Documentation

9.2.3.1 `enum eLOG_LEVEL`

log levels

Enumerator

eLOG_INFO
eLOG_DEBUG
eLOG_WARN
eLOG_FATAL

9.2.3.2 `enum eQMI_SVC`

qmi service

Enumerator

eCTL
eWDS
eDMS
eNAS
eQOS
eSMS
eUIM

eLOC
eTMD
eSWIOMA
eSWILOC

9.2.3.3 enum eTimeout

eTimeout

Enumerator

eTIMEOUT_2_S
eTIMEOUT_5_S
eTIMEOUT_8_S
eTIMEOUT_10_S
eTIMEOUT_20_S
eTIMEOUT_30_S
eTIMEOUT_60_S
eTIMEOUT_300_S
eTIMEOUT_DEFAULT

9.2.3.4 enum msgtype

qmi message type

Enumerator

eREQ
eRSP
eIND

9.2.4 Function Documentation

9.2.4.1 void fill_pack_ctx (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t svc, int timeout)

9.2.4.2 void fill_sdu_hdr (pack_qmi_t * pCtx, uint8_t * pReqBuf)

9.2.4.3 char* get_version ()

Returns

version string

9.2.4.4 const char* helper_get_resp_ctx (uint8_t svc, uint8_t * pbuf, uint32_t len, unpack_qmi_t * pCtx)

extract msgid/xid/type from modem reply

Parameters

in	svc	qmi service
in	pbuf	qmi response/indication
in	len	response/indication length
out	pCtx	unpacked context

Returns

qmi message string

9.2.4.5 `uint16_t helper_get_xid (uint8_t * qmi_resp)`

9.2.4.6 `int helper_set_log_func (logger func)`

set log function

9.2.4.7 `int helper_set_log_lvl (uint8_t lvl)`

set log level

9.2.4.8 `char* libpack_GetVersion ()`

Returns

version string

9.2.4.9 `void libpack_log (uint8_t lvl, const char * fmt, ...)`

9.2.4.10 `unsigned unpack_result_code_only (uint8_t * pMdmResp)`

common handler for unpacking response with TLV type 0x02 only

9.2.5 Variable Documentation

9.2.5.1 `logger glog`

9.2.5.2 `uint8_t gloglvl`

9.3 dms.h File Reference

Data Structures

- struct [unpack_dms_GetModelID_t](#)
- struct [unpack_dms_GetIMSI_t](#)
- struct [unpack_dms_GetFirmwareInfo_t](#)
- struct [unpack_dms_GetPower_t](#)
- struct [unpack_dms_GetSerialNumbers_t](#)
- struct [unpack_dms_GetHardwareRevision_t](#)
- struct [unpack_dms_SLQSGetBandCapability_t](#)
- struct [unpack_dms_GetDeviceCapabilities_t](#)
- struct [unpack_dms_GetFirmwareRevisions_t](#)
- struct [unpack_dms_GetFirmwareRevision_t](#)
- struct [unpack_dms_GetDeviceSerialNumbers_t](#)
- struct [unpack_dms_GetPRLVersion_t](#)
- struct [unpack_dms_GetNetworkTime_t](#)
- struct [unpack_dms_GetVoiceNumber_t](#)
- struct [unpack_dms_GetDeviceHardwareRev_t](#)
- struct [unpack_dms_GetFSN_t](#)

- struct [unpack_dms_GetDeviceCap_t](#)
- struct [pack_dms_SetPower_t](#)
- struct [unpack_dms_SetPower_t](#)
- struct [unpack_dms_GetBandCapability_t](#)
- struct [unpack_dms_GetUSBComp_t](#)
- struct [pack_dms_SetUSBComp_t](#)
- struct [unpack_dms_SetUSBComp_t](#)
- struct [pack_dms_SetCustFeature_t](#)
- struct [unpack_dms_SetCustFeature_t](#)
- struct [unpack_dms_GetCustFeature_t](#)
- struct [unpack_dms_SetFirmwarePreference_t](#)
- struct [unpack_dms_GetCrashAction_t](#)
- struct [pack_dms_SetCrashAction_t](#)
- struct [unpack_dms_SetCrashAction_t](#)
- struct [unpack_dms_GetDeviceMfr_t](#)
- struct [pack_dms_SetEventReport_t](#)
- struct [unpack_dms_SetEventReport_t](#)
- struct [dms_OperatingModeTlv](#)
- struct [dms_ActivationStatusTlv](#)
- struct [unpack_dms_SetEventReport_ind_t](#)
- struct [pack_dms_UIMGetICCID_t](#)
- struct [unpack_dms_UIMGetICCID_t](#)
- struct [pack_dms_SetCustFeaturesV2_t](#)
- struct [unpack_dms_SetCustFeaturesV2_t](#)
- struct [pack_dms_GetCustFeaturesV2_t](#)
- struct [DMSgetCustomInput](#)
- struct [DMScustSettingInfo](#)
- struct [DMScustSettingList](#)
- struct [DMSgetCustomFeatureV2](#)
- struct [unpack_dms_GetCustFeaturesV2_t](#)
- struct [unpack_dms_GetActivationState_t](#)
- struct [image_info_t](#)
- struct [unpack_dms_SLQSSwiGetFirmwareCurr_t](#)
- struct [pack_dms_SLQSSwiSetDyingGaspCfg_t](#)
- struct [unpack_dms_SLQSSwiSetDyingGaspCfg_t](#)
- struct [unpack_dms_SLQSSwiClearDyingGaspStatistics_t](#)
- struct [packgetDyingGaspStatistics](#)
- struct [unpack_dms_SLQSSwiGetDyingGaspStatistics_t](#)
- struct [packgetDyingGaspCfg](#)
- struct [unpack_dms_SLQSSwiGetDyingGaspCfg_t](#)
- struct [unpack_dms_SLQSDmsSwiGetResetInfo_t](#)
- struct [unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t](#)
- struct [pack_dms_SLQSDmsSwiIndicationRegister_t](#)
- struct [unpack_dms_SLQSDmsSwiIndicationRegister_t](#)
- struct [unpack_dms_SLQSSwiGetFwUpdateStatus_t](#)

Macros

- #define [DMS_UINT8_MAX_STRING_SZ](#) 255
- #define [DMS_MAX_CUST_ID_LEN](#) 64
- #define [DMS_MAX_CUST_VALUE_LEN](#) 8
- #define [DMS_IMGDETAILS_LEN](#) 16
- #define [DMS_MAX_FWUPDATE_LOG_STR_SZ](#) 255
- #define [DMS_MAX_FWUPDATE_REF_STR_SZ](#) 15
- #define [DMS_SLQSFWINFO_MODELID_SZ](#) 20
- #define [DMS_SLQSFWINFO_BOOTVERSION_SZ](#) 85
- #define [DMS_SLQSFWINFO_APPVERSION_SZ](#) 85
- #define [DMS_SLQSFWINFO_SKU_SZ](#) 15
- #define [DMS_SLQSFWINFO_PACKAGEID_SZ](#) 85
- #define [DMS_SLQSFWINFO_CARRIER_SZ](#) 20
- #define [DMS_SLQSFWINFO_PRIVERSION_SZ](#) 16
- #define [DMS_SLQSFWINFO_CUR_CARR_NAME](#) 17
- #define [DMS_SLQSFWINFO_CUR_CARR_REV](#) 13
- #define [MAX_BUILD_ID_LEN](#) 255
- #define [UNIQUE_ID_LEN](#) 16
- #define [SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH](#) 160
- #define [SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH](#) 20
- #define [DMS_PM_ONLINE](#) 0x00 /* Online */
- #define [DMS_PM_LOW](#) 0x01 /* Low Power */
- #define [DMS_PM_FACTORY](#) 0x02 /* Factory Test Mode */
- #define [DMS_PM_OFFLINE](#) 0x03 /* Offline */
- #define [DMS_PM_RESET](#) 0x04 /* Reset */
- #define [DMS_PM_SHUT_DOWN](#) 0x05 /* Shut Down */
- #define [DMS_PM_PERSISTENT_LOW](#) 0x06 /* Persistent Low Power */
- #define [DMS_SET_REPORT_ENABLE](#) 1
- #define [DMS_SET_REPORT_DISABLE](#) 0
- #define [DMS_SWI_SET_IND_ENABLE](#) 1
- #define [DMS_SWI_SET_IND_DISABLE](#) 0

Functions

- int [pack_dms_GetIMSI](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetIMSI](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetIMSI_t](#) *pOutput)
- int [pack_dms_GetModelID](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetModelID](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetModelID_t](#) *pOutput)
- int [pack_dms_GetFirmwareInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetFirmwareInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetFirmwareInfo_t](#) *pOutput)
- int [pack_dms_GetPower](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetPower](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetPower_t](#) *pOutput)
- int [pack_dms_GetSerialNumbers](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetSerialNumbers](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetSerialNumbers_t](#) *pOutput)
- int [pack_dms_GetHardwareRevision](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetHardwareRevision](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetHardwareRevision_t](#) *pOutput)
- int [pack_dms_SLQSGetBandCapability](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_SLQSGetBandCapability](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSGetBandCapability_t](#) *pOutput)
- int [pack_dms_GetDeviceCapabilities](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)

- int [unpack_dms_GetDeviceCapabilities](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetDeviceCapabilities_t](#) *pOutput)
- int [pack_dms_GetFirmwareRevisions](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetFirmwareRevisions](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetFirmwareRevisions_t](#) *pOutput)
- int [pack_dms_GetFirmwareRevision](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetFirmwareRevision](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetFirmwareRevision_t](#) *pOutput)
- int [pack_dms_GetDeviceSerialNumbers](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetDeviceSerialNumbers](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetDeviceSerialNumbers_t](#) *pOutput)
- int [pack_dms_GetPRLVersion](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetPRLVersion](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetPRLVersion_t](#) *pOutput)
- int [pack_dms_GetNetworkTime](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetNetworkTime](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetNetworkTime_t](#) *pOutput)
- int [pack_dms_GetVoiceNumber](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetVoiceNumber](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetVoiceNumber_t](#) *pOutput)
- int [pack_dms_GetDeviceHardwareRev](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetDeviceHardwareRev](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetDeviceHardwareRev_t](#) *pOutput)
- int [pack_dms_GetFSN](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetFSN](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetFSN_t](#) *pOutput)
- int [pack_dms_GetDeviceCap](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetDeviceCap](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetDeviceCap_t](#) *pOutput)
- int [pack_dms_SetPower](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetPower_t](#) *reqArg)
- int [unpack_dms_SetPower](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetPower_t](#) *pOutput)
- int [pack_dms_GetBandCapability](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetBandCapability](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetBandCapability_t](#) *pOutput)
- int [pack_dms_GetUSBComp](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetUSBComp](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetUSBComp_t](#) *pOutput)
- int [pack_dms_SetUSBComp](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetUSBComp_t](#) *reqArg)
- int [unpack_dms_SetUSBComp](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetUSBComp_t](#) *pOutput)
- int [pack_dms_SetCustFeature](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetCustFeature_t](#) *reqArg)
- int [unpack_dms_SetCustFeature](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetCustFeature_t](#) *pOutput)
- int [pack_dms_GetCustFeature](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetCustFeature](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetCustFeature_t](#) *pOutput)
- int [pack_dms_SetFirmwarePreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_SetFirmwarePreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetFirmwarePreference_t](#) *pOutput)
- int [pack_dms_GetCrashAction](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_dms_GetCrashAction](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetCrashAction_t](#) *pOutput)
- int [pack_dms_SetCrashAction](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetCrashAction_t](#) reqArg)
- int [unpack_dms_SetCrashAction](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetCrashAction_t](#) *pOutput)
- int [pack_dms_GetDeviceMfr](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)

- int [unpack_dms_GetDeviceMfr](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetDeviceMfr_t](#) *pOutput)
- int [pack_dms_SetEventReport](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetEventReport_t](#) *reqArg)
- int [unpack_dms_SetEventReport](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetEventReport_t](#) *pOutput)
- int [unpack_dms_SetEventReport_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetEventReport_ind_t](#) *pOutput)
- int [pack_dms_UIMGetICCID](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_UIMGetICCID_t](#) *reqArg)
- int [unpack_dms_UIMGetICCID](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_UIMGetICCID_t](#) *pOutput)
- int [pack_dms_SetCustFeaturesV2](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SetCustFeaturesV2_t](#) *reqArg)
- int [unpack_dms_SetCustFeaturesV2](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SetCustFeaturesV2_t](#) *pOutput)
- int [pack_dms_GetCustFeaturesV2](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_GetCustFeaturesV2_t](#) *reqArg)
- int [unpack_dms_GetCustFeaturesV2](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetCustFeaturesV2_t](#) *pOutput)
- int [pack_dms_GetActivationState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_GetActivationState](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetActivationState_t](#) *pOutput)
- int [pack_dms_SLQSSwiGetFirmwareCurr](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSSwiGetFirmwareCurr](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiGetFirmwareCurr_t](#) *pOutput)
- int [pack_dms_SLQSSwiSetDyingGaspCfg](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SLQSSwiSetDyingGaspCfg_t](#) *reqArg)
- int [unpack_dms_SLQSSwiSetDyingGaspCfg](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiSetDyingGaspCfg_t](#) *pOutput)
- int [pack_dms_SLQSSwiClearDyingGaspStatistics](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSSwiClearDyingGaspStatistics](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiClearDyingGaspStatistics_t](#) *pOutput)
- int [pack_dms_SLQSSwiGetDyingGaspStatistics](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSSwiGetDyingGaspStatistics](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiGetDyingGaspStatistics_t](#) *pOutput)
- int [pack_dms_SLQSSwiGetDyingGaspCfg](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSSwiGetDyingGaspCfg](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiGetDyingGaspCfg_t](#) *pOutput)
- int [pack_dms_SLQSDmsSwiGetResetInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSDmsSwiGetResetInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSDmsSwiGetResetInfo_t](#) *pOutput)
- int [unpack_dms_SLQSDmsSwiGetResetInfo_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSDmsSwiGetResetInfo_ind_t](#) *pOutput)
- int [pack_dms_SLQSDmsSwiIndicationRegister](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_dms_SLQSDmsSwiIndicationRegister_t](#) *reqArg)
- int [unpack_dms_SLQSDmsSwiIndicationRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSDmsSwiIndicationRegister_t](#) *pOutput)
- int [pack_dms_SLQSSwiGetFwUpdateStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_SLQSSwiGetFwUpdateStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_SLQSSwiGetFwUpdateStatus_t](#) *pOutput)

9.3.1 Macro Definition Documentation

9.3.1.1 `#define DMS_IMGDETAILS_LEN 16`

9.3.1.2 `#define DMS_MAX_CUST_ID_LEN 64`

```
9.3.1.3  #define DMS_MAX_CUST_VALUE_LEN 8

9.3.1.4  #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255

9.3.1.5  #define DMS_MAX_FWUPDATE_REF_STR_SZ 15

9.3.1.6  #define DMS_PM_FACTORY 0x02 /* Factory Test Mode */

9.3.1.7  #define DMS_PM_LOW 0x01 /* Low Power */

9.3.1.8  #define DMS_PM_OFFLINE 0x03 /* Offline */

9.3.1.9  #define DMS_PM_ONLINE 0x00 /* Online */

9.3.1.10 #define DMS_PM_PERSISTENT_LOW 0x06 /* Persistent Low Power */

9.3.1.11 #define DMS_PM_RESET 0x04 /* Reset */

9.3.1.12 #define DMS_PM_SHUT_DOWN 0x05 /* Shut Down */

9.3.1.13 #define DMS_SET_REPORT_DISABLE 0

9.3.1.14 #define DMS_SET_REPORT_ENABLE 1

9.3.1.15 #define DMS_SLQSFWINFO_APPVERSION_SZ 85

9.3.1.16 #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85

9.3.1.17 #define DMS_SLQSFWINFO_CARRIER_SZ 20

9.3.1.18 #define DMS_SLQSFWINFO_CUR_CARR_NAME 17

9.3.1.19 #define DMS_SLQSFWINFO_CUR_CARR_REV 13

9.3.1.20 #define DMS_SLQSFWINFO_MODELID_SZ 20

9.3.1.21 #define DMS_SLQSFWINFO_PACKAGEID_SZ 85

9.3.1.22 #define DMS_SLQSFWINFO_PRIVERSION_SZ 16

9.3.1.23 #define DMS_SLQSFWINFO_SKU_SZ 15

9.3.1.24 #define DMS_SWI_SET_IND_DISABLE 0

9.3.1.25 #define DMS_SWI_SET_IND_ENABLE 1

9.3.1.26 #define DMS_UINT8_MAX_STRING_SZ 255

9.3.1.27 #define MAX_BUILD_ID_LEN 255

9.3.1.28 #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160

9.3.1.29 #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

9.3.1.30 #define UNIQUE_ID_LEN 16
```

9.3.2 Function Documentation

9.3.2.1 `int pack_dms_GetActivationState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Get Activation State pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.2 `int pack_dms_GetBandCapability (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Band Capability pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.3 `int pack_dms_GetCrashAction (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Crash Action pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.4 `int pack_dms_GetCustFeature (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use [pack_dms_GetCustFeatures-V2\(\)](#) instead for EM74xx/MC74xx.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.5 `int pack_dms_GetCustFeaturesV2 (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_GetCustFeaturesV2_t * reqArg)`

9.3.2.6 `int pack_dms_GetDeviceCap (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Device Capabilities pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.7 `int pack_dms_GetDeviceCapabilities (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get device capability pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.8 `int pack_dms_GetDeviceHardwareRev (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Hardware Revision pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.9 `int pack_dms_GetDeviceMfr (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Manufacture pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.10 `int pack_dms_GetDeviceSerialNumbers (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get Device Serial Number pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.11 int pack_dms_GetFirmwareInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get firmware info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.12 int pack_dms_GetFirmwareRevision (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get Firmware Revision pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.13 int pack_dms_GetFirmwareRevisions (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get Firmware Revisions pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.14 `int pack_dms_GetFSN (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get FSN pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.15 `int pack_dms_GetHardwareRevision (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get hardware revision pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.16 `int pack_dms_GetIMSI (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get IMSI pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx-xx and all EM74xx firmware versions. Please use [pack_uim_ReadTransparent\(\)](#)(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.17 int pack_dms_GetModelID (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

get model id pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.18 int pack_dms_GetNetworkTime (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Network Time pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.19 `int pack_dms_GetPower (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get power pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.20 `int pack_dms_GetPRLVersion (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get PRL Versions pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.21 `int pack_dms_GetSerialNumbers (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get serial numbers pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.22 `int pack_dms_GetUSBComp (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get USB Comp pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.23 `int pack_dms_GetVoiceNumber (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Get Voice Number pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.24 `int pack_dms_SetCrashAction (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetCrashAction_t reqArg)`

Set Crash Action pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Set Crash Action unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.25 int pack_dms_SetCustFeature (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetCustFeature_t * reqArg)

Set Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use [pack_dms_SetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.26 int pack_dms_SetCustFeaturesV2 (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetCustFeaturesV2_t * reqArg)

Set Cust Features pack.

Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.27 `int pack_dms_SetEventReport (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetEventReport_t * reqArg)`

Set Event Report pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.28 `int pack_dms_SetFirmwarePreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

Set Firmware Preference pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.29 `int pack_dms_SetPower (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetPower_t * reqArg)`

Set Power pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.30 int pack_dms_SetUSBComp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_dms_SetUSBComp_t * *reqArg*)

Set USB Comp pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.31 int pack_dms_SLQSDmsSwtGetResetInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To get reset info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.32 `int pack_dms_SLQSDmsSwilIndicationRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSDmsSwilIndicationRegister_t * reqArg)`

Set the registration state for different indication pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.33 `int pack_dms_SLQSGetBandCapability (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg)`

get band capability pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.34 `int pack_dms_SLQSSwiClearDyingGaspStatistics (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Clear Dying GASP Statistics pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.35 int pack_dms_SLQSSwiGetDyingGaspCfg (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Get Dying GASP Config pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.36 int pack_dms_SLQSSwiGetDyingGaspStatistics (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Get Dying GASP Statistics pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.37 int pack_dms_SLQSSwiGetFirmwareCurr (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

get currently active image pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.38 `int pack_dms_SLQSSwiGetFwUpdateStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

To get Firmware Update status pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.39 `int pack_dms_SLQSSwiSetDyingGaspCfg (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiSetDyingGaspCfg_t * reqArg)`

Set Dying GASP Config pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.40 `int pack_dms_UIMGetICCID (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMGetICCID_t * reqArg)`

Packs the UIMGetICCID response message to a user-provided response structure. This API is deprecated on MC73xx/EM73xx modules. Since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use [pack_uim_ReadTransparent\(\)](#)(EF ID: 3F00 2FE2) instead for new firmware versions and new modules.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.41 int unpack_dms_GetActivationState (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetActivationState_t * *pOutput*)

Get Activation State unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.42 int unpack_dms_GetBandCapability (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetBandCapability_t * *pOutput*)

Get Band Capabilities unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.43 `int unpack_dms_GetCrashAction (uint8_t * pResp, uint16_t respLen, unpack_dms_GetCrashAction_t * pOutput)`

Get Crash Action unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.44 `int unpack_dms_GetCustFeature (uint8_t * pResp, uint16_t respLen, unpack_dms_GetCustFeature_t * pOutput)`

Get Custom Feature unpack. This API is deprecated for EM74xx/MC74xx, please use [unpack_dms_GetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.45 `int unpack_dms_GetCustFeaturesV2 (uint8_t * pResp, uint16_t respLen, unpack_dms_GetCustFeaturesV2_t * pOutput)`

9.3.2.46 `int unpack_dms_GetDeviceCap (uint8_t * pResp, uint16_t respLen, unpack_dms_GetDeviceCap_t * pOutput)`

Get Device Capabilities unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.47 int unpack_dms_GetDeviceCapabilities (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceCapabilities_t * *pOutput*)

get device capability unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.48 int unpack_dms_GetDeviceHardwareRev (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceHardwareRev_t * *pOutput*)

Get Hardware Revision unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.49 int unpack_dms_GetDeviceMfr (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceMfr_t * *pOutput*)

Get Manufacture unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.50 int unpack_dms_GetDeviceSerialNumbers (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_GetDeviceSerialNumbers_t * *pOutput*)

get Device Serial Number unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.51 int unpack_dms_GetFirmwareInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmwareInfo_t *
pOutput)

get firmware info unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.52 int unpack_dms_GetFirmwareRevision (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetFirmware-
Revision_t * *pOutput*)

get Firmware Revision unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.53 `int unpack_dms_GetFirmwareRevisions (uint8_t * pResp, uint16_t respLen, unpack_dms_GetFirmwareRevisions_t * pOutput)`

get Firmware Revisions unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.54 `int unpack_dms_GetFSN (uint8_t * pResp, uint16_t respLen, unpack_dms_GetFSN_t * pOutput)`

Get FSN unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.55 `int unpack_dms_GetHardwareRevision (uint8_t * pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t * pOutput)`

get hardware revision unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.56 int unpack_dms_GetIMSI (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetIMSI_t * *pOutput*)

get IMSI unpack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx-xx_xx and all EM74xx firmware versions. Please use [unpack_uim_ReadTransparent\(\)](#) (EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.57 int unpack_dms_GetModelID (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetModelID_t * *pOutput*)

get model id unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.58 int unpack_dms_GetNetworkTime (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetNetworkTime_t * *pOutput*)

Get Network Time unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.59 int unpack_dms_GetPower (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPower_t * *pOutput*)

get power unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.60 int unpack_dms_GetPRLVersion (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetPRLVersion_t * *pOutput*)

Get PRL Versions unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.61 int unpack_dms_GetSerialNumbers (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetSerialNumbers_t * *pOutput*)

get serial numbers unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.62 `int unpack_dms_GetUSBComp (uint8_t * pResp, uint16_t respLen, unpack_dms_GetUSBComp_t * pOutput)`

Get USB Comp unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.63 `int unpack_dms_GetVoiceNumber (uint8_t * pResp, uint16_t respLen, unpack_dms_GetVoiceNumber_t * pOutput)`

Get Voice Number unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.64 `int unpack_dms_SetCrashAction (uint8_t * pResp, uint16_t respLen, unpack_dms_SetCrashAction_t * pOutput)`

Set Crash Action unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response. Not used

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.65 `int unpack_dms_SetCustFeature (uint8_t * pResp, uint16_t respLen, unpack_dms_SetCustFeature_t * pOutput)`

Set Custom Feature unpack. This API is deprecated for EM74xx/MC74xx, please use [unpack_dms_SetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.66 `int unpack_dms_SetCustFeaturesV2 (uint8_t * pResp, uint16_t respLen, unpack_dms_SetCustFeaturesV2_t * pOutput)`

Set Cust features unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.67 `int unpack_dms_SetEventReport (uint8_t * pResp, uint16_t respLen, unpack_dms_SetEventReport_t * pOutput)`

Set Event Report unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.68 int unpack_dms_SetEventReport_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetEventReport_ind_t * *pOutput*)

Event Report Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.69 int unpack_dms_SetFirmwarePreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetFirmwarePreference_t * *pOutput*)

Set Firmware Preference unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.70 int unpack_dms_SetPower (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetPower_t * *pOutput*)

Set Power unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.71 int unpack_dms_SetUSBComp (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetUSBComp_t * *pOutput*)

Set USB Comp unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.72 int unpack_dms_SLQSDmsSwiGetResetInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSDmsSwiGetResetInfo_t * *pOutput*)

To get reset info unpack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.73 int unpack_dms_SLQSDmsSwiGetResetInfo_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t * *pOutput*)

DMS reset info Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.74 `int unpack_dms_SLQSDmsSwilIndicationRegister (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwilIndicationRegister_t * pOutput)`

Set the registration state for different indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.75 `int unpack_dms_SLQSGetBandCapability (uint8_t * pResp, uint16_t respLen, unpack_dms_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

9.3.2.76 `int unpack_dms_SLQSSwiClearDyingGaspStatistics (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiClearDyingGaspStatistics_t * pOutput)`

Clear Dying GASP Statistics unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.77 `int unpack_dms_SLQSSwiGetDyingGaspCfg (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspCfg_t * pOutput)`

Get Dying GASP Config unpack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.78 `int unpack_dms_SLQSSwiGetDyingGaspStatistics (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspStatistics_t * pOutput)`

Get Dying GASP Statistics unpack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.79 int unpack_dms_SLQSSwiGetFirmwareCurr (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_SLQSSwiGetFirmwareCurr_t * *pOutput*)

get currently active image unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.80 int unpack_dms_SLQSSwiGetFwUpdateStatus (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_SLQSSwiGetFwUpdateStatus_t * *pOutput*)

To get Firmware Update status unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.81 int unpack_dms_SLQSSwiSetDyingGaspCfg (uint8_t * *pResp*, uint16_t *respLen*,
unpack_dms_SLQSSwiSetDyingGaspCfg_t * *pOutput*)

Set Dying GASP Config unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.3.2.82 int unpack_dms_UIMGetICCID (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMGetICCID_t * *pOutput*)

Unpacks the UIMGetICCID response message to a user-provided response structure.

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.4 fms.h File Reference

Data Structures

- struct [CarrierImage_t](#)
- struct [pack_fms_GetImagesPreference_t](#)
- struct [FMSImageElement](#)
- struct [FMSPrefImageList](#)
- struct [unpack_fms_GetImagesPreference_t](#)
- struct [pack_fms_GetStoredImages_t](#)
- struct [FMSImageIdElement](#)
- struct [FMSImageIdEntries](#)
- struct [FMSImageList](#)
- struct [unpack_fms_GetStoredImages_t](#)
- struct [pack_fms_SetImagesPreference_t](#)
- struct [unpack_fms_SetImagesPreference_t](#)

Macros

- #define [FMS_GOB_I_MBN_IMG_ID_STR_LEN](#) 16
- #define [FMS_GOB_I_MBN_BUILD_ID_STR_LEN](#) 100
- #define [FMS_GOB_I_LISTENTRIES_MAX](#) 2
- #define [FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE](#) 255
- #define [FMS_MAX_IMAGE_ID_ELEMENT](#) 50
- #define [FMS_IMAGE_ID_MAX_ENTRIES](#) 2
- #define [FMS_FW_PRI_BUILD_MATCH_LEN](#) 11
- #define [FMS_IMAGE_ID_IMG_ID_LEN](#) 16
- #define [FMS_IMAGE_ID_BUILD_ID_LEN](#) 32
- #define [FMS_IMAGE_ID_PRI_IMGTYPE](#) 0x01

Functions

- int [pack_fms_GetImagesPreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_fms_GetImagesPreference_t](#) *reqArg)
- int [unpack_fms_GetImagesPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_fms_GetImagesPreference_t](#) *pOutput)
- int [pack_fms_GetStoredImages](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_fms_GetStoredImages_t](#) *reqArg)
- int [unpack_fms_GetStoredImages](#) (uint8_t *pResp, uint16_t respLen, [unpack_fms_GetStoredImages_t](#) *pOutput)
- int [pack_fms_SetImagesPreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_fms_SetImagesPreference_t](#) *reqArg)
- int [unpack_fms_SetImagesPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_fms_SetImagesPreference_t](#) *pOutput)
- uint32_t [GetValidFwPriCombinations](#) ([FMSImageList](#) *pStoredImageList, uint32_t *pValidCombinationSize, [CarrierImage_t](#) *pValidCombinations)

9.4.1 Macro Definition Documentation

9.4.1.1 `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`

9.4.1.2 `#define FMS_GOBI_LISTENTRIES_MAX 2`

9.4.1.3 `#define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100`

9.4.1.4 `#define FMS_GOBI_MBN_IMG_ID_STR_LEN 16`

9.4.1.5 `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`

9.4.1.6 `#define FMS_IMAGE_ID_IMG_ID_LEN 16`

9.4.1.7 `#define FMS_IMAGE_ID_MAX_ENTRIES 2`

9.4.1.8 `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

9.4.1.9 `#define FMS_MAX_IMAGE_ID_ELEMENT 50`

9.4.1.10 `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`

9.4.2 Function Documentation

9.4.2.1 `uint32_t GetValidFwPriCombinations (FMSImageList * pStoredImageList, uint32_t * pValidCombinationSize, CarrierImage_t * pValidCombinations)`

This API distills valid Firmware/PRI combinations from `GetStoredImages` result

Parameters

in	<i>pStoredImageList</i>	<ul style="list-style-type: none"> • image list returned from <code>GetStoredImages</code> • See FMSImageList
in, out	<i>pValidCombinationSize</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](#)
- struct [unpack_loc_SetExtPowerConfig_Ind_t](#)
- struct [unpack_loc_SLQSLOCGetBestAvailPos_t](#)
- struct [pack_loc_SLQSLOCGetBestAvailPos_t](#)
- struct [unpack_loc_BestAvailPos_Ind_t](#)

Macros

- #define [LOC_UINT8_MAX_STRING_SZ](#) 255
- #define [LOCEVENTMASKPOSITIONREPORT](#) 0x00000001
- #define [LOCEVENTMASKGNSSSVINFO](#) 0x00000002
- #define [LOCEVENTMASKNMEA](#) 0x00000004
- #define [LOCEVENTMASKNINOTIFYVERIFYREQ](#) 0x00000008
- #define [LOCEVENTMASKINJECTTIMERREQ](#) 0x00000010
- #define [LOCEVENTMASKINJECTPREDICTEDORBITSREQ](#) 0x00000020
- #define [LOCEVENTMASKINJECTPOSITIONREQ](#) 0x00000040
- #define [LOCEVENTMASKENGINESTATE](#) 0x00000080
- #define [LOCEVENTMASKFIXSESSIONSTATE](#) 0x00000100
- #define [LOCEVENTMASKWIFIREQ](#) 0x00000200
- #define [LOCEVENTMASKSENSORSTREAMINGREADYSTATUS](#) 0x00000400
- #define [LOCEVENTMASKTIMESYNCREQ](#) 0x00000800
- #define [LOCEVENTMASKSETSPISTREAMINGREPORT](#) 0x00001000
- #define [LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ](#) 0x00002000
- #define [LOCEVENTMASKNIGEOFENCENOTIFICATION](#) 0x00004000
- #define [LOCEVENTMASKGEOFENCEGENALERT](#) 0x00008000
- #define [LOCEVENTMASKGEOFENCEBREACHNOTIFICATION](#) 0x00010000
- #define [LOCEVENTMASKPEDOMETERCONTROL](#) 0x00020000
- #define [LOCEVENTMASKMOTIONDATACONTROL](#) 0x00040000
- #define [LOCEVENTMASKBATCHFULLNOTIFICATION](#) 0x00080000
- #define [LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT](#) 0x00100000
- #define [LOCEVENTMASKINJECTWIFIAPDATAREQ](#) 0x00200000
- #define [LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION](#) 0x00400000
- #define [LOCEVENTMASKVEHICLEDATAREADYSTATUS](#) 0x00800000
- #define [LOCEVENTMASKGNSSMEASUREMENTREPORT](#) 0x01000000
- #define [LOCEVENTMASKINVALIDVALUE](#) 0xFFFFFFFF

Enumerations

- enum {
[eQMI_LOC_SESS_STATUS_SUCCESS](#) =0,
[eQMI_LOC_SESS_STATUS_IN_PROGRESS](#) =1,
[eQMI_LOC_SESS_STATUS_FAILURE](#) =2,
[eQMI_LOC_SESS_STATUS_TIMEOUT](#) =3 }

Functions

- int [pack_loc_EventRegister](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_EventRegister_t](#) *reqArg)
- int [unpack_loc_EventRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_EventRegister_t](#) *pOutput)
- int [pack_loc_SetExtPowerState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_SetExtPowerState_t](#) *reqArg)
- int [unpack_loc_SetExtPowerState](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_SetExtPowerState_t](#) *pOutput)
- int [pack_loc_Start](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_Start_t](#) *reqArg)
- int [unpack_loc_Start](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_Start_t](#) *pOutput)
- int [pack_loc_Stop](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_Stop_t](#) *reqArg)
- int [unpack_loc_Stop](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_Stop_t](#) *pOutput)
- int [pack_loc_SetOperationMode](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_SetOperationMode_t](#) *reqArg)
- int [unpack_loc_SetOperationMode](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_SetOperationMode_t](#) *pOutput)
- int [pack_loc_DeleteAssistData](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_Delete_Assist_Data_t](#) *reqArg)
- int [unpack_loc_DeleteAssistData](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_Delete_Assist_Data_t](#) *pOutput)
- int [unpack_loc_PositionRpt_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_PositionRpt_Ind_t](#) *pOutput)
- int [unpack_loc_EngineState_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_EngineState_Ind_t](#) *pOutput)
- int [unpack_loc_SetExtPowerConfig_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_SetExtPowerConfig_Ind_t](#) *pOutput)
- int [pack_loc_SLQSLOCGetBestAvailPos](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_loc_SLQSLOCGetBestAvailPos_t](#) *reqArg)
- int [unpack_loc_SLQSLOCGetBestAvailPos](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_SLQSLOCGetBestAvailPos_t](#) *pOutput)
- int [unpack_loc_BestAvailPos_Ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_loc_BestAvailPos_Ind_t](#) *pOutput)

9.5.1 Macro Definition Documentation

9.5.1.1 `#define LOC_UINT8_MAX_STRING_SZ 255`

9.5.1.2 `#define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000`

The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.

9.5.1.3 `#define LOCEVENTMASKENGINESTATE 0x00000080`

The control point must enable this mask to receive engine state report event indications.

9.5.1.4 `#define LOCEVENTMASKFIXSESSIONSTATE 0x00000100`

The control point must enable this mask to receive fix session status report event indications.

9.5.1.5 `#define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000`

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

9.5.1.6 #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.

9.5.1.7 #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000

The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, for example, if GPS is turned off or if the network is unavailable.

9.5.1.8 #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000

The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG(Not yet supported).

9.5.1.9 #define LOCEVENTMASKGNSSSVINFO 0x00000002

The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate.

9.5.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040

The control point must enable this mask to receive position injection request event indications.

9.5.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020

The control point must enable this mask to receive predicted orbits request event indications.

9.5.1.12 #define LOCEVENTMASKINJECTTIMERREQ 0x00000010

The control point must enable this mask to receive time injection request event indications.

9.5.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000

The control point must enable this mask to receive Wi-Fi Access Point (AP) data inject request event indications.

9.5.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF

Invalid Event Mask

9.5.1.15 #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000

The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.

9.5.1.16 #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000

The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.

9.5.1.17 #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000

The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.

9.5.1.18 #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000

The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.

9.5.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008

The control point must enable this mask to receive NI Notify/Verify request event indications.

9.5.1.20 #define LOCEVENTMASKNMEA 0x00000004

The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

9.5.1.21 #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000

The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

9.5.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001

The control point must enable this mask to receive position report event indications.

9.5.1.23 #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.).

9.5.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000

The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications.

9.5.1.25 #define LOCEVENTMASKTIMESYNCREQ 0x00000800

The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

9.5.1.26 #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).

9.5.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200

The control point must enable this mask to receive Wi-Fi position request event indications.

9.5.2 Enumeration Type Documentation

9.5.2.1 anonymous enum

Enumerator

eQMI_LOC_SESS_STATUS_SUCCESS
eQMI_LOC_SESS_STATUS_IN_PROGRESS
eQMI_LOC_SESS_STATUS_FAILURE
eQMI_LOC_SESS_STATUS_TIMEOUT

9.5.3 Function Documentation

9.5.3.1 `int pack_loc_DeleteAssistData (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Delete_Assist_Data_t * reqArg)`

Delete Assistant Data pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.2 `int pack_loc_EventRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_EventRegister_t * reqArg)`

Event Register pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.3 `int pack_loc_SetExtPowerState (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetExtPowerState_t * reqArg)`

Set Ext Power State pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.4 int pack_loc_SetOperationMode (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetOperationMode_t * reqArg)

Set Operation Mode pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.5 int pack_loc_SLQSLOCGetBestAvailPos (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCGetBestAvailPos_t * reqArg)

Get Best Avail position pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.6 int pack_loc_Start (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Start_t * reqArg)

LOC Start pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.7 int pack_loc_Stop (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Stop_t * reqArg)

Loc Stop pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.8 int unpack_loc_BestAvailPos_Ind (uint8_t * pResp, uint16_t respLen, unpack_loc_BestAvailPos_Ind_t * pOutput)

Loc Best Avail position Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.9 int unpack_loc_DeleteAssistData (uint8_t * pResp, uint16_t respLen, unpack_loc_Delete_Assist_Data_t * pOutput)

Delete Assistant Data unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.10 int unpack_loc_EngineState_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EngineState_Ind_t * *pOutput*)

Loc Engine State Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.11 int unpack_loc_EventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EventRegister_t * *pOutput*)

Event Register unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.12 int unpack_loc_PositionRpt_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_PositionRpt_Ind_t * *pOutput*)

Loc Position Report Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.13 int unpack_loc_SetExtPowerConfig_Ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerConfig_Ind_t * *pOutput*)

Loc Set External Power Configure Indication unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.14 int unpack_loc_SetExtPowerState (uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerState_t * *pOutput*)

Set Ext Power State unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.15 `int unpack_loc_SetOperationMode (uint8_t * pResp, uint16_t respLen, unpack_loc_SetOperationMode_t * pOutput)`

Set Operation Mode unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.16 `int unpack_loc_SLQSLOCGetBestAvailPos (uint8_t * pResp, uint16_t respLen, unpack_loc_SLQSLOCGetBestAvailPos_t * pOutput)`

Get Best Avail position unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.17 `int unpack_loc_Start (uint8_t * pResp, uint16_t respLen, unpack_loc_Start_t * pOutput)`

Loc Start unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.5.3.18 `int unpack_loc_Stop (uint8_t * pResp, uint16_t respLen, unpack_loc_Stop_t * pOutput)`

Loc Stop unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6 nas.h File Reference

Data Structures

- struct [unpack_nas_GetSignalStrengths_t](#)
- struct [unpack_nas_SLQSGetSysSelectionPref_t](#)
- struct [nas_netSelectionPref](#)
- struct [nas_acqOrderPref](#)
- struct [nas_CSGID](#)
- struct [pack_nas_SLQSSetSysSelectionPref_t](#)
- struct [pack_nas_SLQSNasIndicationRegisterExt_t](#)
- struct [RFBandInfoElements](#)
- struct [unpack_nas_GetRFInfo_t](#)
- struct [cdmaSSInfo](#)
- struct [hdrSSInfo](#)
- struct [lteSSInfo](#)
- struct [tdscdmaSigInfoExt](#)
- struct [unpack_nas_SLQSNasGetSigInfo_t](#)
- struct [unpack_nas_SLQSNasSigInfoCallback_ind_t](#)
- struct [unpack_nas_GetHomeNetwork_t](#)
- struct [nas_SrvStatusInfo](#)
- struct [nas_GSMSrvStatusInfo](#)
- struct [nas_sysInfoCommon](#)
- struct [nas_CDMASysInfo](#)
- struct [nas_HDRSysInfo](#)
- struct [nas_GSMSSysInfo](#)
- struct [nas_WCDMASysInfo](#)
- struct [nas_LTESysInfo](#)
- struct [nas_AddCDMASysInfo](#)
- struct [nas_AddSysInfo](#)
- struct [nas_CallBarringSysInfo](#)
- struct [unpack_nas_SLQSGetSysInfo_t](#)
- struct [unpack_nas_SLQSSysInfoCallback_ind_t](#)
- struct [unpack_nas_GetServingNetwork_t](#)
- struct [unpack_nas_GetServingNetworkCapabilities_t](#)
- struct [nas_QmiNas3GppNetworkInfo](#)

- struct [nas_QmiNas3GppNetworkRAT](#)
- struct [nas_QmisNasPcsDigit](#)
- struct [unpack_nas_PerformNetworkScan_t](#)
- struct [unpack_nas_SLQSSwiGetLteCQI_t](#)
- struct [nas_CommInfo](#)
- struct [nas_LTEInfo](#)
- struct [unpack_nas_SLQSNasSwiModemStatus_t](#)
- struct [nas_servSystem](#)
- struct [nas_dataSrvCapabilities](#)
- struct [nas_currentPLMN](#)
- struct [nas_roamIndList](#)
- struct [nas_qaQmi3Gpp2TimeZone](#)
- struct [nas_detailSvcInfo](#)
- struct [nas_CDMASysInfoExt](#)
- struct [nas_callBarStatus](#)
- struct [unpack_nas_SLQSGetServingSystem_t](#)
- struct [nas_rxSignalStrengthListElement](#)
- struct [nas_ecioListElement](#)
- struct [nas_errorRateListElement](#)
- struct [nas_rsrqInformation](#)
- struct [nas_lteSnrInformation](#)
- struct [nas_lteRsrpInformation](#)
- struct [unpack_nas_SLQSGetSignalStrength_t](#)
- struct [nas_SLQSSignalStrengthsIndReq](#)
- struct [pack_nas_SLQSSetSignalStrengthsCallback_t](#)
- struct [nas_SLQSSignalStrengthsInformation](#)
- struct [nas_RejectReasonTlv](#)
- struct [nas_SignalStrengthTlv](#)
- struct [nas_RFInfoTlv](#)
- struct [nas_SLQSSignalStrengthsTlv](#)
- struct [unpack_nas_SetEventReportInd_t](#)
- struct [unpack_nas_GetCDMANetworkParameters_t](#)
- struct [pack_nas_SetACCOLC_t](#)
- struct [nas_CDMARSSIThresh](#)
- struct [nas_CDMAECIOThresh](#)
- struct [nas_HDRRSSIThresh](#)
- struct [nas_HDRECIOThresh](#)
- struct [nas_HDRSINRThreshold](#)
- struct [nas_HDRIOThresh](#)
- struct [nas_GSMRSSIThresh](#)
- struct [nas_WCDMARSSIThresh](#)
- struct [nas_WCDMAECIOThresh](#)
- struct [nas_LTERSSIThresh](#)
- struct [nas_LTESNRThreshold](#)
- struct [nas_LTERSRQThresh](#)
- struct [nas_LTERSRPThresh](#)
- struct [nas_LTESigRptConfig](#)
- struct [nas_TDSCDMARSCPTthresh](#)
- struct [nas_TDSCDMARSSIThresh](#)
- struct [nas_TDSCDMAECIOThresh](#)
- struct [nas_TDSCDMASINRThresh](#)
- struct [pack_nas_SLQSNasConfigSigInfo2_t](#)
- struct [unpack_nas_SetDataCapabilitiesCallback_ind_t](#)
- struct [unpack_nas_GetNetworkPreference_t](#)
- struct [pack_nas_SetNetworkPreference_t](#)

- struct [unpack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetRoamingIndicatorCallback_ind_t](#)
- struct [NAServingSystemInfo](#)
- struct [unpack_nas_SetServingSystemCallback_ind_t](#)
- struct [NASPhyCaAggScellIndType](#)
- struct [NASPhyCaAggScellIDBw](#)
- struct [NASPhyCaAggScellInfo](#)
- struct [NASPhyCaAggPcellInfo](#)
- struct [NASPhyCaAggScellIndex](#)
- struct [NasGetLTECphyCAInfo](#)
- struct [unpack_nas_SlqsGetLTECphyCAInfo_t](#)
- struct [NASEmergencyModeTlv](#)
- struct [NASModePreferenceTlv](#)
- struct [NASBandPreferenceTlv](#)
- struct [NASPRLPreferenceTlv](#)
- struct [NASRoamPreferenceTlv](#)
- struct [NASLTEBandPreferenceTlv](#)
- struct [NASNetSelPreferenceTlv](#)
- struct [NASServDomainPrefTlv](#)
- struct [NASGWAcqOrderPrefTlv](#)
- struct [NASQmiCbkNasSystemSelPrefInd](#)
- struct [unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t](#)
- struct [NASOTAMessageTlv](#)
- struct [NASLteNasReleaseInfoTlv](#)
- struct [NASTimeInfoTlv](#)
- struct [NASQmiCbkNasSwiOTAMessageInd](#)
- struct [unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t](#)
- struct [nas_MNRInfo](#)
- struct [pack_nas_SLQSInitiateNetworkRegistration_t](#)
- struct [pack_nas_SLQSNasSwiOTAMessageCallback_t](#)
- struct [pack_nas_SLQSGetPLMNName_t](#)
- struct [unpack_nas_SLQSGetPLMNName_t](#)
- struct [nas_nmrCellInfo](#)
- struct [nas_GERANInfo](#)
- struct [nas_geranInstInfo](#)
- struct [nas_UMTSinstInfo](#)
- struct [nas_UMTSInfo](#)
- struct [nas_CDMAInfo](#)
- struct [nas_cellParams](#)
- struct [nas_LTEInfoIntrafreq](#)
- struct [nas_infoInterFreq](#)
- struct [nas_LTEInfoInterfreq](#)
- struct [nas_gsmCellInfo](#)
- struct [nas_lteGsmCellInfo](#)
- struct [nas_LTEInfoNeighboringGSM](#)
- struct [nas_wcdmaCellInfo](#)
- struct [nas_lteWcdmaCellInfo](#)
- struct [nas_LTEInfoNeighboringWCDMA](#)
- struct [nas_umtsLTENbrCell](#)
- struct [nas_WCDMAInfoLTENeighborCell](#)
- struct [unpack_nas_SLQSNasGetCellLocationInfo_t](#)
- struct [nas_timeInfo](#)
- struct [unpack_nas_SLQSGetNetworkTime_t](#)
- struct [nas_UniversalTime](#)
- struct [unpack_nas_SLQSNasNetworkTimeCallBack_ind_t](#)

- struct [nas_PhyCaAggScellIndType](#)
- struct [nas_PhyCaAggScellDIBw](#)
- struct [nas_PhyCaAggScellInfo](#)
- struct [nas_PhyCaAggPcellInfo](#)
- struct [nas_PhyCaAggScellIndex](#)
- struct [unpack_nas_SetNasLTECphyCaIndCallback_ind_t](#)

Macros

- #define [NAS_OTA_MESSAGE_MAX_BUF_SIZE](#) 2048
- #define [NAS_MAX_NUM_NETWORKS](#) 30
- #define [NAS_MAX_DESCRIPTION_LENGTH](#) 255
- #define [NAS_PLMN_LENGTH](#) 3
- #define [NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST](#) 255

Enumerations

- enum [LIBPACK_NAS_LTE_CPHY_SCELL_STATE](#) {
[eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED](#) =0x00,
[eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED](#) =0x01,
[eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED](#) =0x02 }
- enum [LIBPACK_NAS_LTE_CPHY_CA_BW_NRB](#) {
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6](#) =0x00,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15](#) =0x01,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25](#) =0x02,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50](#) =0x03,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75](#) =0x04,
[eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100](#) =0x05 }
- enum [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) {
[eNAS_LTE_CPHY_CA_BW_NRB_LITE_6](#) =0x00,
[eNAS_LTE_CPHY_CA_BW_NRB_LITE_15](#) =0x01,
[eNAS_LTE_CPHY_CA_BW_NRB_LITE_25](#) =0x02,
[eNAS_LTE_CPHY_CA_BW_NRB_LITE_50](#) =0x03,
[eNAS_LTE_CPHY_CA_BW_NRB_LITE_75](#) =0x04,
[eNAS_LTE_CPHY_CA_BW_NRB_LITE_100](#) =0x05 }
- enum [NAS_LTE_CPHY_SCELL_STATE_LITE](#) {
[eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE](#) =0x00,
[eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE](#) =0x01,
[eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE](#) =0x02 }

Functions

- int [unpack_nas_GetSignalStrengths](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetSignalStrengths_t](#) *pOutput)
- int [pack_nas_GetSignalStrengths](#) (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen)
- int [pack_nas_SLQSGetSysSelectionPref](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSGetSysSelectionPref](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetSysSelectionPref_t](#) *pOutput)
- int [pack_nas_SLQSSetSysSelectionPref](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSSetSysSelectionPref_t](#) *pReqParam)
- int [unpack_nas_SLQSSetSysSelectionPref](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSSetBandPreference](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint64_t bandPref)
- int [unpack_nas_SLQSSetBandPreference](#) (uint8_t *pResp, uint16_t respLen)

- int [pack_nas_SLQSNasIndicationRegisterExt](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSNasIndicationRegisterExt_t](#) *pReqParam)
- int [unpack_nas_SLQSNasIndicationRegisterExt](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_GetRFInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetRFInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetRFInfo_t](#) *pOutput)
- int [pack_nas_SLQSNasGetSigInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSNasGetSigInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasGetSigInfo_t](#) *pOutput)
- int [unpack_nas_SLQSNasSigInfoCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSigInfoCallback_ind_t](#) *pOutput)
- int [unpack_nas_GetHomeNetwork](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetHomeNetwork_t](#) *pOutput)
- int [pack_nas_GetHomeNetwork](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [pack_nas_SLQSGetSysInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSGetSysInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetSysInfo_t](#) *pOutput)
- int [unpack_nas_SLQSNasSysInfoCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSysInfoCallback_ind_t](#) *pOutput)
- int [pack_nas_GetServingNetwork](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetServingNetwork](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetServingNetwork_t](#) *pOutput)
- int [pack_nas_GetServingNetworkCapabilities](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetServingNetworkCapabilities](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetServingNetworkCapabilities_t](#) *pOutput)
- int [pack_nas_PerformNetworkScan](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_PerformNetworkScan](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_PerformNetworkScan_t](#) *pOutput)
- int [pack_nas_SLQSSwiGetLteCQI](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSSwiGetLteCQI](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSwiGetLteCQI_t](#) *pOutput)
- int [pack_nas_SLQSNasSwiModemStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSNasSwiModemStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSwiModemStatus_t](#) *pOutput)
- int [pack_nas_SLQSGetServingSystem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSGetServingSystem](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetServingSystem_t](#) *pOutput)
- int [pack_nas_SLQSGetSignalStrength](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint16_t reqMask)
- int [unpack_nas_SLQSGetSignalStrength](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetSignalStrength_t](#) *pOutput)
- int [pack_nas_SLQSSetSignalStrengthsCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSSetSignalStrengthsCallback_t](#) *pReqParam)
- int [unpack_nas_SLQSSetSignalStrengthsCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SetRFInfoCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int [unpack_nas_SetRFInfoCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SetLURejectCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int [unpack_nas_SetLURejectCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_nas_SetEventReportInd](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetEventReportInd_t](#) *pOutput)
- int [pack_nas_GetCDMANetworkParameters](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetCDMANetworkParameters](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetCDMA-NetworkParameters_t](#) *pOutput)
- int [pack_nas_GetANAAAAuthenticationStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetANAAAAuthenticationStatus](#) (uint8_t *pResp, uint16_t respLen, uint32_t *pAuthStatus)
- int [pack_nas_GetACCOLC](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int [unpack_nas_GetACCOLC](#) (uint8_t *pResp, uint16_t respLen, uint8_t *pAccolc)
- int [pack_nas_SetACCOLC](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SetACCOLC_t](#) reqParam)
- int [unpack_nas_SetACCOLC](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSNasConfigSigInfo2](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSNasConfigSigInfo2_t](#) *pReqParam)
- int [unpack_nas_SLQSNasConfigSigInfo2](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_nas_SetDataCapabilitiesCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetDataCapabilitiesCallback_ind_t](#) *pOutput)
- int [pack_nas_GetNetworkPreference](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_GetNetworkPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_GetNetworkPreference_t](#) *pOutput)
- int [pack_nas_SetNetworkPreference](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SetNetworkPreference_t](#) *reqArg)
- int [unpack_nas_SetNetworkPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetNetworkPreference_t](#) *pOutput)
- int [unpack_nas_SetRoamingIndicatorCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetRoamingIndicatorCallback_ind_t](#) *pOutput)
- int [unpack_nas_SetServingSystemCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetServingSystemCallback_ind_t](#) *pOutput)
- int [pack_nas_SlqsGetLTECphyCAInfo](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SlqsGetLTECphyCAInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SlqsGetLTECphyCAInfo_t](#) *pOutput)
- int [unpack_nas_SLQSSetSysSelectionPrefCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t](#) *pOutput)
- int [unpack_nas_SLQSNasSwiOTAMessageCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t](#) *pOutput)
- int [pack_nas_SLQSInitiateNetworkRegistration](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSInitiateNetworkRegistration_t](#) *pReqParam)
- int [unpack_nas_SLQSInitiateNetworkRegistration](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSNasSwiOTAMessageCallback](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSNasSwiOTAMessageCallback_t](#) *pReqParam)
- int [unpack_nas_SLQSNasSwiOTAMessageCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_nas_SLQSGetPLMNName](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_nas_SLQSGetPLMNName_t](#) *reqArg)
- int [unpack_nas_SLQSGetPLMNName](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetPLMNName_t](#) *pOutput)
- int [pack_nas_SLQSNasGetCellLocationInfo](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSNasGetCellLocationInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasGetCellLocationInfo_t](#) *pOutput)
- int [pack_nas_SLQSGetNetworkTime](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_nas_SLQSGetNetworkTime](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSGetNetworkTime_t](#) *pOutput)
- int [unpack_nas_SLQSNasNetworkTimeCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSNasNetworkTimeCallBack_ind_t](#) *pOutput)
- int [unpack_nas_SetNasLTECphyCalndCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SetNasLTECphyCalndCallback_ind_t](#) *pOutput)

9.6.1 Macro Definition Documentation

9.6.1.1 `#define NAS_MAX_DESCRIPTION_LENGTH 255`

9.6.1.2 `#define NAS_MAX_NUM_NETWORKS 30`

9.6.1.3 `#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048`

9.6.1.4 `#define NAS_PLMN_LENGTH 3`

9.6.1.5 `#define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255`

9.6.2 Enumeration Type Documentation

9.6.2.1 `enum LIBPACK_NAS_LTE_CPHY_CA_BW_NRB`

Enumerator

eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75
eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100

9.6.2.2 `enum LIBPACK_NAS_LTE_CPHY_SCELL_STATE`

Enumerator

eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED

9.6.2.3 `enum NAS_LTE_CPHY_CA_BW_NRB_LITE`

Enumerator

eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100

9.6.2.4 `enum NAS_LTE_CPHY_SCELL_STATE_LITE`

Enumerator

eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE

9.6.3 Function Documentation

9.6.3.1 `int pack_nas_GetACCOLC (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

<i>in, out</i>	<i>pCtx</i>	qmi request context
<i>out</i>	<i>pReq</i>	qmi request buffer
<i>out</i>	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.2 `int pack_nas_GetANAAAuthenticationStatus (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.3 `int pack_nas_GetCDMANetworkParameters (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.4 `int pack_nas_GetHomeNetwork (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get home network pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.5 `int pack_nas_GetNetworkPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

9.6.3.6 `int pack_nas_GetRFInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get rf info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.7 `int pack_nas_GetServingNetwork (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.8 `int pack_nas_GetServingNetworkCapabilities (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.9 int pack_nas_GetSignalStrengths (pack_qmi_t * pCtx, uint8_t * pReq, uint16_t * pLen)

get signal strengths pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.10 int pack_nas_PerformNetworkScan (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.11 int pack_nas_SetACCOLC (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SetACCOLC_t reqParam)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.12 `int pack_nas_SetLURejectCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>bEnable</i>	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.13 `int pack_nas_SetNetworkPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SetNetworkPreference_t * reqArg)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pack</i>	default prototype

9.6.3.14 `int pack_nas_SetRFInfoCallback (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>bEnable</i>	0/1 value to disable/enable indication respectively

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.15 `int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

9.6.3.16 `int pack_nas_SLQSGetNetworkTime (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.17 `int pack_nas_SLQSGetPLMNName (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSGetPLMNName_t * reqArg)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.18 `int pack_nas_SLQSGetServingSystem (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.19 `int pack_nas_SLQSGetSignalStrength (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint16_t reqMask)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqMask</i>	request mask for fetching extra signal info

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.20 `int pack_nas_SLQSGetSysInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.21 `int pack_nas_SLQSGetSysSelectionPref (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.22 `int pack_nas_SLQSIInitiateNetworkRegistration (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSIInitiateNetworkRegistration_t * pReqParam)`

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.23 int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_nas_SLQSNasConfigSigInfo2_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.24 int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.25 int pack_nas_SLQSNasGetSigInfo (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get sig info pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.26 int pack_nas_SLQSNasIndicationRegisterExt (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasIndicationRegisterExt_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.27 int pack_nas_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

9.6.3.28 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

9.6.3.29 int pack_nas_SLQSSetBandPreference (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint64_t *bandPref*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>band</i>	preference

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.30 int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSSetSignalStrengthsCallback_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request prarmeters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.31 int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSSetSysSelectionPref_t * *pReqParam*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.32 int pack_nas_SLQSSwiGetLteCQI (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.33 int unpack_nas_GetACCOLC (uint8_t * *pResp*, uint16_t *respLen*, uint8_t * *pAccolc*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pAccolc</i>	accolc

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.34 int unpack_nas_GetANAAAAAuthenticationStatus (uint8_t * *pResp*, uint16_t *respLen*, uint32_t * *pAuthStatus*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>authStatus</i>	auth status

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.35 int unpack_nas_GetCDMANetworkParameters (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_GetCDMANetworkParameters_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	qmi output parameters

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.36 int unpack_nas_GetHomeNetwork (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetHomeNetwork_t *
pOutput)

get home network unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.37 int unpack_nas_GetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetNetwork-
Preference_t * *pOutput*)

9.6.3.38 int unpack_nas_GetRFInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetRFInfo_t * *pOutput*)

get rf info unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.39 `int unpack_nas_GetServingNetwork (uint8_t * pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.40 `int unpack_nas_GetServingNetworkCapabilities (uint8_t * pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.41 `int unpack_nas_GetSignalStrengths (uint8_t * pResp, uint16_t respLen, unpack_nas_GetSignalStrengths_t * pOutput)`

get signal strengths unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.42 int unpack_nas_PerformNetworkScan (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_PerformNetworkScan_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.43 int unpack_nas_SetACCOLC (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.44 int unpack_nas_SetDataCapabilitiesCallback_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetDataCapabilitiesCallback_ind_t * *pOutput*)

Data Capabilities indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.45 `int unpack_nas_SetEventReportInd (uint8_t * pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t * pOutput)`

9.6.3.46 `int unpack_nas_SetLURjectCallback (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.47 `int unpack_nas_SetNasLTECphyCalIndCallback_ind (uint8_t * pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCalIndCallback_ind_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.48 `int unpack_nas_SetNetworkPreference (uint8_t * pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t * pOutput)`

9.6.3.49 `int unpack_nas_SetRFInfoCallback (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.50 `int unpack_nas_SetRoamingIndicatorCallback_ind (uint8_t * pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t * pOutput)`

Roaming indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.51 `int unpack_nas_SetServingSystemCallback_ind (uint8_t * pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t * pOutput)`

9.6.3.52 `int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t * pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t * pOutput)`

9.6.3.53 `int unpack_nas_SLQSGetNetworkTime (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.54 `int unpack_nas_SLQSGetPLMNName (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.55 `int unpack_nas_SLQSGetservingSystem (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetservingSystem_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.56 `int unpack_nas_SLQSGetSignalStrength (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.57 `int unpack_nas_SLQSGetSysInfo (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.58 `int unpack_nas_SLQSGetSysSelectionPref (uint8_t * pResp, uint16_t respLen,
unpack_nas_SLQSGetSysSelectionPref_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.59 `int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.60 `int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.61 `int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t * pResp, uint16_t respLen,
unpack_nas_SLQSNasGetCellLocationInfo_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.62 int unpack_nas_SLQSNasGetSigInfo (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetSigInfo_t * *pOutput*)

get sig info unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.63 int unpack_nas_SLQSNasIndicationRegisterExt (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.64 int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.65 int unpack_nas_SLQSNasSigInfoCallback_ind (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSNasSigInfoCallback_ind_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.66 int unpack_nas_SLQSNasSwtModemStatus (uint8_t * *pResp*, uint16_t *respLen*,
unpack_nas_SLQSNasSwtModemStatus_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.67 int unpack_nas_SLQSNasSwtOTAMessageCallback (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.68 `int unpack_nas_SLQSNasSwiOTAMessageCallback_ind (uint8_t * pResp, uint16_t respLen,
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t * pOutput)`

OTA message indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.69 `int unpack_nas_SLQSNasSysInfoCallback_ind (uint8_t * pResp, uint16_t respLen,
unpack_nas_SLQSSysInfoCallback_ind_t * pOutput)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.70 `int unpack_nas_SLQSSetBandPreference (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

9.6.3.71 `int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t * pResp, uint16_t respLen)`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.72 int unpack_nas_SLQSSetSysSelectionPref (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.73 int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t * *pOutput*)

System Selection Preference indication unpack

Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.6.3.74 int unpack_nas_SLQSSwiGetLteCQI (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetLteCQI_t * *pOutput*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.7 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

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_OUTGOINGINTEXTHOME - Outgoing external to home
- 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming
- 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred
- 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI_VOICE_REASON_CALLWAITING - Call waiting
- 0x10 - VOICE_CC_SUPS_RESULT_REASON_CLIP - CLIP
- 0x11 - VOICE_CC_SUPS_RESULT_REASON_CLIR - CLIR
- 0x12 - VOICE_CC_SUPS_RESULT_REASON_COLP - COLP
- 0x13 - VOICE_CC_SUPS_RESULT_REASON_COLR - COLR
- 0x14 - VOICE_CC_SUPS_RESULT_REASON_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

9.9 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 SNDCP(Sub Network Dependent Convergence Protocol) failure
- 1005 - Insufficient resources, 3GPP equivalent ESM(EPS Session Management) cause code value 26, Insufficient resources
- 1006 - Service option temporarily out of order, 3GPP equivalent ESM(EPS Session Management) cause code value 34, Service option temporarily out of order
- 1007 - PTI already used, 3GPP equivalent ESM(EPS Session Management) cause code value 35, PTI(- Procedure Transaction Identity) already in use
- 1008 - Regular PDP context deactivation, 3GPP equivalent ESM(EPS Session Management) cause code value 36, Regular deactivation
- 1009 - Network failure, 3GPP equivalent ESM(EPS Session Management) cause code value 38, Network failure
- 1010 - Reactivation requested, 3GPP equivalent ESM(EPS Session Management) cause code value 39, Reactivation requested
- 1011 - Protocol error, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 111, Protocol error, unspecified
- 1012 - Operator determined barring, 3GPP equivalent ESM(EPS Session Management) cause code value 8, Operator Determined Barring
- 1013 - Unknown or missing Access Point Name (APN), 3GPP equivalent ESM(EPS Session Management) cause code value 27, Missing or unknown APN
- 1014 - Unknown PDP address or PDP type, 3GPP equivalent ESM(EPS Session Management) cause code value 28, Unknown PDN type
- 1015 - Activation rejected by GGSN, 3GPP equivalent ESM(EPS Session Management) cause code value 30, Requested rejected by Serving GW or PDN GW
- 1016 - Activation rejected, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 31, Request rejected, unspecified
- 1017 - Service option not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 32, Service option not supported
- 1018 - Requested service option not subscribed, 3GPP equivalent ESM(EPS Session Management) cause code value 33, Requested service option not subscribed
- 1019 - EPS Quality of Service (QoS) not accepted, 3GPP equivalent ESM(EPS Session Management) cause code value 37, EPS QoS not accepted
- 1020 - Semantic error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 41, Semantic error in the TFT operation
- 1021 - Syntactical error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 42, Syntactical error in the TFT operation
- 1022 - Unknown PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 43, Invalid EPS bearer identity

- 1023 - Semantic errors in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 44, Semantic errors in packet filter(s)
- 1024 - Syntactical error in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 45, Syntactical errors in packet filter(s)
- 1025 - PDP context without TFT already activated, 3GPP equivalent ESM(EPS Session Management) cause code value 46, Unused
- 1026 - Invalid transaction identifier value, 3GPP equivalent ESM(EPS Session Management) cause code value 81, Invalid PTI value
- 1027 - Semantically incorrect message, 3GPP equivalent ESM(EPS Session Management) cause code value 95, Semantically incorrect message
- 1028 - Invalid mandatory information, 3GPP equivalent ESM(EPS Session Management) cause code value 96, Invalid mandatory information
- 1029 - Message type non-existent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 97, Message type non-existent or not implemented
- 1030 - Message not compatible with state, 3GPP equivalent ESM(EPS Session Management) cause code value 98, Message type not compatible with the protocol state
- 1031 - Information element nonexistent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 99, Information element non-existent or not implemented
- 1032 - Conditional information element error, 3GPP equivalent ESM(EPS Session Management) cause code value 100, Conditional IE error
- 1033 - Message not compatible with protocol state, 3GPP equivalent ESM(EPS Session Management) cause code value 101, Message not compatible with the protocol state
- 1034 - APN restriction value incompatible with active PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 112, APN restriction value incompatible with
 - active EPS bearer context
- 1035 - No GPRS context present
- 1036 - Requested feature not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 40, Feature not supported
- 1037 - Illegal MS, 3GPP equivalent EMM(EPS Mobility Management) cause code value 3, Illegal UE (MS)
- 1038 - Illegal ME, 3GPP equivalent EMM(EPS Mobility Management) cause code value 6, Illegal ME. This error code is sent to the MS if the ME used is not acceptable
 - to the network, e.g. blacklisted
- 1039 - GPRS and non GPRS services not allowed
- 1040 - GPRS services not allowed
- 1041 - MS identity not derived by the network, 3GPP equivalent EMM(EPS Mobility Management) cause code value 9, UE (MS) Identify cannot be derived by the network
- 1042 - Implicitly detached, 3GPP equivalent EMM(EPS Mobility Management) cause code value 10, Implicitly Detached
- 1043 - PLMN not allowed, 3GPP equivalent EMM(EPS Mobility Management) cause code value 11, PLMN not allowed
- 1044 - LA not allowed, this cause is sent to the MS if it requests location updating in a location area where the HPLMN determines that the MS, by subscription, is not allowed to operate.
- 1045 - GPRS services not allowed in PLMN

- 1046 - PDP duplicate
- 1047 - UE radio access technology change
- 1048 - app preempted
- 1049 - Congestion, This cause is sent if the service request or LOCATION UPDATING REQUEST message cannot be actioned because of congestion (e.g. congestion of the MSC or SGSN or GGSN or PDN Gateway; no channel; facility busy/congested etc.).
- 1050 - No PDP context activated
- 1051 - Access class DSAC rejection

9.9.2.4 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(C(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRR(C(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRR(C(LTE Radio Resource Control) UL DATA CNF FAILURE CONN REL
- 1049 - LRR(C(LTE Radio Resource Control) UL DATA CNF FAILURE RLF
- 1050 - LRR(C(LTE Radio Resource Control) UL DATA CNF FAILURE CTRL NOT CONN
- 1051 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE
- 1052 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE ABORTED
- 1053 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE ACCESS BARRED
- 1054 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRR(C(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRR(C(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRR(C(LTE Radio Resource Control) CONN REL RLF

- 1063 - LRRCLTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRCLTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRCLTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRCLTE Radio Resource Control) CONN REL SIB READ ERROR
- 1067 - DETACH WITH REATTACH LTE NW DETACH
- 1068 - DETACH WITH OUT REATTACH LTE NW DETACH
- 1069 - ESM(EPS Session Management) PROC TIME OUT
- 1070 - MESSAGE EXCEED MAX L2 LIMIT
- 1500 - CD GEN OR BUSY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of general or network busy
- 1501 - CD BILL OR AUTH, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of billing failure or authentication failure
- 1502 - CHG HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is a change to HDR system due to redirection or PRL not preferred
- 1503 - EXIT HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device exited HDR due to redirection or PRL not preferred
- 1504 - HDR NO SESSION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have a HDR session
- 1505 - HDR ORIG DURING GPS FIX, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since it is ending an HDR call origination in favor of a GPS fix
- 1506 - HDR CS TIMEOUT, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since connection setup on HDR system timed out
- 1507 - HDR RELEASED BY CM, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when it wants to release a HDR call so a 1X call can continue
- 1508 - COLLOC ACQ FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when device failed to acquire co-located HDR for origination
- 1509 - OTASP COMMIT IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since an OTASP commit is in progress
- 1510 - NO HYBR HDR SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device has no Hybrid HDR service
- 1511 - HDR NO LOCK GRANTED, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module could not get the RF lock
- 1512 - HOLD OTHER IN PROG, this error code is returned when data call is brought down by CM(Call Manager) because DBM or SMS is in progress
- 1513 - HDR FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module released the call due to fade
- 1514 - HDR ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to HDR system Access Failure
- 2000 - CLIENT END, this error code is returned when client ends the data call

- 2001 - NO SRV, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device has no service
- 2002 - FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device lost the system due to fade
- 2003 - REL NORMAL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with no reason
- 2004 - ACC IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access attempt already in progress
- 2005 - ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access Failure
- 2006 - REDIR OR HANDOFF, this error code is returned when data call is brought down because device is in the process of redirecting/handing off to a different target system
- 2500 - OFFLINE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device went offline
- 2501 - EMERGENCY MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device is operating in Emergency mode
- 2502 - PHONE IN USE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device is in use (e.g voice call)
- 2503 - INVALID MODE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the device's operational mode is different from the mode requested in the traffic channel bring up
- 2504 - INVALID SIM STATE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since the SIM was marked by network as invalid for circuit and/or packet service domain
- 2505 - NO COLLOC HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is no collocated HDR
- 2506 - CALL CONTROL REJECTED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since Call control module rejected the request

9.9.2.9 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 SNDP FAILURE, PDP context deactivation initiated by the MS or by the Network
- 26 - INSUFFICIENT RESOURCES, this reason is posted to indicate that the network cannot provide the requested service due to insufficient resources
- 27 - MISSING OR UNKNOWN APN, the APN was required and not specified or APN could not be resolved. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 28 - UNKNOWN PDN TYPE, the reason is posted by the network to indicate that the PDN type was not recognized
- 29 - AUTH FAILED, the reason is posted when authentication fails. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 30 - GGSN REJECT, the reason is posted when the request was rejected by Serving GW or PDN GW. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN

- 31 - ACTIVATION REJECT, the reason is posted when the request is rejected by the network due to unspecified reasons
- 32 - OPTION NOT SUPPORTED, the reason is posted when UE requested a service not supported by the PLMN
- 33 - OPTION UNSUBSCRIBED, This cause is sent when the MS requests a service option for which it has no subscription
- 34 - OPTION TEMP OOO, service option temporarily out of order, this reason is posted when the network is temporarily out of resources to service the request
- 35 - PTI ALREADY USED, the reason is posted to indicate that PTI (Procedure Transaction Identifier) used in the request is already active via another UE requested procedure
- 36 - REGULAR DEACTIVATION, this reason is posted by the network to initiate a regular release of bearer resources
- 37 - EPS QOS NOT ACCEPTED, this reason is posted by the network to indicate that the QoS requested by the UE could not be accepted
- 38 - NETWORK FAILURE, this reason is posted when an error occurs in the network
- 39 - UMTS REACTIVATION REQ, this reason is posted by the network to request for bearer reactivation. This code may be posted during network congestion
- 40 - FEATURE NOT SUPPORTED, Unsuccessful MBMS context activation requested by the network
- 41 - TFT SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying TFT operation included in the request
- 42 - TFT SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying TFT operation included in the request
- 43 - UNKNOWN PDP CONTEXT, the reason is posted when the bearer identity (or linked bearer identity) in the request is invalid (or inactive)
- 44 - FILTER SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying packet filter(s) associated with a TFT
- 45 - FILTER SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying packet filter(s) associated with a TFT
- 46 - PDP WITHOUT ACTIVE TFT, the reason is posted by the network when UW requested more than one PDP connection without TFT
- 50 - IPV4 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 50, PDN type IPv4 only allowed.
- 51 - IPV6 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 51, PDN type IPv6 only allowed
- 52 - SINGLE ADDRESS BEARER ONLY, 3GPP equivalent ESM(EPS Session Management) cause code value 52, Single address bearers only allowed. The reason is posted when the network supports single address bearers only, meaning dual IP bearers are not supported
- 53 - ESM INFORMATION NOT RECEIVED, 3GPP equivalent ESM(EPS Session Management) cause code value 53, ESM information not received. The reason is posted by the network to indicate that the PDN connection request was rejected because ESM information was not received
- 54 - PND CONNECTION DOES NOT EXIST, 3GPP equivalent ESM(EPS Session Management) cause code value 54, PDN connection does not exist The reason is posted by the network during handover from a non-3G-PP network to indicate that the MME does not have any information regarding the requested PDN connection

- 55 - MULTIPLE CONNECTION TO SAME PDN NOT ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 55, Multiple PDN connections for a given APN not allowed. The reason is posted by the network to indicate that the UE is already connected to the requested APN via another PDN/PDN connection
- 81 - INVALID TRANSACTION ID, the reason is posted by the network to indicate that the PTI used in the request is unassigned or reserved
- 95 - MESSAGE INCORRECT SEMANTIC, the reason is posted by the network to indicate receipt of an invalid message
- 96 - INVALID MANDATORY INFO, the reason is posted by the network to indicate receipt of a message with semantic error in a mandatory information element
- 97 - MESSAGE TYPE UNSUPPORTED, the reason is posted by the network to indicate receipt of a message that is either undefined or defined but not implemented by the equipment sending this ESM cause
- 98 - MSG TYPE NONCOMPATIBLE STATE, the reason is posted by the network to indicate receipt of a message type that cannot be handled in the current network protocol state
- 99 - UNKNOWN INFO ELEMENT, the reason is posted by the network to indicate receipt of a message that includes an information element that is either not defined or defined but not implemented by the equipment sending the ESM cause
- 100 - CONDITIONAL IE ERROR, the reason is posted by the network to indicate receipt of a message that includes a syntactically incorrect information element. This message is ignored by the network.
- 101 - MSG AND PROTOCOL STATE UNCOMPATIBLE, the reason is posted by the network to indicate receipt of a message that cannot be handled in the current network protocol state
- 111 - PROTOCOL ERROR, the reason is posted by the network to indicate a protocol error when no other error applies
- 112 - APN TYPE CONFLICT
- 113 - INVALID PROXY-CALL SESSION CONTROL FUNCTION ADDRESS

9.9.2.10 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 to 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

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

Valid transitions for Power Modes

- Online to Low Power, Persistent low power, Factory test, Offline or Shut Down
- Low power to online, Persistent low power, Offline, or Shut Down
- Persistent low power to Online, Low power, Offline or Shut down
- Factory test to online
- Offline to Reset

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.14 qaGobiApiTableRadioInterfaces.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 - Conditonal 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

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_MULTIPLE_SMS_UNSUPPORTED,
 - eQCWWAN_ERR_ENUM_END,
 - eQCWWAN_ERR_QMI_OFFSET = 1000,
 - eQCWWAN_ERR_QMI_MALFORMED_MSG = 1001,
 - eQCWWAN_ERR_QMI_NO_MEMORY,
 - eQCWWAN_ERR_QMI_INTERNAL,
 - eQCWWAN_ERR_QMI_ABORTED,
 - eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED,
 - eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION,
 - eQCWWAN_ERR_QMI_INVALID_CLIENT_ID,
 - eQCWWAN_ERR_QMI_NO_THRESHOLDS,
 - eQCWWAN_ERR_QMI_INVALID_HANDLE,
 - eQCWWAN_ERR_QMI_INVALID_PROFILE,
 - eQCWWAN_ERR_QMI_INVALID_PINID,
 - eQCWWAN_ERR_QMI_INCORRECT_PIN,
 - eQCWWAN_ERR_QMI_NO_NETWORK_FOUND,
 - eQCWWAN_ERR_QMI_CALL_FAILED,
 - eQCWWAN_ERR_QMI_OUT_OF_CALL,
 - eQCWWAN_ERR_QMI_NOT_PROVISIONED,
 - eQCWWAN_ERR_QMI_MISSING_ARG,
 - eQCWWAN_ERR_QMI_ARG_TOO_LONG = 1019,
 - eQCWWAN_ERR_QMI_INVALID_TX_ID = 1022,
 - eQCWWAN_ERR_QMI_DEVICE_IN_USE,
 - eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED,
 - eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED,
 - eQCWWAN_ERR_QMI_NO_FREE_PROFILE,
 - eQCWWAN_ERR_QMI_INVALID_PDP_TYPE,
 - eQCWWAN_ERR_QMI_INVALID_TECH_PREF,
 - eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE,

```

eQCWWAN_ERR_QMI_WIDTH = 0xFFFF }
• enum qm_wds_ds_profile_extended_err_codes {
eWDS_ERR_PROFILE_REG_RESULT_FAIL = 1,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID,
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED,
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID,
eWDS_ERR_PROFILE_REG_RESULT_LIST_END,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID,
eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY,
eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY = 1001,
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR,
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED,
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET,
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET,
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES,
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE = 1101,
eWDS_ERR_PROFILE_REG_END }

```

9.21.1 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_MULTIPLE_SMS_UNSUPPORTED 37- Unsupported multiple SMS

eQCWWAN_ERR_ENUM_END End of SLQS SDK specific error codes

eQCWWAN_ERR_QMI_OFFSET 1000 - This is not an error code but the offset from which mapped QMI error codes start from

eQCWWAN_ERR_QMI_MALFORMED_MSG 1001 - Malformed or Corrupted QMI msg

eQCWWAN_ERR_QMI_NO_MEMORY 1002 - Device could not allocate memory for QMI Resp

eQCWWAN_ERR_QMI_INTERNAL 1003 - Unexpected error occurred during processing

eQCWWAN_ERR_QMI_ABORTED 1004 - Processing aborted

eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED 1005 - QMI client IDs have been exhausted

eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION 1006 - Unable to abort QMI transaction

eQCWWAN_ERR_QMI_INVALID_CLIENT_ID 1007 - Invalid QMI client ID

eQCWWAN_ERR_QMI_NO_THRESHOLDS 1008 - No thresholds were provided

eQCWWAN_ERR_QMI_INVALID_HANDLE 1009 - Invalid Handle provided in the QMI request

eQCWWAN_ERR_QMI_INVALID_PROFILE 1010 - Profile specified is invalid

eQCWWAN_ERR_QMI_INVALID_PINID 1011 - Invalid PIN ID specified

eQCWWAN_ERR_QMI_INCORRECT_PIN 1012 - Incorrect PIN ID specified

eQCWWAN_ERR_QMI_NO_NETWORK_FOUND 1013 - No network found

eQCWWAN_ERR_QMI_CALL_FAILED 1014 - Call failed

eQCWWAN_ERR_QMI_OUT_OF_CALL 1015 - Device is not in a call

eQCWWAN_ERR_QMI_NOT_PROVISIONED 1016 - Requested information element not provisioned on device

eQCWWAN_ERR_QMI_MISSING_ARG 1017 - Mandatory QMI TLV not provided

eQCWWAN_ERR_QMI_ARG_TOO_LONG 1019 - Arg passed in QMI TLV larger than available storage in device

eQCWWAN_ERR_QMI_INVALID_TX_ID 1022 - Invalid TX ID specified

eQCWWAN_ERR_QMI_DEVICE_IN_USE 1023 - Device currently in a call

eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED 1024 - The selected operation is not supported by the network

eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED 1025 - The selected operation is not supported by the device

eQCWWAN_ERR_QMI_NO_EFFECT 1026 - Requested operation would have no effect

eQCWWAN_ERR_QMI_NO_FREE_PROFILE 1027 - No space for a profile is available

eQCWWAN_ERR_QMI_INVALID_PDP_TYPE 1028 - Invalid PDP type specified

eQCWWAN_ERR_QMI_INVALID_TECH_PREF 1029 - Invalid technology preference specified

eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE 1030 - Invalid profile type specified

eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE 1031 - Invalid service type specified
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION 1032 - Invalid register action specified
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION 1033 - Invalid PS attach/detach action specified
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED 1034 - Authentication of supplied information element failed
eQCWWAN_ERR_QMI_PIN_BLOCKED 1035 - PIN is blocked; an unblock operation needs to be issued
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED 1036 - PIN is permanently blocked; the UIM is unusable
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED 1037 - UIM initialization has not completed
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE 1038 - Max QOS requests are used
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER 1039 - The Flow filter is incorrect
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE 1040 - Network unaware of the QOS requested
eQCWWAN_ERR_QMI_INVALID_ID 1041 - Invalid QOS ID
eQCWWAN_ERR_QMI_INVALID_QOS_ID 1041 - Invalid QOS ID
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED 1042 - The request number is not supported

eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND 1043 - Unable to find the interface
eQCWWAN_ERR_QMI_FLOW_SUSPENDED 1044 - Flow suspended
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT 1045 - Data format is invalid
eQCWWAN_ERR_QMI_GENERAL 1046 - General error
eQCWWAN_ERR_QMI_UNKNOWN 1047 - Unknown error
eQCWWAN_ERR_QMI_INVALID_ARG 1048 - A specified argument is invalid
eQCWWAN_ERR_QMI_INVALID_INDEX 1049 - A specified index is invalid
eQCWWAN_ERR_QMI_NO_ENTRY 1050 - No information element exists at specified memory designation
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL 1051 - The memory storage specified in the request is full
eQCWWAN_ERR_QMI_DEVICE_NOT_READY 1052 - Device not in a ready state
eQCWWAN_ERR_QMI_NETWORK_NOT_READY 1053 - Network not in a ready state
eQCWWAN_ERR_QMI_CAUSE_CODE 1054 - Error provided in SMS cause code
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT 1055 - The message could not be sent
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE 1056 - The message could not be delivered
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID 1057 - The message ID specified for the message is invalid

eQCWWAN_ERR_QMI_ENCODING 1058 - The message is not encoded properly
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK 1059 - Maximum number of authentication failures has been reached
eQCWWAN_ERR_QMI_INVALID_TRANSITION 1060 - Operating mode transition from the current mode is invalid
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE 1061 - The intercase is not muticast
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE 1062 - Maximum requests in use
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE 1063 - Invalid muticast handle
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF 1064 - Invalid IP family preference
eQCWWAN_ERR_QMI_SESSION_INACTIVE 1065 - No tracking session has been started
eQCWWAN_ERR_QMI_SESSION_INVALID 1066 - Current session does not allow this operation
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP 1067 - Current tracking session not started by this QMI control point
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES 1068 - Device GPS service resources insufficient for request

eQCWWAN_ERR_QMI_DISABLED 1069 - Device GPS service disabled
eQCWWAN_ERR_QMI_INVALID_OPERATION 1070 - Invalid operation specified
eQCWWAN_ERR_QMI_INVALID_QMI_CMD 1071 - Invalid/unknown QMI command specified
eQCWWAN_ERR_QMI_TPDU_TYPE 1072 - Message contains TPDU type that cannot be read as raw message
eQCWWAN_ERR_QMI_SMSC_ADDR 1073 - The SMSC address specified is invalid
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE 1074 - Information element is unavailable at this point
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG 1075 - Segment size too large
eQCWWAN_ERR_QMI_SEGMENT_ORDER 1076 - Segment order is incorrect
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED 1077 - Bundling not supported
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE 1078 - The operation failed partially
eQCWWAN_ERR_QMI_POLICY_MISMATCH 1079 - Policy mismatch
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND 1080 - SIM file not found
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL 1081 - Extended internal error
eQCWWAN_ERR_QMI_ACCESS_DENIED 1082 - Access to a required entity is not available
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED 1083 - Selected operating mode is invalid with current hardware setting
eQCWWAN_ERR_QMI_ACK_NOT_SENT 1084 - ACK not sent
eQCWWAN_ERR_QMI_INJECT_TIMEOUT 1084 - Inject a timeout for the request
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE 1090 - Incompatible state
eQCWWAN_ERR_QMI_FDN_RESTRICT 1091 - FDN Restrict
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE 1092 - SUPS failure cause
eQCWWAN_ERR_QMI_NO_RADIO 1093 - No Radio
eQCWWAN_ERR_QMI_NOT_SUPPORTED 1094 - Not Supported
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION 1095 - No Subscription
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED 1096 - Card call control failed
eQCWWAN_ERR_QMI_NETWORK_ABORTED 1097 - Network Aborted
eQCWWAN_ERR_QMI_MSG_BLOCKED 1098 - Open Error
eQCWWAN_ERR_QMI_MAX Error - End of QMI specific defines
eQCWWAN_ERR_SWICM_START Vendor defines - Connection Manager error codes
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED 0xE001 - The API is yet to be implemented
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED 0xE002 - The service is not supported
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED 0xE003 - The client is not supported
eQCWWAN_ERR_SWICM_TIMEOUT 0xE004 - API Timeout
eQCWWAN_ERR_SWICM_SOCKET_IN_USE 0xE005 - The communication socket is in use
eQCWWAN_ERR_SWICM_AM_VERS_ERROR 0xE006 - SLQS API and SDK version mismatch
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS 0xE007 - Failed to kill SDK process
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS 0xE008 - Call in progress
eQCWWAN_ERR_SWICM_V4DWN_V6DWN 0xE009 - IPV4 and IPV6 is down
eQCWWAN_ERR_SWICM_V4DWN_V6UP 0xE00A - IPV4 is down and IPV6 is up
eQCWWAN_ERR_SWICM_V4UP_V6DWN 0xE00B - IPV4 is up and IPV6 is down
eQCWWAN_ERR_SWICM_V4UP_V6UP 0xE00C - IPV4 and IPV6 is up
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID 0xE00D - Invalid V4 Session ID
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID 0xE00E - Invalid V4 Session ID
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID 0xE00F - Invalid V6 Session ID

eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS 0xE010 - No available Session Manager slots for additional data sessions

eQCWWAN_ERR_SWICM_END 0xE011 - End of connection manager specific codes

eQCWWAN_ERR_SWISMS_START Vendor defines - SMS Error codes

eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG 0xE101 - SMS message length is long

eQCWWAN_ERR_SWISMS_MSG_CORRUPTED 0xE102 - The SMS message is corrupted (encoding wrong)

eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED 0xE103 - The SMS number is corrupted (incorrect number)

eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND 0xE104 - The SMS bearer data is not available

eQCWWAN_ERR_SWISM_END

eQCWWAN_ERR_SWIIM_START Vendor defines - Image Management error codes

eQCWWAN_ERR_SWIIM_INVALID_PATH 0xE801 - Invalid directory path

eQCWWAN_ERR_SWIIM_OPENING_DIR 0xE802 - Unable to open the directory

eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND 0xE803 - No Firmware image present in the path

eQCWWAN_ERR_SWIIM_OPENING_FILE 0xE804 - Unable to open the file

eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE 0xE805 - Firmware image is corrupted

eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED 0xE806 - No Firmware image download needed

eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL 0xE807 - Firmware update failed

eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH 0xE808 - Update success but pri/fw preference mismatch

eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS 0xE809 - Update successful

eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE 0xE80A - Enter Download Mode

eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE 0xE80B - File transfer to modem complete

eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT 0xE80C - Wait for modem to reboot

eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE 0xE80D - Invalid Crash State for Firmware Download

eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE 0xE80E - Same as current active image

eQCWWAN_ERR_SWIIM_END

eQCWWAN_ERR_SWIDCS_START Vendor defines - Device Connectivity error codes

eQCWWAN_ERR_SWIDCS_IOCTL_ERR 0xE901 - IO Control error

eQCWWAN_ERR_SWIDCS_FILEIO_ERR 0xE902 - file open/read/write error

eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND 0xE903 - The device is not found

eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED 0xE904 - Application is disconnected from SDK

eQCWWAN_ERR_SWIDCS_END

eQCWWAN_ERR_QMI_CAT_START QMI errors related to CAT

eQCWWAN_ERR_QMI_EVENT_REG_FAILED 62441 - CAT event registration failed

eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP 62442 - Invalid terminal response

eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD 62443 - Invalid envelope command

eQCWWAN_ERR_QMI_CARD_BUSY_RSP 62444 - Card busy response for envelope command

eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE 62445 - Envelope command failure

eQCWWAN_ERR_QMI_CAT_END

eQCWWAN_ERR_NULL_TLV

eQCWWAN_ERR_QMI_WIDTH 0xFFFF - Not an error, represent the end of QMI errors

9.21.1.2 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_SLQSQosSmiReadApnExtraParams_t](#)
- struct [unpack_qos_SLQSQosSmiReadApnExtraParams_t](#)

- struct [pack_qos_SLQSQosSwiReadDataStats_t](#)
- struct [unpack_QosFlowStat_t](#)
- struct [unpack_qos_SLQSQosSwiReadDataStats_t](#)
- struct [unpack_qos_SLQSSetQosNWStatusCallback_ind_t](#)
- struct [unpack_qos_SLQSSetQosStatusCallback_ind_t](#)
- struct [unpack_qos_SLQSSetQosPriEventCallback_ind_t](#)
- struct [pack_qos_SLQSSetQosEventCallback_t](#)
- struct [unpack_qos_QosFlowInfoState_t](#)
- struct [unpack_qos_dataRate_t](#)
- struct [unpack_qos_tokenBucket_t](#)
- struct [unpack_qos_pktErrRate_t](#)
- struct [unpack_qos_swiQosFlow_t](#)
- struct [unpack_qos_IPv4Addr_t](#)
- struct [unpack_qos_Tos_t](#)
- struct [unpack_qos_IPv6Addr_t](#)
- struct [unpack_qos_IPv6TrafCls_t](#)
- struct [unpack_qos_Port_t](#)
- struct [unpack_qos_swiQosFilter_t](#)
- struct [unpack_qos_QosFlowInfo_t](#)
- struct [unpack_qos_SLQSSetQosEventCallback_ind_t](#)

Macros

- `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`
- `#define LIBPACK_MAX_QOS_FILTERS 25`
- `#define LIBPACK_MAX_QOS_FLOWS 8`

Functions

- int [pack_qos_SLQSQosGetNetworkStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_qos_SLQSQosGetNetworkStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSQosGetNetworkStatus_t](#) *pOutput)
- int [pack_qos_SLQSQosSwiReadApnExtraParams](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_qos_SLQSQosSwiReadApnExtraParams_t](#) reqParam)
- int [unpack_qos_SLQSQosSwiReadApnExtraParams](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSQosSwiReadApnExtraParams_t](#) *pOutput)
- int [pack_qos_SLQSQosSwiReadDataStats](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_qos_SLQSQosSwiReadDataStats_t](#) reqParam)
- int [unpack_qos_SLQSQosSwiReadDataStats](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSQosSwiReadDataStats_t](#) *pOutput)
- int [unpack_qos_SLQSSetQosNWStatusCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosNWStatusCallback_ind_t](#) *pOutput)
- int [unpack_qos_SLQSSetQosStatusCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosStatusCallback_ind_t](#) *pOutput)
- int [unpack_qos_SLQSSetQosPriEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosPriEventCallback_ind_t](#) *pOutput)
- int [pack_qos_SLQSSetQosEventCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_qos_SLQSSetQosEventCallback_t](#) reqParam)
- int [unpack_qos_SLQSSetQosEventCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_qos_SLQSSetQosEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_qos_SLQSSetQosEventCallback_ind_t](#) *pOutput)

9.22.1 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](#) 2048
- #define [MAX_SMS_LIST_SIZE](#) 255
- #define [MAX_MS_TRANSFER_ROUTE_MSG](#) 256
- #define [MAX_MSE_TWS_MSG](#) 1254
- #define [MAX_MSC_ADDRESS_SIZE](#) 256
- #define [MAX_CDMA_ENC_MO_TXT_MSG_SIZE](#) 255

Typedefs

- typedef struct [sMSMTMessage](#) [sMSMTMessageInfo](#)
- typedef struct [sMSTransferRouteMTMessage](#) [sMSTransferRouteMTMessageInfo](#)
- typedef struct [sMSMessageMode](#) [sMSMessageModeInfo](#)
- typedef struct [sMSEtwsMessage](#) [sMSEtwsMessageInfo](#)
- typedef struct [sMSEtwsPlmn](#) [sMSEtwsPlmnInfo](#)
- typedef struct [sMSCAddress](#) [sMSCAddressInfo](#)
- typedef struct [sMSOnIMS](#) [sMSOnIMSInfo](#)

Enumerations

- enum [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](#) *pOutput)
- int [unpack_sms_SetNewSMSCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_sms_SetNewSMSCallback_ind_t](#) *pOutput)
- int [unpack_sms_SLQSWmsMemoryFullCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_sms_SLQSWmsMemoryFullCallBack_ind_t](#) *pOutput)

9.23.1 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 2048`

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 **sMSEtwsPlmn** **sMSEtwsPlmnInfo**

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** **sSMSMessageModeInfo**

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 eqmiCbkJSetStatus

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

<i>in, out</i>	<i>pCtx</i>	qmi request context
<i>out</i>	<i>pReqBuf</i>	qmi request buffer
<i>out</i>	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_XXX error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_XXX error values

9.23.4.2 `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_SLQSDDeleteSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SLQSDDeleteSMS_t * *reqParam*)

delete sms pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.4 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_SLQSModifySMSStatus (uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSModifySMSStatus_t * *pOutput*)

modify sms status unpack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.23.4.14 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_SLQSOMADMSelectSelection_t](#)
- struct [pack_swioma_SLQSOMADMGetSessionInfo_t](#)
- struct [unpack_swioma_SLQSOMADMGetSessionInfo_t](#)
- struct [unpack_omaDmFotaTlv_t](#)
- struct [unpack_omaDmConfigTlv_t](#)
- struct [unpack_omaDmNotificationsTlv_t](#)
- struct [unpack_swioma_SLQSOMADMAAlertCallback_ind_t](#)

Macros

- #define [LIBPACK_MAX_SWIOMA_STR_LEN](#) 255

Functions

- int [pack_swioma_SLQSOMADMStartSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swioma_SLQSOMADMStartSession_t](#) reqParam)
- int [unpack_swioma_SLQSOMADMStartSession](#) (uint8_t *pResp, uint16_t respLen, [unpack_swioma_SLQSOMADMStartSession_t](#) *pOutput)

- int [pack_swima_SLQSOMADMCancelSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMCancelSession_t](#) reqParam)
- int [unpack_swima_SLQSOMADMCancelSession](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_swima_SLQSOMADMGetSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_swima_SLQSOMADMGetSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_swima_SLQSOMADMGetSettings_t](#) *pOutput)
- int [pack_swima_SLQSOMADMSetSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMSetSettings_t](#) reqParam)
- int [unpack_swima_SLQSOMADMSetSettings](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_swima_SLQSOMADMSendSelection](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMSendSelection_t](#) reqParam)
- int [unpack_swima_SLQSOMADMSendSelection](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_swima_SLQSOMADMGetSessionInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_swima_SLQSOMADMGetSessionInfo_t](#) reqParam)
- int [unpack_swima_SLQSOMADMGetSessionInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_swima_SLQSOMADMGetSessionInfo_t](#) *pOutput)
- int [pack_swima_SLQSOMADMAAlertCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_swima_SLQSOMADMAAlertCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_swima_SLQSOMADMAAlertCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_swima_SLQSOMADMAAlertCallback_ind_t](#) *pOutput)

9.26.1 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_swioma_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_swima_SLQSOMADMStartSession_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

Timeout: 20 seconds

9.26.2.8 `int unpack_swima_SLQSOMADMAAlertCallback (uint8_t * pResp, uint16_t respLen)`

Function to unpack response of QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_swioma_SLQSOMADMAAlertCallback_ind_t for more information

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.26.2.10 int unpack_swioma_SLQSOMADMCancelSession (uint8_t * *pResp*, uint16_t *respLen*)

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_swima_SLQSOMADMGetSessionInfo (uint8_t * *pResp*, uint16_t *respLen*,
unpack_swima_SLQSOMADMGetSessionInfo_t * *pOutput*)

Function to unpack information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_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_swima_SLQSOMADMGetSettings (uint8_t * *pResp*, uint16_t *respLen*,
unpack_swima_SLQSOMADMGetSettings_t * *pOutput*)

Function to unpack OMA-DM get settings response from modem This maps to SLQSOMADMGetSettings2

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_swioma_SLQSOMADMSetSettings (uint8_t * *pResp*, uint16_t *respLen*)

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> • 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_SLQSUIMEventRegister_t](#)
- struct [unpack_uim_SLQSUIMEventRegister_t](#)
- struct [appStats](#)
- struct [slotInf](#)
- struct [unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t](#)
- struct [slot_t](#)
- struct [slots_t](#)
- struct [unpack_uim_SLQSUIMGetSlotsStatus_t](#)
- struct [pack_uim_SLQSUIMSwitchSlot_t](#)
- struct [unpack_uim_SetUimSlotStatusChangeCallback_ind_t](#)

Macros

- [#define UIM_UINT8_MAX_STRING_SZ 255](#)
- [#define UIM_MAX_DESCRIPTION_LENGTH 255](#)
- [#define UIM_MAX_NO_OF_SLOTS 5](#)
- [#define UIM_MAX_NO_OF_APPLICATIONS 10](#)
- [#define MAX_NO_OF_SLOTS 5](#)
- [#define MAX_NO_OF_APPLICATIONS 10](#)
- [#define MAX_DESCRIPTION_LENGTH 255](#)
- [#define MAX_SLOTS_STATUS 255](#)
- [#define MAX_ICCID_LENGTH 255](#)

Functions

- int [pack_uim_GetCardStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int [unpack_uim_GetCardStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_GetCardStatus_t](#) *pOutput)
- int [pack_uim_VerifyPin](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_VerifyPin_t](#) *reqArg)
- int [unpack_uim_VerifyPin](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_VerifyPin_t](#) *pOutput)
- int [pack_uim_UnblockPin](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_UnblockPin_t](#) *reqArg)
- int [unpack_uim_UnblockPin](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_UnblockPin_t](#) *pOutput)
- int [pack_uim_SetPinProtection](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SetPinProtection_t](#) *reqArg)
- int [unpack_uim_SetPinProtection](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SetPinProtection_t](#) *pOutput)
- int [pack_uim_ChangePin](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_ChangePin_t](#) *reqArg)
- int [unpack_uim_ChangePin](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_ChangePin_t](#) *pOutput)
- int [pack_uim_ReadTransparent](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_ReadTransparent_t](#) *reqArg)
- int [unpack_uim_ReadTransparent](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_ReadTransparent_t](#) *pOutput)
- int [pack_uim_SLQSUIMEventRegister](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIMEventRegister_t](#) *reqArg)
- int [unpack_uim_SLQSUIMEventRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIMEventRegister_t](#) *pOutput)

- int [unpack_uim_SLQSUIMSetStatusChangeCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIMSetStatusChangeCallback_ind_t](#) *pOutput)
- int [pack_uim_SLQSUIMGetSlotsStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_uim_SLQSUIMGetSlotsStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIMGetSlotsStatus_t](#) *pOutput)
- int [pack_uim_SLQSUIMSwitchSlot](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIMSwitchSlot_t](#) *reqArg)
- int [unpack_uim_SLQSUIMSwitchSlot](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_uim_SetUimSlotStatusChangeCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SetUimSlotStatusChangeCallback_ind_t](#) *pOutput)

9.28.1 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>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

9.28.2.2 int [pack_uim_GetCardStatus](#) ([pack_qmi_t](#) * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg*)

Get Card Status pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.3 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>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

9.28.2.4 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_SLQSUIMEventRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIMEventRegister_t * reqArg)`

UIM Status Change callback enable pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.6 `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_SLQSUISSwitchSlot (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUISSwitchSlot_t * reqArg)`

switch slot pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.8 `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_SLQSUIMEventRegister to subscribe

9.28.2.15 int unpack_uim_SLQSUIMEventRegister (uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMEvent-
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_SLQSUIMGetSlotsStatus (uint8_t * *pResp*, uint16_t *respLen*,
unpack_uim_SLQSUIMGetSlotsStatus_t * *pOutput*)

get slot status unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.28.2.17 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_SLQSSStartDataSession_t](#)
- struct [unpack_wds_SLQSSStartDataSession_t](#)
- struct [unpack_wds_SLQSSetPacketSrvStatusCallback_t](#)
- struct [pack_wds_SLQSSStopDataSession_t](#)
- struct [wds_ProfileIdentifier](#)
- struct [wds_GPRSQoS](#)
- struct [wds_PCSCFIPv4ServerAddressList](#)
- struct [wds_PCSCFFQDNAddress](#)
- struct [wds_PCSCFFQDNAddressList](#)

- struct [wds_Domain](#)
- struct [wds_DomainNameList](#)
- struct [wds_IPV6AddressInfo](#)
- struct [wds_IPV6GWAddressInfo](#)
- struct [unpack_wds_SLQSGetRuntimeSettings_t](#)
- struct [wds_currNetworkInfo](#)
- struct [unpack_wds_SLQSSetWdsEventCallback_ind_t](#)
- struct [pack_wds_SLQSSetWdsEventCallback_t](#)
- struct [pack_wds_SLQSGetRuntimeSettings_t](#)
- struct [wds_UMTSMInQoS](#)
- struct [LibPackprofile_3GPP](#)
- struct [LibPackprofile_3GPP2](#)
- union [wds_profileInfo](#)
- struct [pack_wds_SLQSCreateProfile_t](#)
- struct [PackCreateProfileOut](#)
- struct [unpack_wds_SLQSCreateProfile_t](#)
- struct [pack_wds_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 [rmTrasnferStaticsReq](#)
- struct [pack_wds_RMSetTransferStatistics_t](#)
- struct [unpack_wds_RMSetTransferStatistics_t](#)
- struct [pack_wds_SetMobileIPProfile_t](#)
- struct [unpack_wds_SetMobileIPProfile_t](#)
- struct [pack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#)
- struct [ipv6AddressInfo](#)

- struct [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#)
- struct [transferStatInd](#)
- struct [pack_wds_SLQSGetDUNCallInfo_t](#)
- struct [connectionStatus](#)
- struct [dunchannelRate](#)
- struct [unpack_wds_SLQSGetDUNCallInfo_t](#)
- struct [qmiWDSDataBearerTechnology](#)
- struct [unpack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [unpack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [pack_wds_SetDefaultProfileNum_t](#)
- struct [pack_wds_GetDefaultProfileNum_t](#)
- struct [unpack_wds_GetDefaultProfileNum_t](#)
- struct [wdsDhcpv4ProfileId](#)
- struct [wdsDhcpv4HwConfig](#)
- struct [wdsDhcpv4Option](#)
- struct [wdsDhcpv4OptionList](#)
- struct [pack_wds_SLQSSetDHCPv4ClientConfig_t](#)
- struct [unpack_wds_SLQSSetDHCPv4ClientConfig_t](#)

Macros

- [#define IPV6_ADDRESS_ARRAY_SIZE 8](#)
- [#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24](#)
- [#define PACK_WDS_IPV4 4](#)
- [#define PACK_WDS_IPV6 6](#)

Typedefs

- typedef union [unpackWdsProfileParam](#) [UnpackQmiProfileInfo](#)

Functions

- int [pack_wds_SLQSStartDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSStartDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSStartDataSession](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSStartDataSession_t](#) *pOutput)
- int [unpack_wds_SLQSSetPacketSrvStatusCallback](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetPacketSrvStatusCallback_t](#) *pOutput)
- int [pack_wds_SLQSStopDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSStopDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSStopDataSession](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_wds_SLQSGetRuntimeSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetRuntimeSettings_t](#) *pOutput)
- int [unpack_wds_SLQSSetWdsEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetWdsEventCallback_ind_t](#) *pOutput)
- int [unpack_wds_SLQSSetWdsEventCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_SLQSSetWdsEventCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSetWdsEventCallback_t](#) *reqArg)
- int [pack_wds_SLQSGetRuntimeSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetRuntimeSettings_t](#) *reqArg)
- int [pack_wds_SLQSCreateProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSCreateProfile_t](#) *reqArg)

- [int unpack_wds_SLQSCreateProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSCreateProfile_t](#) *pOutput)
- [int pack_wds_SLQSModifyProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSModifyProfile_t](#) *reqArg)
- [int unpack_wds_SLQSModifyProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSModifyProfile_t](#) *pOutput)
- [int pack_wds_SLQSGetProfileSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetProfileSettings_t](#) *reqArg)
- [int unpack_wds_SLQSGetProfileSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetProfileSettings_t](#) *pOutput)
- [int pack_wds_GetSessionState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_wds_GetSessionState](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetSessionState_t](#) *pOutput)
- [int pack_wds_GetDefaultProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDefaultProfile_t](#) *reqParam)
- [int unpack_wds_GetDefaultProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDefaultProfile_t](#) *pOutput)
- [int pack_wds_GetConnectionRate](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_wds_GetConnectionRate](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetConnectionRate_t](#) *pOutput)
- [int pack_wds_GetPacketStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetPacketStatus_t](#) *reqParam)
- [int unpack_wds_GetPacketStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetPacketStatus_t](#) *pOutput)
- [int pack_wds_GetSessionDuration](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetSessionDuration_t](#) *reqParam)
- [int unpack_wds_GetSessionDuration](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetSessionDuration_t](#) *pOutput)
- [int pack_wds_GetDormancyState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDormancyState_t](#) *reqParam)
- [int unpack_wds_GetDormancyState](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDormancyState_t](#) *pOutput)
- [int pack_wds_SLQSDeleteProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSDeleteProfile_t](#) *reqParam)
- [int unpack_wds_SLQSDeleteProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSDeleteProfile_t](#) *pOutput)
- [int pack_wds_SetDefaultProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetDefaultProfile_t](#) *reqParam)
- [int unpack_wds_SetDefaultProfile](#) (uint8_t *pResp, uint16_t respLen)
- [int pack_wds_SLQSGet3GPPConfigItem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- [int unpack_wds_SLQSGet3GPPConfigItem](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGet3GPPConfigItem_t](#) *pOutput)
- [int pack_wds_SLQSSet3GPPConfigItem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSet3GPPConfigItem_t](#) *reqParam)
- [int unpack_wds_SLQSSet3GPPConfigItem](#) (uint8_t *pResp, uint16_t respLen)
- [int pack_wds_GetMobileIP](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetMobileIP_t](#) *pReqParam)
- [int unpack_wds_GetMobileIP](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetMobileIP_t](#) *pOutput)
- [int pack_wds_GetMobileIPProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetMobileIPProfile_t](#) *reqParam)
- [int unpack_wds_GetMobileIPProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetMobileIPProfile_t](#) *pOutput)
- [int pack_wds_SLQSGetCurrDataSystemStat](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetCurrDataSystemStat_t](#) *pReqParam)
- [int unpack_wds_SLQSGetCurrDataSystemStat](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetCurrDataSystemStat_t](#) *pOutput)

- int [pack_wds_GetLastMobileIPError](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetLastMobileIPError_t](#) *pReqParam)
- int [unpack_wds_GetLastMobileIPError](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetLastMobileIPError_t](#) *pOutput)
- int [pack_wds_RMSetTransferStatistics](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_RMSetTransferStatistics_t](#) *reqParam)
- int [unpack_wds_RMSetTransferStatistics](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_RMSetTransferStatistics_t](#) *pOutput)
- int [pack_wds_SetMobileIPProfile](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetMobileIPProfile_t](#) *reqParam)
- int [unpack_wds_SetMobileIPProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SetMobileIPProfile_t](#) *pOutput)
- int [pack_wds_SLQSWdsSviPDPRuntimeSettings](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSWdsSviPDPRuntimeSettings_t](#) *reqParam)
- int [unpack_wds_SLQSWdsSviPDPRuntimeSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSWdsSviPDPRuntimeSettings_t](#) *pOutput)
- int [pack_wds_SLQSGetDUNCallInfo](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetDUNCallInfo_t](#) *reqParam)
- int [unpack_wds_SLQSGetDUNCallInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetDUNCallInfo_t](#) *pOutput)
- int [pack_wds_SLQSGetDataBearerTechnology](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetDataBearerTechnology_t](#) *pReqParam)
- int [unpack_wds_SLQSGetDataBearerTechnology](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetDataBearerTechnology_t](#) *pOutput)
- int [pack_wds_SLQSSetIPFamilyPreference](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSetIPFamilyPreference_t](#) *pReqParam)
- int [unpack_wds_SLQSSetIPFamilyPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetIPFamilyPreference_t](#) *pOutput)
- int [pack_wds_SetDefaultProfileNum](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetDefaultProfileNum_t](#) *pReqParam)
- int [unpack_wds_SetDefaultProfileNum](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_GetDefaultProfileNum](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDefaultProfileNum_t](#) *pReqParam)
- int [unpack_wds_GetDefaultProfileNum](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDefaultProfileNum_t](#) *pOutput)
- int [pack_wds_SLQSSetDHCPv4ClientConfig](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSetDHCPv4ClientConfig_t](#) *pReq)
- int [unpack_wds_SLQSSetDHCPv4ClientConfig](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetDHCPv4ClientConfig_t](#) *pOutput)

9.29.1 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	pCtx	qmi request context
out	pReq	qmi request buffer
out	pLen	qmi request length
out	reqParam	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.15 int pack_wds_SLQSCreateProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSCreateProfile_t * reqArg)

Parameters

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	reqParam	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

Note

PDN Specific: Yes

9.29.3.16 `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_SLQSSGetDHCIPv4ClientConfig (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSSGetDHCIPv4ClientConfig_t * *pReq*)

get DHCPv4 Client Config pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReq</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.28 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_GetLastMobileIPError_t * *pOutput*)

get current data system unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.36 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_SLQSSGetDhCpV4ClientConfig (uint8_t * *pResp*, uint16_t *respLen*,
unpack_wds_SLQSSGetDhCpV4ClientConfig_t * *pOutput*)

get DHCPv4 Client Config unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.60 int unpack_wds_SLQSStartDataSession (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSStartData-
Session_t * *pOutput*)

start data session unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

See [qmerrno.h](#) for eQCWWAN_xxx error values

9.29.3.61 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, 227](#)
 - [unpack_wds_SLQSGet3GPPConfigItem_t, 376](#)
- [AAASPI](#)
 - [unpack_wds_GetMobileIPProfile_t, 372](#)
- [AAASState](#)
 - [unpack_wds_GetMobileIPProfile_t, 372](#)
- [AMSSString](#)
 - [unpack_dms_GetFirmwareRevision_t, 263](#)
 - [unpack_dms_GetFirmwareRevisions_t, 264](#)
- [APNName](#)
 - [unpack_wds_SLQSGetRuntimeSettings_t, 380](#)
- [accolc](#)
 - [pack_nas_SetACCOLC_t, 184](#)
- [ackIndicator](#)
 - [sMSTransferRouteMTMessage, 241](#)
- [acqOrdeLen](#)
 - [nas_acqOrderPref, 78](#)
- [activationStatus](#)
 - [dms_ActivationStatusTlv, 29](#)
- [ActivationStatusTlv](#)
 - [unpack_dms_SetEventReport_ind_t, 270](#)
- [activeBandClass](#)
 - [nas_RFInfoTlv, 135](#)
 - [RFBandInfoElements, 233](#)
- [activeChannel](#)
 - [nas_RFInfoTlv, 135](#)
 - [RFBandInfoElements, 233](#)
- [ActiveTechPref](#)
 - [unpack_nas_GetNetworkPreference_t, 301](#)
- [addr](#)
 - [unpack_qos_IPv4Addr_t, 333](#)
 - [unpack_qos_IPv6Addr_t, 333](#)
- [address](#)
 - [unpack_wds_GetMobileIPProfile_t, 373](#)
- [aid](#)
 - [uim_sessionInformation, 252](#)
 - [uim_UIMSessionInformation, 256](#)
- [aidLength](#)
 - [appStats, 25](#)
 - [uim_appStatus, 245](#)
 - [uim_sessionInformation, 252](#)
 - [uim_UIMSessionInformation, 256](#)
- [aidVal](#)
 - [appStats, 25](#)
 - [uim_appStatus, 245](#)
- [aidingIndicatorMask](#)
 - [loc_sensorDataUsage, 74](#)
- [alertmsg](#)
 - [unpack_omaDmConfigTlv_t, 329](#)
- [alertmsglength](#)
 - [unpack_omaDmConfigTlv_t, 329](#)
- [ambr_dl](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
- [ambr_dl_ext](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
- [ambr_dl_ext2](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
- [ambr_ul](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
- [ambr_ul_ext](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
- [ambr_ul_ext2](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
- [amssSize](#)
 - [unpack_dms_GetFirmwareRevision_t, 263](#)
 - [unpack_dms_GetFirmwareRevisions_t, 264](#)
- [apdoxypages.c, 401](#)
- [apnId](#)
 - [pack_qos_SLQSQosSwiReadApnExtraParams_t, 201](#)
 - [pack_qos_SLQSQosSwiReadDataStats_t, 201](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 338](#)
 - [unpack_qos_SLQSQosSwiReadDataStats_t, 340](#)
- [apnName](#)
 - [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 386](#)
- [apnname](#)
 - [unpack_wds_GetDefaultProfile_t, 370](#)
- [apnsize](#)
 - [unpack_wds_GetDefaultProfile_t, 370](#)
- [appNameLength](#)
 - [loc_LocApplicationInfo, 73](#)
- [appProviderLength](#)
 - [loc_LocApplicationInfo, 73](#)
- [appState](#)
 - [appStats, 25](#)
 - [uim_appStatus, 245](#)
- [appStats, 23](#)
 - [aidLength, 25](#)
 - [aidVal, 25](#)

- appState, 25
- appType, 25
- persoFeature, 25
- persoRetries, 25
- persoState, 25
- persoUnblockRetries, 25
- pin1Retries, 26
- pin1State, 26
- pin2Retries, 26
- pin2State, 26
- puk1Retries, 26
- puk2Retries, 26
- univPin, 26
- AppStatus
 - slotInf, 236
 - uim_slotInfo, 255
- appType
 - appStats, 25
 - uim_appStatus, 245
- appVersionLength
 - loc_LocApplicationInfo, 73
- appVersionValid
 - loc_LocApplicationInfo, 73
- Application
 - unpack_nas_GetCDMANetworkParameters_t, 299
- appversion_str
 - unpack_dms_GetFirmwareInfo_t, 263
- arfcn
 - nas_GERANInfo, 97
 - nas_gsmCellInfo, 99
- auth
 - unpack_wds_GetDefaultProfile_t, 370
- Authentication
 - unpack_wds_SLQSGetRuntimeSettings_t, 380
- authentication
 - pack_wds_SetDefaultProfile_t, 220
- Autosdm
 - unpack_swima_SLQSOMADMGetSettings_t, 362
- avgPeriod
 - nas_LTESigRptConfig, 119
- bEnable
 - pack_nas_SLQSSetSignalStrengthsCallback_t, 195
- bForceDownload
 - pack_fms_SetImagesPreference_t, 177
- bICCID
 - slot_t, 234
- bICCIDLength
 - slot_t, 234
- bLogicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, 215
 - slot_t, 234
- bNumberOfPhySlots
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 366
- BOOL
 - SwiDataTypes.h, 543
- BPTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- bResetStatistics
 - rmTrasferStaticsReq, 233
- BYTE
 - SwiDataTypes.h, 543
- band
 - nas_LTEInfo, 112
- band1900
 - nas_gsmCellInfo, 99
- band_pref
 - NASBandPreferenceTlv, 160
- BandCapability
 - unpack_dms_GetBandCapability_t, 258
- bandCapability
 - unpack_dms_SLQSGetBandCapability_t, 276
- bandwidth
 - nas_LTEInfo, 112
- baseId
 - nas_CDMAInfo, 82
 - nas_CDMA SysInfo, 86
- baseLat
 - nas_CDMAInfo, 82
 - nas_CDMA SysInfo, 86
- baseLong
 - nas_CDMAInfo, 82
 - nas_CDMA SysInfo, 86
- BasestationID
 - unpack_nas_SLQSGetServingSystem_t, 310
- BasestationLatitude
 - unpack_nas_SLQSGetServingSystem_t, 311
- BasestationLongitude
 - unpack_nas_SLQSGetServingSystem_t, 311
- BearerID
 - unpack_qos_QosFlowInfo_t, 336
- bearerID
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 382
- bearerId
 - unpack_QosFlowStat_t, 352
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 386
- bootSize
 - unpack_dms_GetFirmwareRevisions_t, 264
- BootString
 - unpack_dms_GetFirmwareRevisions_t, 264
- bootversion_str
 - unpack_dms_GetFirmwareInfo_t, 263
- Broadcast
 - unpack_nas_GetCDMANetworkParameters_t, 299
- bsInfoValid
 - nas_CDMA SysInfo, 86
- bsPRev
 - nas_CDMA SysInfo, 86
- bsPRevValid
 - nas_CDMA SysInfo, 86
- bsic
 - nas_GERANInfo, 97
- bsicId

- nas_gsmCellInfo, 99
- bucketSz
 - unpack_qos_tokenBucket_t, 350
- buildID
 - FMSImageIdElement, 35
 - image_info_t, 37
- buildIDLen
 - image_info_t, 37
- buildIDLength
 - FMSImageIdElement, 35
- buildId
 - FMSImageElement, 34
- buildIdLength
 - FMSImageElement, 34
- CDMA_P_Rev
 - unpack_nas_SLQSGetServingSystem_t, 311
- CDMAECIOThreshListLen
 - nas_CDMAECIOThresh, 81
- CDMARSSIThreshListLen
 - nas_CDMARSSIThresh, 83
- CDMASSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 321
- CDMASystemInfoExt
 - unpack_nas_SLQSGetServingSystem_t, 311
- CHAR
 - SwiDataTypes.h, 543
- CQIValueCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, 325
- CQIValueCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, 325
- CSDomain
 - unpack_nas_GetServingNetwork_t, 302
- CallBarStatus
 - unpack_nas_SLQSGetServingSystem_t, 311
- callDuration
 - unpack_wds_GetSessionDuration_t, 374
- callEndReason
 - unpack_wds_SLQSGetDUNCallInfo_t, 378
- cardState
 - slotInf, 236
 - uim_slotInfo, 255
- carrier
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 278
- carrier_str
 - unpack_dms_GetFirmwareInfo_t, 263
- CarrierImage_t, 26
 - m_FwBuildId, 27
 - m_FwImageld, 27
 - m_PriBuildId, 27
 - m_PriImageld, 27
 - m_nCarrierId, 27
 - m_nFolderId, 27
 - m_nStorage, 27
- ccsSupported
 - nas_CDMASysInfo, 86
- ccsSupportedValid
 - nas_CDMASysInfo, 86
- cdmaSSInfo, 27
 - ecio, 27
 - rsi, 27
- cdmaSysIdValid
 - nas_CDMASysInfo, 86
- cell_resel_priority
 - nas_infoInterFreq, 109
- cellBroadcastCap
 - nas_AddSysInfo, 79
- CellID
 - unpack_nas_SLQSGetServingSystem_t, 311
- cellID
 - nas_GERANInfo, 97
 - nas_UMTSInfo, 151
- cellId
 - nas_GSMSysInfo, 103
 - nas_LTESysInfo, 122
 - nas_WCDMASysInfo, 160
- cellIdValid
 - nas_gsmCellInfo, 99
 - nas_GSMSysInfo, 103
 - nas_LTESysInfo, 122
 - nas_WCDMASysInfo, 160
- cellInterFreqParams
 - nas_infoInterFreq, 109
- cellsTDD
 - nas_umtsLTENbrCell, 153
- CellParams
 - nas_LTEInfoIntrafreq, 115
- cellReselPriority
 - nas_lteGsmCellInfo, 110
 - nas_LTEInfoIntrafreq, 115
 - nas_lteWcdmaCellInfo, 124
- cells_len
 - nas_infoInterFreq, 109
 - nas_lteGsmCellInfo, 110
- cellsLen
 - nas_LTEInfoIntrafreq, 115
 - nas_lteWcdmaCellInfo, 124
- chaddr
 - wdsDhcpv4HwConfig, 397
- chaddrLen
 - wdsDhcpv4HwConfig, 397
- changePIN
 - pack_uim_ChangePin_t, 211
- channelRate
 - unpack_wds_SLQSGetDUNCallInfo_t, 378
- common.h
 - eCTL, 403
 - eDMS, 403
 - eIND, 404
 - eLOC, 403
 - eLOG_DEBUG, 403
 - eLOG_FATAL, 403
 - eLOG_INFO, 403
 - eLOG_WARN, 403
 - eNAS, 403
 - eQOS, 403
 - eREQ, 404

- eRSP, [404](#)
- eSMS, [403](#)
- eSWILOC, [404](#)
- eSWIOMA, [404](#)
- eTIMEOUT_10_S, [404](#)
- eTIMEOUT_20_S, [404](#)
- eTIMEOUT_2_S, [404](#)
- eTIMEOUT_300_S, [404](#)
- eTIMEOUT_30_S, [404](#)
- eTIMEOUT_5_S, [404](#)
- eTIMEOUT_60_S, [404](#)
- eTIMEOUT_8_S, [404](#)
- eTIMEOUT_DEFAULT, [404](#)
- eTMD, [404](#)
- eUIM, [403](#)
- eWDS, [403](#)
- common.h, [401](#)
 - eLOG_LEVEL, [403](#)
 - eQMI_SVC, [403](#)
 - eTimeout, [404](#)
 - fill_pack_ctx, [404](#)
 - fill_sdu_hdr, [404](#)
 - get_version, [404](#)
 - glog, [405](#)
 - gloglvl, [405](#)
 - helper_get_resp_ctx, [404](#)
 - helper_get_xid, [405](#)
 - helper_set_log_func, [405](#)
 - helper_set_log_lvl, [405](#)
 - libpack_GetVersion, [405](#)
 - libpack_log, [405](#)
 - logger, [403](#)
 - MINREQBKLEN, [403](#)
 - MSGID_AND_LEN, [403](#)
 - MSGID_DONT_CARE, [403](#)
 - msgtype, [404](#)
 - SDU_HDR_LEN, [403](#)
 - UNUSEDPARAM, [403](#)
 - unpack_result_code_only, [405](#)
- commonInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, [323](#)
- ConcSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, [311](#)
- conn_status
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [382](#)
- connectionStatus, [27](#)
 - MDMCallDuration, [28](#)
 - MDMConnStatus, [28](#)
 - unpack_wds_GetSessionState_t, [375](#)
 - unpack_wds_SLQSGetDUNCallInfo_t, [378](#)
- content
 - uim_readResult, [250](#)
- contentLen
 - uim_readResult, [250](#)
- contextId
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [230](#)
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [386](#)
- contextType
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [230](#)
- cpich_ecno
 - nas_wcdmaCellInfo, [155](#)
- cpich_rscp
 - nas_wcdmaCellInfo, [155](#)
- crashAction
 - pack_dms_SetCrashAction_t, [172](#)
- csAttachState
 - nas_servSystem, [139](#)
 - NASServingSystemInfo, [170](#)
- csBarStatus
 - nas_CallBarringSysInfo, [80](#)
 - nas_callBarStatus, [81](#)
- cur_carr_name
 - unpack_dms_GetFirmwareInfo_t, [263](#)
- cur_carr_rev
 - unpack_dms_GetFirmwareInfo_t, [263](#)
- curDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, [377](#)
- curProfile
 - pack_wds_SLQSModifyProfile_t, [226](#)
 - UnPackGetProfileSettingOut, [387](#)
- CurrChanRxRate
 - dunchannelRate, [33](#)
- CurrChanTxRate
 - dunchannelRate, [33](#)
- currDBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- currNWInfo
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- currNetworkInfo, [28](#)
 - NetworkType, [28](#)
 - RATMask, [28](#)
 - SOMask, [28](#)
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [377](#)
- currentChannelRXRate
 - unpack_wds_GetConnectionRate_t, [369](#)
- currentChannelTXRate
 - unpack_wds_GetConnectionRate_t, [369](#)
- currentDataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, [228](#)
- currentNetwork
 - qmiWDSDataBearerTechnology, [232](#)
- CurrentPLMN
 - unpack_nas_SLQSGetServingSystem_t, [311](#)
- cust_attr
 - DMScustSettingInfo, [30](#)
- cust_id
 - DMScustSettingInfo, [30](#)
 - DMSgetCustomInput, [32](#)

- pack_dms_GetCustFeaturesV2_t, 171
- pack_dms_SetCustFeaturesV2_t, 173
- cust_value
 - DMScustSettingInfo, 30
 - pack_dms_SetCustFeaturesV2_t, 173
- custSetting
 - DMScustSettingList, 31
- CustomSCP
 - unpack_nas_GetCDMANetworkParameters_t, 299
- dBTEchAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- dBTEchnology
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- DHCPRelayEnabled
 - pack_dms_SetCustFeature_t, 172
 - unpack_dms_GetCustFeature_t, 259
- DMS_IMGDETAILS_LEN
 - dms.h, 409
- DMS_PM_FACTORY
 - dms.h, 410
- DMS_PM_LOW
 - dms.h, 410
- DMS_PM_OFFLINE
 - dms.h, 410
- DMS_PM_ONLINE
 - dms.h, 410
- DMS_PM_RESET
 - dms.h, 410
- DMS_PM_SHUT_DOWN
 - dms.h, 410
- DMScustSettingInfo, 30
 - cust_attr, 30
 - cust_id, 30
 - cust_value, 30
 - id_length, 30
 - value_length, 30
- DMScustSettingList, 30
 - custSetting, 31
 - list_type, 31
 - num_instances, 31
- DMSgetCustomFeatureV2, 31
 - pCustSettingInfo, 32
 - pCustSettingList, 32
 - pGetCustomInput, 32
- DMSgetCustomInput, 32
 - cust_id, 32
 - list_type, 32
- DTMInd
 - unpack_nas_SLQSGetServingSystem_t, 311
- data
 - sMSCAddress, 236
 - sMSEtwsMessage, 237
 - sMSTransferRouteMTMessage, 241
- data_buf
 - NASOTAMessageTlv, 163
- data_len
 - NASOTAMessageTlv, 163
- dataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, 228
- dataBearerMask
 - unpack_wds_SLQSGetDataBearerTechnology_t, 377
- dataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, 378
- dataCapabilities
 - nas_dataSrvCapabilities, 92
- dataCapabilitiesLen
 - nas_dataSrvCapabilities, 92
- DataCaps
 - unpack_nas_GetServingNetwork_t, 302
 - unpack_nas_GetServingNetworkCapabilities_t, 303
- dataCaps
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 304
- DataCapsLen
 - unpack_nas_GetServingNetwork_t, 302
 - unpack_nas_GetServingNetworkCapabilities_t, 303
- dataCapsSize
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 304
- DataRate
 - unpack_qos_swiQosFlow_t, 349
- dataRateMax
 - unpack_qos_dataRate_t, 332
- dataServiceCaCapability
 - unpack_dms_GetDeviceCapabilities_t, 261
- DataServiceCapability
 - unpack_dms_GetDeviceCap_t, 260
- DataSrvCapabilities
 - unpack_nas_SLQSGetServingSystem_t, 311
- dataSysStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- dataSystemStatus
 - pack_wds_SLQSSetWdsEventCallback_t, 228
- Date
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 360
- DateLength
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 360
- day
 - nas_timeInfo, 149
 - nas_UniversalTime, 154
- dayLtSavingAdj
 - nas_timeInfo, 149
- dayOfWeek
 - nas_timeInfo, 149
 - nas_UniversalTime, 154
- daylightSavings
 - nas_qaQmi3Gpp2TimeZone, 132
- defaultPDNEnabled

- pack_wds_SLQSSet3GPPConfigItem_t, 227
- unpack_wds_SLQSGet3GPPConfigItem_t, 376
- DefaultRoamInd
 - unpack_nas_SLQSGetServingSystem_t, 311
- delayClass
 - LibPackGPRSRequestedQoS, 39
 - wds_GPRSQoS, 391
- deliveryErrSDU
 - LibPackUMTSQoS, 66
 - wds_UMTSMInQoS, 396
- description
 - unpack_omaDmFotaTlv_t, 331
- descriptionlength
 - unpack_omaDmFotaTlv_t, 331
- Description
 - nas_QmiNas3GppNetworkInfo, 133
- destPortRangeEnd
 - LibPackTFTIDParams, 63
- destPortRangeStart
 - LibPackTFTIDParams, 63
- DetailedSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, 311
- DevCrashState
 - unpack_dms_GetCrashAction_t, 258
- DisableIMSI
 - pack_dms_SetCustFeature_t, 172
 - unpack_dms_GetCustFeature_t, 259
- dl_bw_value
 - nas_PhyCaAggPcellInfo, 127
 - nas_PhyCaAggScellIDIBw, 128
 - nas_PhyCaAggScellInfo, 132
 - NASPhyCaAggPcellInfo, 164
 - NASPhyCaAggScellIDIBw, 164
 - NASPhyCaAggScellInfo, 166
- dms.h, 405
 - DMS_IMGDETAILS_LEN, 409
 - DMS_PM_FACTORY, 410
 - DMS_PM_LOW, 410
 - DMS_PM_OFFLINE, 410
 - DMS_PM_ONLINE, 410
 - DMS_PM_RESET, 410
 - DMS_PM_SHUT_DOWN, 410
 - MAX_BUILD_ID_LEN, 410
 - pack_dms_GetActivationState, 411
 - pack_dms_GetBandCapability, 411
 - pack_dms_GetCrashAction, 411
 - pack_dms_GetCustFeature, 412
 - pack_dms_GetCustFeaturesV2, 412
 - pack_dms_GetDeviceCap, 412
 - pack_dms_GetDeviceCapabilities, 412
 - pack_dms_GetDeviceHardwareRev, 413
 - pack_dms_GetDeviceMfr, 413
 - pack_dms_GetDeviceSerialNumbers, 413
 - pack_dms_GetFSN, 415
 - pack_dms_GetFirmwareInfo, 414
 - pack_dms_GetFirmwareRevision, 414
 - pack_dms_GetFirmwareRevisions, 414
 - pack_dms_GetHardwareRevision, 415
 - pack_dms_GetIMSI, 415
 - pack_dms_GetModelID, 416
 - pack_dms_GetNetworkTime, 416
 - pack_dms_GetPRLVersion, 417
 - pack_dms_GetPower, 416
 - pack_dms_GetSerialNumbers, 417
 - pack_dms_GetUSBComp, 418
 - pack_dms_GetVoiceNumber, 418
 - pack_dms_SLQSDmsSwiGetResetInfo, 421
 - pack_dms_SLQSDmsSwiIndicationRegister, 421
 - pack_dms_SLQSGetBandCapability, 422
 - pack_dms_SLQSSwiClearDyingGaspStatistics, 422
 - pack_dms_SLQSSwiGetDyingGaspCfg, 423
 - pack_dms_SLQSSwiGetDyingGaspStatistics, 423
 - pack_dms_SLQSSwiGetFirmwareCurr, 423
 - pack_dms_SLQSSwiGetFwUpdateStatus, 424
 - pack_dms_SLQSSwiSetDyingGaspCfg, 424
 - pack_dms_SetCrashAction, 418
 - pack_dms_SetCustFeature, 419
 - pack_dms_SetCustFeaturesV2, 419
 - pack_dms_SetEventReport, 420
 - pack_dms_SetFirmwarePreference, 420
 - pack_dms_SetPower, 420
 - pack_dms_SetUSBComp, 421
 - pack_dms_UIMGetICCID, 424
 - UNIQUE_ID_LEN, 410
 - unpack_dms_GetActivationState, 425
 - unpack_dms_GetBandCapability, 425
 - unpack_dms_GetCrashAction, 425
 - unpack_dms_GetCustFeature, 426
 - unpack_dms_GetCustFeaturesV2, 426
 - unpack_dms_GetDeviceCap, 426
 - unpack_dms_GetDeviceCapabilities, 427
 - unpack_dms_GetDeviceHardwareRev, 427
 - unpack_dms_GetDeviceMfr, 427
 - unpack_dms_GetDeviceSerialNumbers, 428
 - unpack_dms_GetFSN, 429
 - unpack_dms_GetFirmwareInfo, 428
 - unpack_dms_GetFirmwareRevision, 428
 - unpack_dms_GetFirmwareRevisions, 429
 - unpack_dms_GetHardwareRevision, 429
 - unpack_dms_GetIMSI, 430
 - unpack_dms_GetModelID, 430
 - unpack_dms_GetNetworkTime, 430
 - unpack_dms_GetPRLVersion, 431
 - unpack_dms_GetPower, 431
 - unpack_dms_GetSerialNumbers, 431
 - unpack_dms_GetUSBComp, 432
 - unpack_dms_GetVoiceNumber, 432
 - unpack_dms_SLQSDmsSwiGetResetInfo, 435
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 435
 - unpack_dms_SLQSDmsSwiIndicationRegister, 436
 - unpack_dms_SLQSGetBandCapability, 436
 - unpack_dms_SLQSSwiClearDyingGaspStatistics, 437

- unpack_dms_SLQSSwiGetDyingGaspCfg, [437](#)
- unpack_dms_SLQSSwiGetDyingGaspStatistics, [437](#)
- unpack_dms_SLQSSwiGetFirmwareCurr, [438](#)
- unpack_dms_SLQSSwiGetFwUpdateStatus, [438](#)
- unpack_dms_SLQSSwiSetDyingGaspCfg, [438](#)
- unpack_dms_SetCrashAction, [432](#)
- unpack_dms_SetCustFeature, [433](#)
- unpack_dms_SetCustFeaturesV2, [433](#)
- unpack_dms_SetEventReport, [433](#)
- unpack_dms_SetEventReport_ind, [434](#)
- unpack_dms_SetFirmwarePreference, [434](#)
- unpack_dms_SetPower, [434](#)
- unpack_dms_SetUSBComp, [435](#)
- unpack_dms_UIMGetICCID, [439](#)
- dms_ActivationStatusTlv, [28](#)
 - activationStatus, [29](#)
 - TlvPresent, [29](#)
- dms_OperatingModeTlv, [29](#)
 - operatingMode, [29](#)
 - TlvPresent, [29](#)
- domain
 - wds_DomainNameList, [390](#)
- domainLen
 - wds_Domain, [390](#)
- DomainList
 - unpack_wds_SLQSGetRuntimeSettings_t, [380](#)
- domainName
 - wds_Domain, [390](#)
- dormancyStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- dormancyState
 - unpack_wds_GetDormancyState_t, [371](#)
- dormancyStatus
 - pack_wds_SLQSSetWdsEventCallback_t, [228](#)
 - unpack_wds_SLQSGetDUNCallInfo_t, [378](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- dtmSupp
 - nas_GSMSysInfo, [103](#)
- dtmSuppValid
 - nas_GSMSysInfo, [103](#)
- dunchannelRate, [32](#)
 - CurrChanRxRate, [33](#)
 - CurrChanTxRate, [33](#)
 - MaxChanRxRate, [33](#)
 - MaxChanTxRate, [33](#)
- Duration
 - pack_nas_SetNetworkPreference_t, [184](#)
 - unpack_nas_GetNetworkPreference_t, [301](#)
- eCTL
 - common.h, [403](#)
- eDMS
 - common.h, [403](#)
- eIND
 - common.h, [404](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
 - nas.h, [459](#)
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
 - nas.h, [459](#)
- eLOC
 - common.h, [403](#)
- eLOG_DEBUG
 - common.h, [403](#)
- eLOG_FATAL
 - common.h, [403](#)
- eLOG_INFO
 - common.h, [403](#)
- eLOG_WARN
 - common.h, [403](#)
- eNAS
 - common.h, [403](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
 - nas.h, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
 - nas.h, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
 - nas.h, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
 - nas.h, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
 - nas.h, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
 - nas.h, [459](#)
- eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
 - nas.h, [459](#)
- eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
 - nas.h, [459](#)
- eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
 - nas.h, [459](#)
- eQCWWAN_ERR_API_MUTEX_TIMEOUT
 - qmerrno.h, [521](#)
- eQCWWAN_ERR_BUFFER_SZ
 - qmerrno.h, [520](#)
- eQCWWAN_ERR_CANCEL_OP

qmerrno.h, [521](#)
 eQCWWAN_ERR_DRIVER
 qmerrno.h, [521](#)
 eQCWWAN_ERR_ENUM_BEGIN
 qmerrno.h, [520](#)
 eQCWWAN_ERR_ENUM_END
 qmerrno.h, [521](#)
 eQCWWAN_ERR_FILE_COPY
 qmerrno.h, [520](#)
 eQCWWAN_ERR_FILE_OPEN
 qmerrno.h, [520](#)
 eQCWWAN_ERR_GENERAL
 qmerrno.h, [520](#)
 eQCWWAN_ERR_INTERNAL
 qmerrno.h, [520](#)
 eQCWWAN_ERR_INVALID_ARG
 qmerrno.h, [520](#)
 eQCWWAN_ERR_INVALID_DEVID
 qmerrno.h, [520](#)
 eQCWWAN_ERR_INVALID_FILE
 qmerrno.h, [520](#)
 eQCWWAN_ERR_INVALID_QMI_RSP
 qmerrno.h, [520](#)
 eQCWWAN_ERR_INVALID_XID
 qmerrno.h, [521](#)
 eQCWWAN_ERR_MALFORMED_QMI_RSP
 qmerrno.h, [520](#)
 eQCWWAN_ERR_MEMORY
 qmerrno.h, [520](#)
 eQCWWAN_ERR_MULTIPLE_DEVICES
 qmerrno.h, [521](#)
 eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
 qmerrno.h, [521](#)
 eQCWWAN_ERR_NO_CANCELABLE_OP
 qmerrno.h, [521](#)
 eQCWWAN_ERR_NO_CONNECTION
 qmerrno.h, [520](#)
 eQCWWAN_ERR_NO_DEVICE
 qmerrno.h, [520](#)
 eQCWWAN_ERR_NO_SIGNAL
 qmerrno.h, [521](#)
 eQCWWAN_ERR_NONE
 qmerrno.h, [520](#)
 eQCWWAN_ERR_NULL_TLV
 qmerrno.h, [524](#)
 eQCWWAN_ERR_OFFLINE
 qmerrno.h, [521](#)
 eQCWWAN_ERR_PDU_GENERATION
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_ABORTED
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_ACCESS_DENIED
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_ACK_NOT_SENT
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_ARG_TOO_LONG
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_CALL_FAILED
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_CARD_BUSY_RSP
 qmerrno.h, [524](#)
 eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_CAT_END
 qmerrno.h, [524](#)
 eQCWWAN_ERR_QMI_CAT_START
 qmerrno.h, [524](#)
 eQCWWAN_ERR_QMI_CAUSE_CODE
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_CONNECT
 qmerrno.h, [520](#)
 eQCWWAN_ERR_QMI_DEVICE_IN_USE
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_DEVICE_NOT_READY
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_DISABLED
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_ENCODING
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
 qmerrno.h, [524](#)
 eQCWWAN_ERR_QMI_EVENT_REG_FAILED
 qmerrno.h, [524](#)
 eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_FDN_RESTRICT
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_FLOW_SUSPENDED
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_GENERAL
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_IFACE
 qmerrno.h, [520](#)
 eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
 qmerrno.h, [522](#)
 eQCWWAN_ERR_QMI_INCORRECT_PIN
 qmerrno.h, [521](#)
 eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
 qmerrno.h, [523](#)
 eQCWWAN_ERR_QMI_INJECT_TIMEOUT

- qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCE-
S
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, [524](#)
- eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTI-
ON
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTIO-
N
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, [524](#)
- eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_MAX
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_I-
N_USE
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_-
USE
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAIL-
URE
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTE-
D
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPOR-
TED
qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL

- qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_REQ
 - qmerrno.h, [520](#)
- eQCWWAN_ERR_QMI_REQ_SCH
 - qmerrno.h, [520](#)
- eQCWWAN_ERR_QMI_REQ_TO
 - qmerrno.h, [520](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_RSP
 - qmerrno.h, [520](#)
- eQCWWAN_ERR_QMI_RSP_TO
 - qmerrno.h, [520](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
 - qmerrno.h, [521](#)
- eQCWWAN_ERR_QMI_UNKNOWN
 - qmerrno.h, [522](#)
- eQCWWAN_ERR_QMI_WIDTH
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_RESET
 - qmerrno.h, [521](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_END
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_START
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_TIMEOUT
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP
 - qmerrno.h, [523](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIDCS_END
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIDCS_START
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_END
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
 - qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
 - qmerrno.h, [524](#)

- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWIIM_START
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWISM_END
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, [524](#)
- eQCWWAN_ERR_SWISMS_START
qmerrno.h, [524](#)
- eQMI_LOC_SESS_STATUS_FAILURE
loc.h, [447](#)
- eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, [447](#)
- eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, [447](#)
- eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, [447](#)
- eQOS
common.h, [403](#)
- eREQ
common.h, [404](#)
- eRSP
common.h, [404](#)
- eSMS
common.h, [403](#)
- eSWILOC
common.h, [404](#)
- eSWIOMA
common.h, [404](#)
- eTIMEOUT_10_S
common.h, [404](#)
- eTIMEOUT_20_S
common.h, [404](#)
- eTIMEOUT_2_S
common.h, [404](#)
- eTIMEOUT_300_S
common.h, [404](#)
- eTIMEOUT_30_S
common.h, [404](#)
- eTIMEOUT_5_S
common.h, [404](#)
- eTIMEOUT_60_S
common.h, [404](#)
- eTIMEOUT_8_S
common.h, [404](#)
- eTIMEOUT_DEFAULT
common.h, [404](#)
- eTMD
common.h, [404](#)
- eUIM
common.h, [403](#)
- eWDS
common.h, [403](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_END
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL
qmerrno.h, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT
qmerrno.h, [525](#)

- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_LEN_INVALID
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_FAIL
 - qmerrno.h, 525
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END
 - qmerrno.h, 525
- eLOG_LEVEL
 - common.h, 403
- EMTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- eQCWWANError
 - qmerrno.h, 520
- eQMI_SVC
 - common.h, 403
- ESNString
 - unpack_dms_GetDeviceSerialNumbers_t, 262
- ETWSPLMNInfo
 - eTWSPLMNInfoTlv, 33
- eTWSPLMNInfoTlv, 33
 - ETWSPLMNInfo, 33
 - TlvPresent, 33
- ETWSPLMNTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 353
- ETWSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 353
- eTimeout
 - common.h, 404
- eValid
 - LibPackTFTIDParams, 63
- earfcn
 - nas_infoInterFreq, 109
 - nas_LTEInfoIntraFreq, 115
 - nas_umtsLTENbrCell, 153
- ecio
 - cdmaSSInfo, 27
 - hdrSSInfo, 37
 - nas_ecioListElement, 94
 - nas_UMTSInfo, 151
 - tdscdmaSigInfoExt, 241
- ecioDelta
 - nas_SLQSSignalStrengthsIndReq, 141
- ecioInfo
 - nas_SLQSSignalStrengthsInformation, 142
- ecioList
 - unpack_nas_SLQSSetSignalStrength_t, 312
- ecioListLen
 - unpack_nas_SLQSSetSignalStrength_t, 312
- ecioThresholdList
 - nas_SLQSSignalStrengthsIndReq, 141
- ecioThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, 141
- egprsSupp
 - nas_GSMSSysInfo, 103
- egprsSuppValid
 - nas_GSMSSysInfo, 103
- EmerMode
 - NASEmergencyModeTlv, 161
- emmConnState
 - nas_LTEInfo, 112
- emmState
 - nas_LTEInfo, 112
- emmSubState
 - nas_LTEInfo, 112
- enable
 - pack_qos_SLQSSetQosEventCallback_t, 202
- enabled
 - unpack_wds_GetMobileIPProfile_t, 373
- EncryptedPIN1
 - pack_uim_ChangePin_t, 211
 - pack_uim_SetPinProtection_t, 213
 - pack_uim_UnblockPin_t, 216
- engineState
 - unpack_loc_EngineState_Ind_t, 289
- eqmiCbkSetStatus
 - sms.h, 537
- error
 - unpack_wds_GetLastMobileIPError_t, 371
- errorRate
 - nas_errorRateListElement, 95
- errorRateInfo
 - nas_SLQSSignalStrengthsInformation, 142
- errorRateList
 - unpack_nas_SLQSSetSignalStrength_t, 312
- errorRateListLen
 - unpack_nas_SLQSSetSignalStrength_t, 312
- errorState
 - slotInf, 236
 - uim_slotInfo, 255
- esn
 - unpack_dms_GetSerialNumbers_t, 267
- esnSize
 - unpack_dms_GetDeviceSerialNumbers_t, 262
- EspSpi
 - unpack_qos_swiQosFilter_t, 345
- EtwsMessageInfo
 - sMSEtwsMessageTlv, 238
- event
 - unpack_qos_SLQSSetQosPriEventCallback_ind_t, 341

- unpack_qos_SLQSSetQosStatusCallback_ind_t, 342
- eventMask
 - pack_uim_SLQSUIEventRegister_t, 214
 - unpack_uim_SLQSUIEventRegister_t, 366
- eventRegister
 - pack_loc_EventRegister_t, 180
- eventType
 - unpack_swima_SLQSOMADMAAlertCallback_ind_t, 358
- executingImage
 - FMSImageIDEntries, 35
- exponent
 - unpack_qos_pktErrRate_t, 334
- ExtErrorCode
 - PackCreateProfileOut, 230
- extPowerState
 - pack_loc_SetExtPowerState_t, 180
- extendedErrorCode
 - unpack_wds_SLQSDeleteProfile_t, 376
- FLOAT
 - SwiDataTypes.h, 543
- FMSImageElement, 33
 - buildId, 34
 - buildIdLength, 34
 - imageId, 34
 - imageType, 34
- FMSImageIDEntries, 35
 - executingImage, 35
 - imageIDElement, 35
 - imageIDSize, 35
 - imageType, 36
 - maxImages, 36
- FMSImageIDElement, 34
 - buildID, 35
 - buildIDLength, 35
 - failureCount, 35
 - imageID, 35
 - storageIndex, 35
- FMSImageList, 36
 - imageIDEntries, 36
 - listSize, 36
- FMSPrefImageList, 36
 - listEntries, 37
 - listSize, 37
- FOTAUpdate
 - pack_swima_SLQSOMADMSetSettings_t, 210
 - unpack_swima_SLQSOMADMGetSettings_t, 362
- FOTAdownload
 - pack_swima_SLQSOMADMSetSettings_t, 210
 - unpack_swima_SLQSOMADMGetSettings_t, 362
- failureCount
 - FMSImageIDElement, 35
- family
 - pack_wds_GetDefaultProfileNum_t, 218
 - pack_wds_SetDefaultProfileNum_t, 220
- fileID
 - uim_fileInfo, 249
- fileIndex
 - pack_uim_ReadTransparent_t, 212
- fill_pack_ctx
 - common.h, 404
- fill_sdu_hdr
 - common.h, 404
- filterId
 - LibPackTFTIDParams, 63
- fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, 207
 - unpack_swiloc_SwiLocGetAutoStart_t, 357
- fix_rate_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 357
- fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, 207
 - unpack_swiloc_SwiLocGetAutoStart_t, 357
- fix_type_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 357
- flowLabel
 - LibPackTFTIDParams, 63
- fms.h, 439
 - GetValidFwPriCombinations, 440
 - pack_fms_GetImagesPreference, 441
 - pack_fms_GetStoredImages, 441
 - pack_fms_SetImagesPreference, 441
 - unpack_fms_GetImagesPreference, 441
 - unpack_fms_GetStoredImages, 442
 - unpack_fms_SetImagesPreference, 442
- Forbidden
 - nas_QmiNas3GppNetworkInfo, 133
- ForceRev0
 - unpack_nas_GetCDMANetworkParameters_t, 299
- format
 - sMSTransferRouteMTMessage, 241
- fqdnAddr
 - wds_PCSCFFQDNAddress, 392
- fqdnLen
 - wds_PCSCFFQDNAddress, 392
- freq
 - nas_PhyCaAggPcellInfo, 127
 - nas_PhyCaAggScellIndType, 129
 - nas_PhyCaAggScellInfo, 132
 - NASPhyCaAggPcellInfo, 164
 - NASPhyCaAggScellIndType, 165
 - NASPhyCaAggScellInfo, 166
- freqsLen
 - nas_LTEInfoInterfreq, 113
 - nas_LTEInfoNeighboringGSM, 116
 - nas_LTEInfoNeighboringWCDMA, 117
- function
 - pack_swiloc_SwiLocSetAutoStart_t, 207
 - unpack_swiloc_SwiLocGetAutoStart_t, 357
- function_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 357
- FwAutoCheck
 - unpack_swima_SLQSOMADMGetSettings_t, 362
- FwAvailability

- unpack_swioma_SLQSOMADMStartSession_t, 362
- fwloadsize
 - unpack_omaDmFotaTlv_t, 331
- fwloadComplete
 - unpack_omaDmFotaTlv_t, 331
- fwvers
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 278
- gDIBitRate
 - LibPackQoSClassID, 62
- GPRSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, 380
- GPSPMP
 - pack_dms_SetCustFeature_t, 172
 - unpack_dms_GetCustFeature_t, 259
- GPSSel
 - pack_dms_SetCustFeature_t, 172
 - unpack_dms_GetCustFeature_t, 259
- GSMRSSIThreshListLen
 - nas_GSMRSSIThresh, 100
- GSMSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 321
- gUIBitRate
 - LibPackQoSClassID, 62
- GWAOPTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- GWAcqOrderPref
 - NASGWAcqOrderPrefTlv, 161
- GWAddressV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 380
- geoSysIdx
 - nas_AddCDMASysInfo, 79
 - nas_AddSysInfo, 79
- geranArfcn
 - nas_geranInstInfo, 98
- geranBsicBcc
 - nas_geranInstInfo, 98
- geranBsicNcc
 - nas_geranInstInfo, 98
- geranInst
 - nas_UMTSInfo, 151
- GeranInstInfo
 - nas_UMTSInfo, 151
- geranRssi
 - nas_geranInstInfo, 98
- get_version
 - common.h, 404
- GetCustomFeatureV2
 - unpack_dms_GetCustFeaturesV2_t, 260
- GetValidFwPriCombinations
 - fms.h, 440
- globalCellId
 - nas_LTEInfoIntrafreq, 115
- glog
 - common.h, 405
- gloglvl
 - common.h, 405
- gnssSvUsedList
 - loc_svUsedforFix, 77
- gnssSvUsedList_len
 - loc_svUsedforFix, 77
- Gpp2TimeZone
 - unpack_nas_SLQSGetServingSystem_t, 311
- GppNetworkDSTAdjustment
 - unpack_nas_SLQSGetServingSystem_t, 311
- GppTimeZone
 - unpack_nas_SLQSGetServingSystem_t, 311
- GpsEnable
 - pack_dms_SetCustFeature_t, 172
 - unpack_dms_GetCustFeature_t, 259
- gpsTimeOfWeekMs
 - loc_gpsTime, 72
- gpsWeek
 - loc_gpsTime, 72
- grntDownlinkBitrate
 - LibPackUMTSQoS, 66
 - wds_UMTSMInQoS, 396
- grntUplinkBitrate
 - LibPackUMTSQoS, 66
 - wds_UMTSMInQoS, 396
- GsmCellInfo
 - nas_lteGsmCellInfo, 110
- gsmUmtsDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 194
- gsmUmtsUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 195
- guaranteedRate
 - unpack_qos_dataRate_t, 332
- gwAddressV6
 - wds_IPV6GWAddressInfo, 392
- gwV6PrefixLen
 - wds_IPV6GWAddressInfo, 392
- HASPI
 - unpack_wds_GetMobileIPProfile_t, 373
- HASState
 - unpack_wds_GetMobileIPProfile_t, 373
- HDOP
 - loc_precisionDilution, 74
- HDRECIOThreshListLen
 - nas_HDRECIOThresh, 104
- HDRIOThreshListLen
 - nas_HDRIOThresh, 105
- HDRRSSIThreshListLen
 - nas_HDRRSSIThresh, 105
- HDRSINRThreshListLen
 - nas_HDRSINRThreshold, 106
- HDRSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 321
- HardwareControlledMode
 - unpack_dms_GetPower_t, 267
- hdrActiveProt
 - nas_HDRSysInfo, 108
- hdrActiveProtValid
 - nas_HDRSysInfo, 108

- hdrHybrid
 - nas_detailSvcInfo, [94](#)
- HdrPersonality
 - unpack_nas_SLQSGetServingSystem_t, [311](#)
- hdrPersonality
 - nas_HDRSysInfo, [108](#)
 - NASServingSystemInfo, [170](#)
- hdrPersonalityValid
 - nas_HDRSysInfo, [108](#)
- hdrSSInfo, [37](#)
 - ecio, [37](#)
 - io, [37](#)
 - rsi, [37](#)
 - sinr, [37](#)
- hdrSrvStatus
 - nas_detailSvcInfo, [94](#)
- helper_get_resp_ctx
 - common.h, [404](#)
- helper_get_xid
 - common.h, [405](#)
- helper_set_log_func
 - common.h, [405](#)
- helper_set_log_lvl
 - common.h, [405](#)
- hotSwap
 - uim_hotSwapStatus, [250](#)
- hotSwapLength
 - uim_hotSwapStatus, [250](#)
- hour
 - nas_timeInfo, [149](#)
 - nas_UniversalTime, [154](#)
- hsCallStatus
 - nas_WCDMASysInfo, [160](#)
- hsCallStatusValid
 - nas_WCDMASysInfo, [160](#)
- hsInd
 - nas_WCDMASysInfo, [160](#)
- hsIndValid
 - nas_WCDMASysInfo, [160](#)
- hwType
 - wdsDhcpv4HwConfig, [397](#)
- hwVer
 - unpack_dms_GetHardwareRevision_t, [265](#)
- iLTEbandValue
 - nas_PhyCaAggPcellInfo, [127](#)
 - nas_PhyCaAggScellInfo, [132](#)
 - NASPhyCaAggPcellInfo, [164](#)
 - NASPhyCaAggScellInfo, [167](#)
- IMCNflag
 - unpack_wds_SLQSGetRuntimeSettings_t, [380](#)
- IMEIString
 - unpack_dms_GetDeviceSerialNumbers_t, [262](#)
- IMSInfo
 - sMSOnIMSTlv, [240](#)
- IMSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [353](#)
- INT32
 - SwiDataTypes.h, [543](#)
- INT8
 - SwiDataTypes.h, [543](#)
- IPAddressV6
 - ipv6AddressInfo, [38](#)
 - wds_IPV6AddressInfo, [391](#)
- IPFamSupport
 - pack_dms_SetCustFeature_t, [172](#)
 - unpack_dms_GetCustFeature_t, [259](#)
- IPFamilyPreference
 - pack_wds_SLQSSetIPFamilyPreference_t, [227](#)
 - unpack_wds_SLQSGetRuntimeSettings_t, [380](#)
- IPSECSPi
 - LibPackTFTIDParams, [63](#)
- IPv6AddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, [381](#)
- IPv6GWAddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, [381](#)
- IPv6PrefixLen
 - ipv6AddressInfo, [38](#)
 - wds_IPV6AddressInfo, [391](#)
- IPv4
 - unpack_wds_SLQSGetRuntimeSettings_t, [380](#)
- IPv4DstAddr
 - unpack_qos_swiQosFilter_t, [345](#)
- IPv4SrcAddr
 - unpack_qos_swiQosFilter_t, [345](#)
- IPv4Tos
 - unpack_qos_swiQosFilter_t, [345](#)
- IPv6DstAddr
 - unpack_qos_swiQosFilter_t, [345](#)
- IPv6Label
 - unpack_qos_swiQosFilter_t, [345](#)
- IPv6SrcAddr
 - unpack_qos_swiQosFilter_t, [345](#)
- IPv6TrafCls
 - unpack_qos_swiQosFilter_t, [345](#)
- Id
 - unpack_qos_swiQosFilter_t, [345](#)
- id
 - loc_BdsSV, [67](#)
 - loc_SV, [75](#)
 - nas_CSGID, [90](#)
 - unpack_qos_QosFlowInfoState_t, [337](#)
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, [342](#)
- id_length
 - DMScustSettingInfo, [30](#)
- image_info_t, [37](#)
 - buildID, [37](#)
 - buildIDLen, [37](#)
 - imageType, [37](#)
 - uniqueID, [37](#)
- imageID
 - FMSImageIDElement, [35](#)
- imageIDElement
 - FMSImageIDEntries, [35](#)
- imageIDEntries
 - FMSImageList, [36](#)

- imageIDSize
 - FMSImageIDEntries, [35](#)
- imageId
 - FMSImageElement, [34](#)
- imageList
 - unpack_fms_GetStoredImages_t, [282](#)
- ImageListSize
 - unpack_fms_GetImagesPreference_t, [281](#)
- imageListSize
 - pack_fms_SetImagesPreference_t, [177](#)
- imageType
 - FMSImageElement, [34](#)
 - FMSImageIDEntries, [36](#)
 - image_info_t, [37](#)
- ImageTypes
 - unpack_fms_SetImagesPreference_t, [282](#)
- ImageTypesSize
 - unpack_fms_SetImagesPreference_t, [282](#)
- imagelistSize
 - unpack_fms_GetStoredImages_t, [282](#)
- imei_no
 - unpack_dms_GetSerialNumbers_t, [267](#)
- imeiSize
 - unpack_dms_GetDeviceSerialNumbers_t, [262](#)
- imeiSvnSize
 - unpack_dms_GetDeviceSerialNumbers_t, [262](#)
- ImeiSvnString
 - unpack_dms_GetDeviceSerialNumbers_t, [262](#)
- imeisv_svn
 - unpack_dms_GetSerialNumbers_t, [267](#)
- imgType
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, [279](#)
- imsRegState
 - nas_CommInfo, [90](#)
- imsi
 - unpack_dms_GetIMSI_t, [265](#)
- imsi_11_12
 - nas_CDMASysInfoExt, [87](#)
- InUse
 - nas_QmiNas3GppNetworkInfo, [133](#)
- includes_pcs_digit
 - nas_QmisNasPcsDigit, [134](#)
- index
 - pack_wds_GetMobileIPProfile_t, [218](#)
 - pack_wds_SetDefaultProfileNum_t, [220](#)
 - pack_wds_SetMobileIPProfile_t, [221](#)
 - unpack_qos_swiQosFilter_t, [345](#)
 - unpack_qos_swiQosFlow_t, [349](#)
 - unpack_wds_GetDefaultProfileNum_t, [371](#)
- index1xPri
 - uim_cardStatus, [247](#)
- index1xSec
 - uim_cardStatus, [247](#)
- indexGwPri
 - uim_cardStatus, [247](#)
- indexGwSec
 - uim_cardStatus, [247](#)
- Info
 - unpack_nas_SLQSNasSwtOTAMessageCallback_ind_t, [324](#)
 - unpack_nas_SLQSSetSysSelectionPrefCallback_ind_t, [324](#)
- InfoInterfreq
 - nas_LTEInfoInterfreq, [113](#)
- insNmrCellInfo
 - nas_GERANInfo, [97](#)
- instancesSize
 - unpack_nas_GetRFInfo_t, [301](#)
- interval
 - pack_wds_SLQSSetWdsEventCallback_t, [228](#)
- Io
 - unpack_nas_SLQSGetSignalStrength_t, [312](#)
- io
 - hdrSSInfo, [37](#)
 - nas_SLQSSignalStrengthsInformation, [142](#)
- ioDelta
 - nas_SLQSSignalStrengthsIndReq, [141](#)
- ipAddress
 - pack_wds_SetDefaultProfile_t, [220](#)
- ipFamily
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [382](#)
- ipVersion
 - LibPackTFTIDParams, [63](#)
- ipaddr
 - unpack_wds_GetDefaultProfile_t, [370](#)
- ipaddrv6
 - unpack_wds_GetDefaultProfile_t, [370](#)
- ipv4Address
 - unpack_wds_SLQSWdsSwtPDPRuntimeSettings_t, [386](#)
- ipv4GWAddress
 - unpack_wds_SLQSWdsSwtPDPRuntimeSettings_t, [386](#)
- ipv6Address
 - unpack_wds_SLQSWdsSwtPDPRuntimeSettings_t, [386](#)
- ipv6AddressInfo, [38](#)
 - IPAddressV6, [38](#)
 - IPv6PrefixLen, [38](#)
- ipv6GWAddress
 - unpack_wds_SLQSWdsSwtPDPRuntimeSettings_t, [386](#)
- is856SysId
 - nas_HDRSysInfo, [108](#)
- is856SysIdValid
 - nas_HDRSysInfo, [108](#)
- is_DataRate_Available
 - unpack_qos_swiQosFlow_t, [349](#)
- is_EspSpi_Available
 - unpack_qos_swiQosFilter_t, [345](#)
- is_IPv4DstAddr_Available
 - unpack_qos_swiQosFilter_t, [345](#)
- is_IPv4SrcAddr_Available
 - unpack_qos_swiQosFilter_t, [345](#)
- is_IPv4Tos_Available

- unpack_qos_swiQosFilter_t, 345
- is_IPv6DstAddr_Available
 - unpack_qos_swiQosFilter_t, 345
- is_IPv6Label_Available
 - unpack_qos_swiQosFilter_t, 345
- is_IPv6SrcAddr_Available
 - unpack_qos_swiQosFilter_t, 345
- is_IPv6TrafCls_Available
 - unpack_qos_swiQosFilter_t, 345
- is_Id_Available
 - unpack_qos_swiQosFilter_t, 345
- is_Jitter_Available
 - unpack_qos_swiQosFlow_t, 349
- is_Latency_Available
 - unpack_qos_swiQosFlow_t, 349
- is_LteBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, 276
- is_LteQci_Available
 - unpack_qos_swiQosFlow_t, 349
- is_MaxAllowedPktSz_Available
 - unpack_qos_swiQosFlow_t, 349
- is_MinPolicedPktSz_Available
 - unpack_qos_swiQosFlow_t, 349
- is_NxtHdrProto_Available
 - unpack_qos_swiQosFilter_t, 345
- is_PktErrRate_Available
 - unpack_qos_swiQosFlow_t, 349
- is_Precedence_Available
 - unpack_qos_swiQosFilter_t, 345
- is_ProfileId3GPP2_Available
 - unpack_qos_swiQosFlow_t, 349
- is_RxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, 336
- is_TCPDstPort_Available
 - unpack_qos_swiQosFilter_t, 345
- is_TCPSrcPort_Available
 - unpack_qos_swiQosFilter_t, 345
- is_TdsBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, 276
- is-TokenBucket_Available
 - unpack_qos_swiQosFlow_t, 349
- is_TrafficClass_Available
 - unpack_qos_swiQosFlow_t, 349
- is_TranDstPort_Available
 - unpack_qos_swiQosFilter_t, 346
- is_TranSrcPort_Available
 - unpack_qos_swiQosFilter_t, 346
- is_TxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, 336
- is_UDPDstPort_Available
 - unpack_qos_swiQosFilter_t, 346
- is_UDPSrcPort_Available
 - unpack_qos_swiQosFilter_t, 346
- is_val_3GPP2Pri_Available
 - unpack_qos_swiQosFlow_t, 349
- is_val_3GPPImCn_Available
 - unpack_qos_swiQosFlow_t, 349
- is_val_3GPPResResidualBER_Available
 - unpack_qos_swiQosFlow_t, 349
- is_val_3GPPSigInd_Available
 - unpack_qos_swiQosFlow_t, 349
- is_val_3GPPTraHdlPri_Available
 - unpack_qos_swiQosFlow_t, 349
- isNewFlow
 - unpack_qos_QosFlowInfoState_t, 337
- isPrefDataPath
 - nas_GSMSrvStatusInfo, 101
 - nas_SrvStatusInfo, 143
- isSysForbidden
 - nas_detailSvcInfo, 94
 - nas_sysInfoCommon, 145
- isSysForbiddenValid
 - nas_sysInfoCommon, 145
- isSysPriMatch
 - nas_CDMASysInfo, 86
 - nas_HDRSysInfo, 108
- isSysPriMatchValid
 - nas_CDMASysInfo, 86
 - nas_HDRSysInfo, 108
- IsVoiceEnabled
 - pack_dms_SetCustFeature_t, 172
 - unpack_dms_GetCustFeature_t, 259
- Jitter
 - unpack_qos_swiQosFlow_t, 349
- LIBPACK_QMI_CBK_PARAM_NOCHANGE
 - sms.h, 537
- LIBPACK_QMI_CBK_PARAM_RESET
 - sms.h, 537
- LIBPACK_QMI_CBK_PARAM_SET
 - sms.h, 537
- LBPTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- LOCEVENTMASKGNSSSVINFO
 - loc.h, 445
- LOCEVENTMASKNMEA
 - loc.h, 446
- LOCEVENTMASKWIFIREQ
 - loc.h, 446
- LPCSTR
 - SwiDataTypes.h, 543
- LTEAttachProfileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 227
 - unpack_wds_SLQSGet3GPPConfigItem_t, 376
- LTEAttachProfileListLen
 - pack_wds_SLQSSet3GPPConfigItem_t, 227
 - unpack_wds_SLQSGet3GPPConfigItem_t, 377
- LTETBandPref
 - NASLTETBandPreferenceTlv, 162
- LTETCphyCAInfo
 - unpack_nas_SlqsGetLTETCphyCAInfo_t, 308
- LTERSRPThreshListLen
 - nas_LTERSRPThresh, 117
- LTERSRQThreshListLen
 - nas_LTERSRQThresh, 118
- LTERSSIThreshListLen

- nas_LTERSSIThresh, 119
- LTESNRThreshListLen
 - nas_LTESNRThreshold, 120
- LTESSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 321
- LTEWCDMACellInfo
 - nas_LTEInfoNeighboringWCDMA, 117
- Lac
 - unpack_nas_SLQSGetServingSystem_t, 311
- lac
 - nas_GERANInfo, 97
 - nas_GSMSSysInfo, 103
 - nas_LTESysInfo, 122
 - nas_UMTSInfo, 151
 - nas_WCDMASysInfo, 160
- lacValid
 - nas_GSMSSysInfo, 103
 - nas_LTESysInfo, 122
 - nas_WCDMASysInfo, 160
- lastCallDataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, 378
- lastCallDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, 378
- lastCallRXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, 379
- lastCallTXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, 379
- Latency
 - unpack_qos_swiQosFlow_t, 349
- leapSeconds
 - nas_qaQmi3Gpp2TimeZone, 132
- len
 - loc_BdsSVInfo, 68
 - loc_SVInfo, 76
 - unpack_nas_GetSignalStrengths_t, 303
- length
 - sMSCAddress, 237
 - sMSEtwSMessage, 238
 - sMSTransferRouteMTMessage, 241
 - uim_readTransparentInfo, 251
- LibPackGPRSRequestedQoS, 38
 - delayClass, 39
 - meanThroughputClass, 39
 - peakThroughputClass, 39
 - precedenceClass, 39
 - reliabilityClass, 39
- LibPackQosClassID, 61
 - gDIBitRate, 62
 - gUIBitRate, 62
 - maxDIBitRate, 62
 - maxUIBitRate, 62
 - QCI, 62
- LibPackTFTIDParams, 62
 - destPortRangeEnd, 63
 - destPortRangeStart, 63
 - eValid, 63
 - filterId, 63
 - flowLabel, 63
 - IPSECSPi, 63
 - ipVersion, 63
 - nextHeader, 64
 - pSourceIP, 64
 - sourceIPMask, 64
 - srcPortRangeEnd, 64
 - srcPortRangeStart, 64
 - tosMask, 64
- LibPackUMTSQoS, 64
 - deliveryErrSDU, 66
 - grntDownlinkBitrate, 66
 - grntUplinkBitrate, 66
 - maxDownlinkBitrate, 66
 - maxSDUSize, 66
 - maxUplinkBitrate, 66
 - qosDeliveryOrder, 66
 - resBerRatio, 66
 - sduErrorRatio, 66
 - trafficClass, 66
 - trafficPriority, 66
 - transferDelay, 66
- LibPackUMTSReqQoSsigInd, 66
 - SigInd, 67
 - UMTSReqQoS, 67
- LibPackprofile_3GPP, 50
 - pAPNClass, 54
 - pAPNDisabledFlag, 54
 - pAPNName, 54
 - pAPNnameSize, 54
 - pAddrAllocPref, 54
 - pAuthenticationPref, 55
 - pGPRSMinimumQoS, 55
 - pGPRSRequestedQoS, 55
 - pIPv4AddrPref, 55
 - pIPv6AddPref, 55
 - pImCnFlag, 55
 - pPDNInactivTimeout, 55
 - pPDPTtype, 55
 - pPassword, 55
 - pPasswordSize, 55
 - pPcscfAddrUsingDhcp, 55
 - pPcscfAddrUsingPCO, 55
 - pPdpAccessConFlag, 55
 - pPdpContext, 55
 - pPdpDataCompType, 55
 - pPdpHdrCompType, 55
 - pPriDNSIPv4AddPref, 55
 - pPriDNSIPv6addpref, 55
 - pPrimaryID, 55
 - pProfileName, 55
 - pProfileNameSize, 55
 - pQosClassID, 55
 - pSecDNSIPv4AddPref, 55
 - pSecDNSIPv6addpref, 55
 - pSecondaryFlag, 55
 - pTFTID1Params, 55
 - pTFTID2Params, 55

- pUMTSMinQoS, 55
- pUMTSMinQoSSigInd, 56
- pUMTSReqQoS, 56
- pUMTSReqQoSSigInd, 56
- pUsername, 56
- pUsernameSize, 56
- LibPackprofile_3GPP2, 56
 - pAPNClass3GPP2, 60
 - pAPNEnabled3GPP2, 60
 - pAllowLinger, 60
 - pApnString, 60
 - pApnStringSize, 60
 - pAppPriority, 60
 - pAppType, 60
 - pAuthPassword, 60
 - pAuthPassword_tSize, 60
 - pAuthProtocol, 60
 - pAuthRetryCount, 60
 - pAuthTimeout, 60
 - pDataMode, 60
 - pDataRate, 60
 - plpcpAckTimeout, 60
 - plpcpCreqRetryCount, 60
 - plsPcscfAddressNedded, 60
 - pLcpAckTimeout, 60
 - pLcpCreqRetryCount, 60
 - pNegoDnsSrvrPref, 61
 - pPDNInactivTimeout3GPP2, 61
 - pPdnType, 61
 - pPppSessCloseTimer1x, 61
 - pPppSessCloseTimerDO, 61
 - pPriV6DnsAddress, 61
 - pPrimaryV4DnsAddress, 61
 - pRATType, 61
 - pSecV6DnsAddress, 61
 - pSecondaryV4DnsAddress, 61
 - pUserId, 61
 - pUserIdSize, 61
- libpack_GetVersion
 - common.h, 405
- libpack_log
 - common.h, 405
- LibpackProfile3GPP, 39
 - pAPNClass, 43
 - pAPNDisabledFlag, 43
 - pAPNName, 43
 - pAPNnameSize, 43
 - pAddrAllocPref, 43
 - pAuthenticationPref, 43
 - pGPRSMinimumQoS, 43
 - pGPRSRequestedQoS, 43
 - plIPv4AddrPref, 44
 - plIPv6AddrPref, 44
 - plmCnFlag, 43
 - pPDNInactivTimeout, 44
 - pPDPTtype, 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, 44
 - pUMTSMinQoSSigInd, 44
 - pUMTSReqQoS, 44
 - pUMTSReqQoSSigInd, 44
 - pUsername, 44
 - pUsernameSize, 45
- LibpackProfile3GPP2, 45
 - pAPNClass3GPP2, 49
 - pAPNEnabled3GPP2, 49
 - pAllowLinger, 49
 - pApnString, 49
 - pApnStringSize, 49
 - pAppPriority, 49
 - pAppType, 49
 - pAuthPassword, 49
 - pAuthPasswordSize, 49
 - pAuthProtocol, 49
 - pAuthRetryCount, 49
 - pAuthTimeout, 49
 - pDataMode, 49
 - pDataRate, 49
 - plpcpAckTimeout, 49
 - plpcpCreqRetryCount, 49
 - plsPcscfAddressNedded, 49
 - pLcpAckTimeout, 49
 - pLcpCreqRetryCount, 49
 - pNegoDnsSrvrPref, 50
 - pPDNInactivTimeout3GPP2, 50
 - pPdnType, 50
 - pPppSessCloseTimer1x, 50
 - pPppSessCloseTimerDO, 50
 - pPriV6DnsAddress, 50
 - pPrimaryV4DnsAddress, 50
 - pRATType, 50
 - pSecV6DnsAddress, 50
 - pSecondaryV4DnsAddress, 50
 - pUserId, 50
 - pUserIdSize, 50
- list_type
 - DMScustSettingList, 31
 - DMSgetCustomInput, 32

- pack_dms_GetCustFeaturesV2_t, 171
- listEntries
 - FMSPrefImageList, 37
- listSize
 - FMSImageList, 36
 - FMSPrefImageList, 37
- loc.h
 - eQMI_LOC_SESS_STATUS_FAILURE, 447
 - eQMI_LOC_SESS_STATUS_IN_PROGRESS, 447
 - eQMI_LOC_SESS_STATUS_SUCCESS, 447
 - eQMI_LOC_SESS_STATUS_TIMEOUT, 447
- loc.h, 442
 - LOCEVENTMASKNMEA, 446
 - LOCEVENTMASKWIFIREQ, 446
 - pack_loc_DeleteAssistData, 447
 - pack_loc_EventRegister, 447
 - pack_loc_SLQSLOCGetBestAvailPos, 448
 - pack_loc_SetExtPowerState, 447
 - pack_loc_SetOperationMode, 448
 - pack_loc_Start, 448
 - pack_loc_Stop, 449
 - unpack_loc_BestAvailPos_Ind, 449
 - unpack_loc_DeleteAssistData, 449
 - unpack_loc_EngineState_Ind, 450
 - unpack_loc_EventRegister, 450
 - unpack_loc_PositionRpt_Ind, 450
 - unpack_loc_SLQSLOCGetBestAvailPos, 452
 - unpack_loc_SetExtPowerConfig_Ind, 451
 - unpack_loc_SetExtPowerState, 451
 - unpack_loc_SetOperationMode, 451
 - unpack_loc_Start, 452
 - unpack_loc_Stop, 452
- loc_BdsSV, 67
 - id, 67
 - mask, 67
- loc_BdsSVInfo, 67
 - len, 68
 - pSV, 68
- loc_CellDb, 68
 - mask, 68
- loc_ClkInfo, 69
 - mask, 70
- loc_GnssData, 70
 - mask, 71
- loc_LocApplicationInfo, 72
 - appNameLength, 73
 - appProviderLength, 73
 - appVersionLength, 73
 - appVersionValid, 73
 - pAppName, 73
 - pAppProvider, 73
 - pAppVersion, 73
- loc_SV, 75
 - id, 75
 - mask, 75
 - system, 75
- loc_SVInfo, 75
 - len, 76
 - pSV, 76
- loc_gpsTime, 71
 - gpsTimeOfWeekMs, 72
 - gpsWeek, 72
- loc_precisionDilution, 73
 - HDOP, 74
 - PDOP, 74
 - VDOP, 74
- loc_sensorDataUsage, 74
 - aidingIndicatorMask, 74
 - usageMask, 75
- loc_svUsedforFix, 76
 - gnssSvUsedList, 77
 - gnssSvUsedList_len, 77
- localTimeOffset
 - nas_qaQmi3Gpp2TimeZone, 132
- logString
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 279
- logger
 - common.h, 403
- longName
 - unpack_nas_SLQSGetPLMNName_t, 309
- longNameCI
 - unpack_nas_SLQSGetPLMNName_t, 309
- longNameEn
 - unpack_nas_SLQSGetPLMNName_t, 309
- longNameLen
 - unpack_nas_SLQSGetPLMNName_t, 309
- longNameSB
 - unpack_nas_SLQSGetPLMNName_t, 309
- LteBandCapability
 - unpack_dms_SLQSGetBandCapability_t, 276
- lteEmmDI
 - pack_nas_SLQSNasSwtOTAMessageCallback_t, 195
- lteEmmUI
 - pack_nas_SLQSNasSwtOTAMessageCallback_t, 195
- lteEsmDI
 - pack_nas_SLQSNasSwtOTAMessageCallback_t, 195
- lteEsmUI
 - pack_nas_SLQSNasSwtOTAMessageCallback_t, 195
- LteGsmCellInfo
 - nas_LTEInfoNeighboringGSM, 116
- LteQci
 - unpack_qos_swtQosFlow_t, 350
- lteRsrpDelta
 - nas_SLQSSignalStrengthsIndReq, 141
- lteRsrpinfo
 - nas_SLQSSignalStrengthsInformation, 142
- lteSSInfo, 77
 - rsrp, 77
 - rsrq, 77
 - rssr, 77
 - snr, 77

- lteSnrDelta
 - nas_SLQSSignalStrengthsIndReq, [141](#)
- lteSnrinfo
 - nas_SLQSSignalStrengthsInformation, [142](#)
- ltersrp
 - unpack_nas_SLQSGetSignalStrength_t, [312](#)
- ltesnr
 - unpack_nas_SLQSGetSignalStrength_t, [312](#)
- m_FwBuildId
 - CarrierImage_t, [27](#)
- m_FwImageld
 - CarrierImage_t, [27](#)
- m_PriBuildId
 - CarrierImage_t, [27](#)
- m_Prilmageld
 - CarrierImage_t, [27](#)
- m_nCarrierId
 - CarrierImage_t, [27](#)
- m_nFolderId
 - CarrierImage_t, [27](#)
- m_nStorage
 - CarrierImage_t, [27](#)
- MAX_BUILD_ID_LEN
 - dms.h, [410](#)
- MAX_ICCID_LENGTH
 - uim.h, [556](#)
- MAX_MSE_TWS_MSG
 - sms.h, [535](#)
- MAX_NO_OF_SLOTS
 - uim.h, [556](#)
- MAX_SLOTS_STATUS
 - uim.h, [556](#)
- MAX_SMS_LIST_SIZE
 - sms.h, [535](#)
- MCC
 - nas_CDMA SysInfo, [86](#)
 - nas_CDMA SysInfoExt, [87](#)
 - nas_currentPLMN, [91](#)
 - nas_GSM SysInfo, [103](#)
 - nas_LTE SysInfo, [122](#)
 - nas_QmiNas3GppNetworkInfo, [133](#)
 - nas_QmiNas3GppNetworkRAT, [134](#)
 - nas_QmisNasPcsDigit, [135](#)
 - nas_WCDMA SysInfo, [160](#)
 - unpack_nas_GetServingNetwork_t, [302](#)
- MDMCallDuration
 - connectionStatus, [28](#)
- MDMConnStatus
 - connectionStatus, [28](#)
- MEIDString
 - unpack_dms_GetDeviceSerialNumbers_t, [262](#)
- MIN
 - unpack_dms_GetVoiceNumber_t, [268](#)
- MINREQBKLEN
 - common.h, [403](#)
- MMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [353](#)
- MNC
 - nas_CDMA SysInfo, [86](#)
 - nas_currentPLMN, [91](#)
 - nas_GSM SysInfo, [103](#)
 - nas_LTE SysInfo, [123](#)
 - nas_QmiNas3GppNetworkInfo, [133](#)
 - nas_QmiNas3GppNetworkRAT, [134](#)
 - nas_QmisNasPcsDigit, [135](#)
 - nas_WCDMA SysInfo, [160](#)
 - unpack_nas_GetServingNetwork_t, [302](#)
- MPTlv
 - NASQmiCbkNasSystemSelPrefInd, [168](#)
- MSGID_AND_LEN
 - common.h, [403](#)
- MSGID_DONT_CARE
 - common.h, [403](#)
- MTMessageInfo
 - newMTMessageTlv, [170](#)
- Mask
 - pack_wds_SLQSGetDUNCallInfo_t, [223](#)
- mask
 - loc_BdsSV, [67](#)
 - loc_CellDb, [68](#)
 - loc_ClkInfo, [70](#)
 - loc_GnssData, [71](#)
 - loc_SV, [75](#)
 - unpack_qos_IPv6TrafCls_t, [334](#)
 - unpack_qos_Tos_t, [351](#)
- max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, [207](#)
 - unpack_swiloc_SwiLocGetAutoStart_t, [357](#)
- max_dist_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, [357](#)
- max_time
 - pack_swiloc_SwiLocSetAutoStart_t, [207](#)
 - unpack_swiloc_SwiLocGetAutoStart_t, [357](#)
- max_time_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, [357](#)
- MaxAllowedPktSz
 - unpack_qos_swiQosFlow_t, [350](#)
- MaxChanRxRate
 - dunchannelRate, [33](#)
- MaxChanTxRate
 - dunchannelRate, [33](#)
- maxChannelIRXRate
 - unpack_wds_GetConnectionRate_t, [369](#)
- maxChannelTXRate
 - unpack_wds_GetConnectionRate_t, [369](#)
- maxDIBitRate
 - LibPackQosClassID, [62](#)
- maxDownlinkBitrate
 - LibPackUMTSQoS, [66](#)
 - wds_UMTSMInQoS, [396](#)
- maxImages
 - FMSImageIDEntries, [36](#)
- MaxRXChannelRate
 - unpack_dms_GetDeviceCap_t, [260](#)
- maxRxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, [261](#)

- maxSDUSize
 - LibPackUMTSQoS, [66](#)
 - wds_UMTSMInQoS, [396](#)
- MaxTXChannelRate
 - unpack_dms_GetDeviceCap_t, [260](#)
- maxTxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, [261](#)
- maxUIBitRate
 - LibPackQoSClassID, [62](#)
- maxUplinkBitrate
 - LibPackUMTSQoS, [66](#)
 - wds_UMTSMInQoS, [396](#)
- mcc
 - nas_CSGID, [91](#)
 - nas_MNRInfo, [124](#)
 - nas_netSelectionPref, [125](#)
 - pack_nas_SLQSGetPLMNName_t, [185](#)
 - unpack_nas_GetHomeNetwork_t, [300](#)
- mdmCallDurationActive
 - unpack_wds_SLQSGetDUNCallInfo_t, [379](#)
- meanThroughputClass
 - LibPackGPRSRequestedQoS, [39](#)
 - wds_GPRSQoS, [391](#)
- meid
 - unpack_dms_GetSerialNumbers_t, [267](#)
- meidSize
 - unpack_dms_GetDeviceSerialNumbers_t, [262](#)
- message
 - unpack_sms_SLQSGetSMS_t, [354](#)
- message_type
 - NASOTAMessageTlv, [163](#)
- messageFailureCode
 - unpack_sms_SendSMS_t, [352](#)
- messageFormat
 - pack_sms_SendSMS_t, [202](#)
 - unpack_sms_SLQSGetSMS_t, [354](#)
- messageID
 - unpack_sms_SendSMS_t, [352](#)
- messageIndex
 - pack_sms_SLQSGetSMS_t, [204](#)
 - pack_sms_SLQSMModifySMSStatus_t, [206](#)
 - qmiSmsMessageList, [232](#)
 - sMSMTMessage, [239](#)
- messageList
 - unpack_sms_SLQSGetSMSList_t, [355](#)
- messageListSize
 - unpack_sms_SLQSGetSMSList_t, [355](#)
- messageMode
 - sMSMessageMode, [239](#)
 - unpack_sms_SLQSWmsMemoryFullCallback_ind_t, [355](#)
- MessageModelInfo
 - messageModeTlv, [78](#)
- messageModeTlv, [77](#)
 - MessageModelInfo, [78](#)
 - TlvPresent, [78](#)
- messageSize
 - pack_sms_SendSMS_t, [202](#)
- unpack_sms_SLQSGetSMS_t, [354](#)
- messageTag
 - pack_sms_SLQSMModifySMSStatus_t, [206](#)
 - qmiSmsMessageList, [232](#)
 - unpack_sms_SLQSGetSMS_t, [354](#)
- MinPolicedPktSz
 - unpack_qos_swiQoSFlow_t, [350](#)
- minSize
 - unpack_dms_GetVoiceNumber_t, [268](#)
- minute
 - nas_timeInfo, [149](#)
 - nas_UniversalTime, [154](#)
- mipMode
 - unpack_wds_GetMobileIP_t, [372](#)
- mipStatus
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- mipstatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- mnc
 - nas_CSGID, [91](#)
 - nas_MNRInfo, [125](#)
 - nas_netSelectionPref, [125](#)
 - pack_nas_SLQSGetPLMNName_t, [185](#)
 - unpack_nas_GetHomeNetwork_t, [300](#)
- mncPcsDigits
 - nas_CSGID, [91](#)
- mobileCountryCode
 - sMSEtwSPlmn, [238](#)
- mobileIP
 - pack_wds_SLQSSetWdsEventCallback_t, [228](#)
- mobileNetworkCode
 - sMSEtwSPlmn, [238](#)
- mode
 - pack_dms_SetEventReport_t, [173](#)
 - pack_dms_SetPower_t, [173](#)
 - pack_loc_SetOperationMode_t, [181](#)
- ModePref
 - NASModePreferenceTlv, [162](#)
- modelid
 - unpack_dms_GetModelID_t, [265](#)
- modelid_str
 - unpack_dms_GetFirmwareInfo_t, [263](#)
- modemMode
 - nas_CommInfo, [90](#)
- modemindex
 - pack_fms_SetImagesPreference_t, [177](#)
- month
 - nas_timeInfo, [149](#)
 - nas_UniversalTime, [154](#)
- msgid
 - pack_qmi_t, [200](#)
 - unpack_qmi_t, [332](#)
- msgtype
 - common.h, [404](#)
- Mtu
 - unpack_wds_SLQSGetRuntimeSettings_t, [381](#)

- multiplier
 - unpack_qos_pktErrRate_t, 334
- NAI
 - unpack_wds_GetMobileIPProfile_t, 373
- NAS_PLMN_LENGTH
 - nas.h, 458
- NASBandPreferenceTlv, 160
 - band_pref, 160
 - TlvPresent, 160
- NASEmergencyModeTlv, 161
 - EmerMode, 161
 - TlvPresent, 161
- NASGWAcqOrderPrefTlv, 161
 - GWAcqOrderPref, 161
 - TlvPresent, 161
- NASLTEBandPreferenceTlv, 162
 - LTEBandPref, 162
 - TlvPresent, 162
- NASLteNasReleaseInfoTlv, 162
 - nas_major, 162
 - nas_minor, 162
 - nas_release, 162
 - TlvPresent, 162
- NASModePreferenceTlv, 162
 - ModePref, 162
 - TlvPresent, 162
- NASNetSelPreferenceTlv, 162
 - NetSelPref, 163
 - TlvPresent, 163
- NASOTAMessageTlv, 163
 - data_buf, 163
 - data_len, 163
 - message_type, 163
 - TlvPresent, 163
- NASPRLPreferenceTlv, 167
 - PRLPref, 167
 - TlvPresent, 167
- NASPhyCaAggPcellInfo, 163
 - dl_bw_value, 164
 - freq, 164
 - iLTEbandValue, 164
 - pci, 164
 - TlvPresent, 164
- NASPhyCaAggScellIDBw, 164
 - dl_bw_value, 164
 - TlvPresent, 164
- NASPhyCaAggScellIndType, 165
 - freq, 165
 - pci, 166
 - scell_state, 166
 - TlvPresent, 166
- NASPhyCaAggScellIndex, 164
 - scell_idx, 165
 - TlvPresent, 165
- NASPhyCaAggScellInfo, 166
 - dl_bw_value, 166
 - freq, 166
 - iLTEbandValue, 167
 - pci, 167
 - scell_state, 167
 - TlvPresent, 167
- NASQmiCbkNasSwiOTAMessageInd, 167
 - nasRelInfoTlv, 167
 - otaMsgTlv, 167
 - timeTlv, 167
- NASQmiCbkNasSystemSelPrefInd, 167
 - BPTlv, 168
 - EMTlv, 168
 - GWAOPTlv, 168
 - LBPTlv, 168
 - MPTlv, 168
 - NSPTlv, 168
 - PRLPTlv, 168
 - RPTlv, 168
 - SDPTlv, 168
- NASRoamPreferenceTlv, 168
 - RoamPref, 168
 - TlvPresent, 168
- NASServDomainPrefTlv, 168
 - SrvDomainPref, 168
 - TlvPresent, 168
- NASServingSystemInfo, 168
 - csAttachState, 170
 - hdrPersonality, 170
 - psAttachState, 170
 - radioInterfaceList, 170
 - radioInterfaceNo, 170
 - registrationState, 170
 - selectedNetwork, 170
- NASTimeInfoTlv, 170
 - time, 170
 - TlvPresent, 170
- NSPTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- NWQoSStatus
 - unpack_qos_SLQSQosGetNetworkStatus_t, 337
- naiSize
 - unpack_wds_GetMobileIPProfile_t, 373
- Name
 - unpack_nas_GetServingNetwork_t, 302
- name
 - unpack_nas_GetHomeNetwork_t, 300
 - unpack_wds_GetDefaultProfile_t, 370
- nameSize
 - unpack_nas_GetServingNetwork_t, 302
- namelength
 - unpack_omaDmFotaTlv_t, 331
- namesize
 - unpack_wds_GetDefaultProfile_t, 370
- nas.h
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100, 459
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15, 459
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25, 459

- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6, [459](#)
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75, [459](#)
- eLIBPACK_NAS_LTE_CPHY_SELL_STATE_CONFIGURED_ACTIVATED, [459](#)
- eLIBPACK_NAS_LTE_CPHY_SELL_STATE_CONFIGURED_DEACTIVATED, [459](#)
- eLIBPACK_NAS_LTE_CPHY_SELL_STATE_DECONFIGURED, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_100, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, [459](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, [459](#)
- eNAS_LTE_CPHY_SELL_STATE_CONFIGURED_ACTIVATED_LITE, [459](#)
- eNAS_LTE_CPHY_SELL_STATE_CONFIGURED_DEACTIVATED_LITE, [459](#)
- eNAS_LTE_CPHY_SELL_STATE_DECONFIGURED_LITE, [459](#)
- nas.h, [453](#)
 - NAS_PLMN_LENGTH, [458](#)
 - pack_nas_GetACCOLC, [459](#)
 - pack_nas_GetANAAAAAuthenticationStatus, [460](#)
 - pack_nas_GetCDMANetworkParameters, [460](#)
 - pack_nas_GetHomeNetwork, [460](#)
 - pack_nas_GetNetworkPreference, [461](#)
 - pack_nas_GetRFInfo, [461](#)
 - pack_nas_GetServingNetwork, [461](#)
 - pack_nas_GetServingNetworkCapabilities, [461](#)
 - pack_nas_GetSignalStrengths, [462](#)
 - pack_nas_PerformNetworkScan, [462](#)
 - pack_nas_SLQSGetNetworkTime, [464](#)
 - pack_nas_SLQSGetPLMNName, [464](#)
 - pack_nas_SLQSGetServingSystem, [464](#)
 - pack_nas_SLQSGetSignalStrength, [464](#)
 - pack_nas_SLQSGetSysInfo, [465](#)
 - pack_nas_SLQSGetSysSelectionPref, [465](#)
 - pack_nas_SLQSInitiateNetworkRegistration, [465](#)
 - pack_nas_SLQSNasConfigSigInfo2, [466](#)
 - pack_nas_SLQSNasGetCellLocationInfo, [466](#)
 - pack_nas_SLQSNasGetSigInfo, [466](#)
 - pack_nas_SLQSNasIndicationRegisterExt, [467](#)
 - pack_nas_SLQSNasSwiModemStatus, [467](#)
 - pack_nas_SLQSNasSwiOTAMessageCallback, [467](#)
 - pack_nas_SLQSSetBandPreference, [468](#)
 - pack_nas_SLQSSetSignalStrengthsCallback, [468](#)
 - pack_nas_SLQSSetSysSelectionPref, [468](#)
 - pack_nas_SLQSSwiGetLteCQI, [469](#)
 - pack_nas_SetACCOLC, [462](#)
 - pack_nas_SetLURRejectCallback, [463](#)
 - pack_nas_SetNetworkPreference, [463](#)
 - pack_nas_SetRFInfoCallback, [463](#)
 - pack_nas_SLqsGetLTECphyCAInfo, [463](#)
 - unpack_nas_GetACCOLC, [469](#)
 - unpack_nas_GetANAAAAAuthenticationStatus, [469](#)
 - unpack_nas_GetCDMANetworkParameters, [470](#)
 - unpack_nas_GetHomeNetwork, [470](#)
 - unpack_nas_GetNetworkPreference, [470](#)
 - unpack_nas_GetRFInfo, [470](#)
 - unpack_nas_GetServingNetwork, [471](#)
 - unpack_nas_GetServingNetworkCapabilities, [471](#)
 - unpack_nas_GetSignalStrengths, [471](#)
 - unpack_nas_PerformNetworkScan, [472](#)
 - unpack_nas_SLQSGetNetworkTime, [474](#)
 - unpack_nas_SLQSGetPLMNName, [474](#)
 - unpack_nas_SLQSGetServingSystem, [475](#)
 - unpack_nas_SLQSGetSignalStrength, [475](#)
 - unpack_nas_SLQSGetSysInfo, [475](#)
 - unpack_nas_SLQSGetSysSelectionPref, [476](#)
 - unpack_nas_SLQSInitiateNetworkRegistration, [476](#)
 - unpack_nas_SLQSNasConfigSigInfo2, [476](#)
 - unpack_nas_SLQSNasGetCellLocationInfo, [476](#)
 - unpack_nas_SLQSNasGetSigInfo, [477](#)
 - unpack_nas_SLQSNasIndicationRegisterExt, [477](#)
 - unpack_nas_SLQSNasNetworkTimeCallback_ind, [477](#)
 - unpack_nas_SLQSNasSigInfoCallback_ind, [478](#)
 - unpack_nas_SLQSNasSwiModemStatus, [478](#)
 - unpack_nas_SLQSNasSwiOTAMessageCallback, [478](#)
 - unpack_nas_SLQSNasSwiOTAMessageCallback_ind, [479](#)
 - unpack_nas_SLQSNasSysInfoCallback_ind, [479](#)
 - unpack_nas_SLQSSetBandPreference, [479](#)
 - unpack_nas_SLQSSetSignalStrengthsCallback, [479](#)
 - unpack_nas_SLQSSetSysSelectionPref, [480](#)
 - unpack_nas_SLQSSetSysSelectionPrefCallback_ind, [480](#)
 - unpack_nas_SLQSSwiGetLteCQI, [480](#)
 - unpack_nas_SetACCOLC, [472](#)
 - unpack_nas_SetDataCapabilitiesCallback_ind, [472](#)
 - unpack_nas_SetEventReportInd, [473](#)
 - unpack_nas_SetLURRejectCallback, [473](#)
 - unpack_nas_SetNasLTECphyCAIndCallback_ind, [473](#)
 - unpack_nas_SetNetworkPreference, [473](#)
 - unpack_nas_SetRFInfoCallback, [473](#)
 - unpack_nas_SetRoamingIndicatorCallback_ind, [474](#)
 - unpack_nas_SetServingSystemCallback_ind, [474](#)
 - unpack_nas_SLqsGetLTECphyCAInfo, [474](#)
- nas_AddCDMASysInfo, [78](#)
 - geoSysIdx, [79](#)
 - regPrd, [79](#)
- nas_AddSysInfo, [79](#)
 - cellBroadcastCap, [79](#)
 - geoSysIdx, [79](#)

- nas_CDMAECIOThresh, 81
 - CDMAECIOThreshListLen, 81
 - pCDMAECIOThreshList, 81
- nas_CDMAInfo, 82
 - baseId, 82
 - baseLat, 82
 - baseLong, 82
 - nid, 82
 - refpn, 83
 - sid, 83
- nas_CDMARSSIThresh, 83
 - CDMARSSIThreshListLen, 83
 - pCDMARSSIThreshList, 83
- nas_CDMA SysInfo, 83
 - baseId, 86
 - baseLat, 86
 - baseLong, 86
 - bsInfoValid, 86
 - bsPRev, 86
 - bsPRevValid, 86
 - ccsSupported, 86
 - ccsSupportedValid, 86
 - cdmaSysIdValid, 86
 - isSysPrIMatch, 86
 - isSysPrIMatchValid, 86
 - MCC, 86
 - MNC, 86
 - networkID, 86
 - networkIdValid, 86
 - pRevInUse, 87
 - pRevInUseValid, 87
 - packetZone, 87
 - packetZoneValid, 87
 - sysInfoCDMA, 87
 - systemID, 87
- nas_CDMA SysInfoExt, 87
 - imsi_11_12, 87
 - MCC, 87
- nas_CSGID, 90
 - id, 90
 - mcc, 91
 - mnc, 91
 - mncPcsDigits, 91
 - rat, 91
- nas_CallBarringSysInfo, 79
 - csBarStatus, 80
 - psBarStatus, 80
- nas_CommInfo, 88
 - imsRegState, 90
 - modemMode, 90
 - psState, 90
 - systemMode, 90
 - temperature, 90
- nas_GERANInfo, 95
 - arfcn, 97
 - bsic, 97
 - cellID, 97
 - insNmrCellInfo, 97
 - lac, 97
 - nmrInst, 97
 - plmn, 97
 - rxLev, 97
 - timingAdvance, 97
- nas_GSMRSSIThresh, 99
 - GSMRSSIThreshListLen, 100
 - pGSMRSSIThreshList, 100
- nas_GSMSrvStatusInfo, 100
 - isPrefDataPath, 101
 - srvStatus, 101
 - trueSrvStatus, 101
- nas_GSMSysInfo, 101
 - cellId, 103
 - cellIdValid, 103
 - dtmSupp, 103
 - dtmSuppValid, 103
 - egprsSupp, 103
 - egprsSuppValid, 103
 - lac, 103
 - lacValid, 103
 - MCC, 103
 - MNC, 103
 - networkIdValid, 103
 - regRejectInfoValid, 103
 - rejCause, 103
 - rejectSrvDomain, 104
 - sysInfoGSM, 104
- nas_HDRECIOThresh, 104
 - HDRECIOThreshListLen, 104
 - pHDRECIOThreshList, 104
- nas_HDRIOThresh, 104
 - HDRIOThreshListLen, 105
 - pHDRIOThreshList, 105
- nas_HDRRSSIThresh, 105
 - HDRRSSIThreshListLen, 105
 - pHRRSSIThreshList, 105
- nas_HDRSINRThreshold, 105
 - HDRSINRThresholdListLen, 106
 - pHDRSINRThresholdList, 106
- nas_HDRSysInfo, 106
 - hdrActiveProt, 108
 - hdrActiveProtValid, 108
 - hdrPersonality, 108
 - hdrPersonalityValid, 108
 - is856SysId, 108
 - is856SysIdValid, 108
 - isSysPrIMatch, 108
 - isSysPrIMatchValid, 108
 - sysInfoHDR, 108
- nas_LTEInfo, 110
 - band, 112
 - bandwidth, 112
 - emmConnState, 112
 - emmState, 112
 - emmSubState, 112
 - RXChan, 112
 - TXChan, 112

- nas_LTEInfoInterfreq, 113
 - freqsLen, 113
 - InfoInterfreq, 113
 - ueInIdle, 113
- nas_LTEInfoIntrafreq, 113
 - CellParams, 115
 - cellReselPriority, 115
 - cellsLen, 115
 - earfcn, 115
 - globalCellId, 115
 - plmn, 115
 - sIntraSearch, 115
 - sNonIntraSearch, 115
 - servingCellId, 115
 - tac, 115
 - threshServingLow, 115
 - ueInIdle, 115
- nas_LTEInfoNeighboringGSM, 115
 - freqsLen, 116
 - LteGsmCellInfo, 116
 - ueInIdle, 116
- nas_LTEInfoNeighboringWCDMA, 116
 - freqsLen, 117
 - ueInIdle, 117
- nas_LTERSRPThresh, 117
 - LTERSRPThreshListLen, 117
 - pLTERSRPThreshList, 118
- nas_LTERSRQThresh, 118
 - LTERSRQThreshListLen, 118
 - pLTERSRQThreshList, 118
- nas_LTERSSIThresh, 118
 - LTERSSIThreshListLen, 119
 - pLTERSSIThreshList, 119
- nas_LTESNRThreshold, 120
 - LTESNRThresholdListLen, 120
 - pLTESNRThresholdList, 120
- nas_LTESigRptConfig, 119
 - avgPeriod, 119
 - rptRate, 119
- nas_LTESysInfo, 120
 - cellId, 122
 - cellIdValid, 122
 - lac, 122
 - lacValid, 122
 - MCC, 122
 - MNC, 123
 - networkIdValid, 123
 - regRejectInfoValid, 123
 - rejCause, 123
 - rejectSrvDomain, 123
 - sysInfoLTE, 123
 - tac, 123
 - tacValid, 123
- nas_MNRInfo, 124
 - mcc, 124
 - mnc, 125
 - rat, 125
- nas_PhyCaAggPcellInfo, 127
 - dl_bw_value, 127
 - freq, 127
 - iLTEbandValue, 127
 - pci, 127
 - TlvPresent, 127
- nas_PhyCaAggScellDIBw, 127
 - dl_bw_value, 128
 - TlvPresent, 128
- nas_PhyCaAggScellIndType, 128
 - freq, 129
 - pci, 129
 - scell_state, 129
 - TlvPresent, 129
- nas_PhyCaAggScellIndex, 128
 - scell_idx, 128
 - TlvPresent, 128
- nas_PhyCaAggScellInfo, 129
 - dl_bw_value, 132
 - freq, 132
 - iLTEbandValue, 132
 - pci, 132
 - scell_state, 132
 - TlvPresent, 132
- nas_QmiNas3GppNetworkInfo, 133
 - Description, 133
 - Forbidden, 133
 - InUse, 133
 - MCC, 133
 - MNC, 133
 - Preferred, 133
 - Roaming, 133
- nas_QmiNas3GppNetworkRAT, 133
 - MCC, 134
 - MNC, 134
 - RAT, 134
- nas_QmisNasPcsDigit, 134
 - includes_pcs_digit, 134
 - MCC, 135
 - MNC, 135
- nas_RFInfoTlv, 135
 - activeBandClass, 135
 - activeChannel, 135
 - radioInterface, 136
 - radioInterfaceSize, 136
 - TlvPresent, 136
- nas_RejectReasonTlv, 135
 - rejectCause, 135
 - serviceDomain, 135
 - TlvPresent, 135
- nas_SLQSSignalStrengthsIndReq, 140
 - ecioDelta, 141
 - ecioThresholdList, 141
 - ecioThresholdListLen, 141
 - ioDelta, 141
 - lteRsrpDelta, 141
 - lteSnrDelta, 141
 - rsrqDelta, 141
 - rxSignalStrengthDelta, 141

- sinrDelta, 141
- sinrThresholdList, 141
- sinrThresholdListLen, 141
- nas_SLQSSignalStrengthsInformation, 141
 - ecioInfo, 142
 - errorRateInfo, 142
 - io, 142
 - lteRsrpinfo, 142
 - lteSnrinfo, 142
 - rsrqInfo, 142
 - rxSignalStrengthInfo, 142
 - sinr, 142
- nas_SLQSSignalStrengthsTlv, 142
 - sSLQSSignalStrengthsInfo, 142
 - TlvPresent, 142
- nas_SignalStrengthTlv, 140
 - radiolInterface, 140
 - signalStrength, 140
 - TlvPresent, 140
- nas_SrvStatusInfo, 142
 - isPrefDataPath, 143
 - srvStatus, 143
- nas_TDSCDMAECIOThresh, 146
- nas_TDSCDMARSCPTThresh, 146
- nas_TDSCDMARSSIThresh, 147
- nas_TDSCDMASINRThresh, 147
- nas_UMTSInfo, 149
 - cellID, 151
 - ecio, 151
 - geranInst, 151
 - GeranInstInfo, 151
 - lac, 151
 - plmn, 151
 - psc, 151
 - rscp, 151
 - UMTSInstInfo, 151
 - uarfcn, 151
 - umtsInst, 151
- nas_UMTSinstInfo, 151
 - umtsEcio, 152
 - umtsPsc, 152
 - umtsRscp, 152
 - umtsUarfcn, 152
- nas_UniversalTime, 153
 - day, 154
 - dayOfWeek, 154
 - hour, 154
 - minute, 154
 - month, 154
 - second, 154
 - year, 154
- nas_WCDMAECIOThresh, 155
- nas_WCDMAInfoLTENeighborCell, 156
 - umtsLTENbrCellLen, 156
 - wcdmaRRCTest, 156
- nas_WCDMARSSIThresh, 156
- nas_WCDMASysInfo, 157
 - cellId, 160
 - cellIdValid, 160
 - hsCallStatus, 160
 - hsCallStatusValid, 160
 - hsInd, 160
 - hsIndValid, 160
 - lac, 160
 - lacValid, 160
 - MCC, 160
 - MNC, 160
 - networkIdValid, 160
 - psc, 160
 - pscValid, 160
 - regRejectInfoValid, 160
 - rejCause, 160
 - rejectSrvDomain, 160
 - sysInfoWCDMA, 160
- nas_acqOrderPref, 78
 - acqOrdeLen, 78
 - pAcqOrder, 78
- nas_callBarStatus, 80
 - csBarStatus, 81
 - psBarStatus, 81
- nas_cellParams, 87
 - pci, 88
 - rsrp, 88
 - rsrq, 88
 - rsi, 88
 - srxlev, 88
- nas_currentPLMN, 91
 - MCC, 91
 - MNC, 91
 - netDescr, 91
 - netDescrLength, 92
- nas_dataSrvCapabilities, 92
 - dataCapabilities, 92
 - dataCapabilitiesLen, 92
- nas_detailSvcInfo, 92
 - hdrHybrid, 94
 - hdrSrvStatus, 94
 - isSysForbidden, 94
 - srvCapability, 94
 - srvStatus, 94
- nas_ecioListElement, 94
 - ecio, 94
 - radiolf, 94
- nas_errorRateListElement, 95
 - errorRate, 95
 - radiolf, 95
- nas_geranInstInfo, 97
 - geranArfcn, 98
 - geranBsicBcc, 98
 - geranBsicNcc, 98
 - geranRssi, 98
- nas_gsmCellInfo, 98
 - arfcn, 99
 - band1900, 99
 - bsicId, 99
 - cellIdValid, 99

- rsi, [99](#)
 - srxlev, [99](#)
- nas_infoInterFreq, [108](#)
 - cell_resel_priority, [109](#)
 - cellInterFreqParams, [109](#)
 - cells_len, [109](#)
 - earfcn, [109](#)
 - threshXHigh, [109](#)
 - threshXLow, [109](#)
- nas_lteGsmCellInfo, [109](#)
 - cellReselPriority, [110](#)
 - cells_len, [110](#)
 - GsmCellInfo, [110](#)
 - nccPermitted, [110](#)
 - threshGsmHigh, [110](#)
 - threshGsmLow, [110](#)
- nas_lteRsrpInformation, [117](#)
 - rsrplevel, [117](#)
- nas_lteSnrinformation, [119](#)
 - snrlevel, [120](#)
- nas_lteWcdmaCellInfo, [123](#)
 - cellReselPriority, [124](#)
 - cellsLen, [124](#)
 - threshXhigh, [124](#)
 - threshXlow, [124](#)
 - uarfcn, [124](#)
 - WCDMACellInfo, [124](#)
- nas_major
 - NASLteNasReleaseInfoTlv, [162](#)
- nas_minor
 - NASLteNasReleaseInfoTlv, [162](#)
- nas_netSelectionPref, [125](#)
 - mcc, [125](#)
 - mnc, [125](#)
 - netReg, [125](#)
- nas_nmrCellInfo, [125](#)
 - nmrArfcn, [126](#)
 - nmrBsic, [126](#)
 - nmrCellID, [126](#)
 - nmrLac, [126](#)
 - nmrPlmn, [126](#)
 - nmrRxLev, [126](#)
- nas_qaQmi3Gpp2TimeZone, [132](#)
 - daylightSavings, [132](#)
 - leapSeconds, [132](#)
 - localTimeOffset, [132](#)
- nas_release
 - NASLteNasReleaseInfoTlv, [162](#)
- nas_roamIndList, [136](#)
 - numInstances, [136](#)
 - radiolInterface, [136](#)
 - roamIndicator, [136](#)
- nas_rsrqInformation, [137](#)
 - radiolf, [137](#)
 - rsrq, [137](#)
- nas_rxSignalStrengthListElement, [137](#)
 - radiolf, [138](#)
 - rxSignalStrength, [138](#)
- nas_servSystem, [138](#)
 - csAttachState, [139](#)
 - numRadiolInterfaces, [139](#)
 - psAttachState, [139](#)
 - radiolInterface, [139](#)
 - regState, [139](#)
 - selNetwork, [139](#)
- nas_sysInfoCommon, [143](#)
 - isSysForbidden, [145](#)
 - isSysForbiddenValid, [145](#)
 - roamStatus, [145](#)
 - roamStatusValid, [145](#)
 - srvCapability, [145](#)
 - srvCapabilityValid, [145](#)
 - srvDomain, [145](#)
 - srvDomainValid, [145](#)
- nas_timeInfo, [148](#)
 - day, [149](#)
 - dayLtSavingAdj, [149](#)
 - dayOfWeek, [149](#)
 - hour, [149](#)
 - minute, [149](#)
 - month, [149](#)
 - radiolInterface, [149](#)
 - second, [149](#)
 - timeZone, [149](#)
 - TlvPresent, [149](#)
 - year, [149](#)
- nas_umtsLTENbrCell, [152](#)
 - cellsTDD, [153](#)
 - earfcn, [153](#)
 - pci, [153](#)
 - rsrp, [153](#)
 - rsrq, [153](#)
 - srxlev, [153](#)
- nas_wcdmaCellInfo, [154](#)
 - cpich_ecno, [155](#)
 - cpich_rscp, [155](#)
 - psc, [155](#)
 - srxlev, [155](#)
- NasGetLTECphyCaInfo, [161](#)
 - PhyCaAggPcellInfo, [161](#)
 - PhyCaAggScellIDBw, [161](#)
 - PhyCaAggScellIndType, [161](#)
 - PhyCaAggScellIndex, [161](#)
 - PhyCaAggScellInfo, [161](#)
- nasRelInfoTlv
 - NASQmiCbkNasSwiOTAMessageInd, [167](#)
- nccPermitted
 - nas_lteGsmCellInfo, [110](#)
- netDescr
 - nas_currentPLMN, [91](#)
- netDescrLength
 - nas_currentPLMN, [92](#)
- netInfoLen
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- netReg

- nas_netSelectionPref, [125](#)
- NetSelPref
 - NASNetSelPreferenceTlv, [163](#)
- NetworkID
 - unpack_nas_SLQSGetServingSystem_t, [311](#)
- networkID
 - nas_CDMA SysInfo, [86](#)
- networkIdValid
 - nas_CDMA SysInfo, [86](#)
 - nas_GSM SysInfo, [103](#)
 - nas_LTE SysInfo, [123](#)
 - nas_WCDMA SysInfo, [160](#)
- networkInfoLen
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [377](#)
- NetworkType
 - currNetworkInfo, [28](#)
 - wds_currNetworkInfo, [389](#)
- NewMMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [353](#)
- newMTMessageTlv, [170](#)
 - MTMessageInfo, [170](#)
 - TlvPresent, [171](#)
- newPINLen
 - uim_unblockUIMPIN, [257](#)
- newPINVal
 - uim_unblockUIMPIN, [257](#)
- nextHeader
 - LibPackTFTIDParams, [64](#)
- nid
 - nas_CDMA Info, [82](#)
 - unpack_nas_GetHomeNetwork_t, [300](#)
- nmrArfcn
 - nas_nmrCellInfo, [126](#)
- nmrBsic
 - nas_nmrCellInfo, [126](#)
- nmrCellID
 - nas_nmrCellInfo, [126](#)
- nmrInst
 - nas_GERAN Info, [97](#)
- nmrLac
 - nas_nmrCellInfo, [126](#)
- nmrPlmn
 - nas_nmrCellInfo, [126](#)
- nmrRxLev
 - nas_nmrCellInfo, [126](#)
- notification
 - unpack_omaDmNotificationsTlv_t, [331](#)
- notificationType
 - sMSEtwsMessage, [238](#)
- notused
 - unpack_dms_SetCrashAction_t, [269](#)
- num_instances
 - DMScustSettingList, [31](#)
- numApp
 - slotInf, [236](#)
 - uim_slotInfo, [255](#)
- numEntries
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [278](#)
- NumFlows
 - unpack_qos_SLQSSetQosEventCallback_ind_t, [340](#)
- numInstances
 - nas_roamIndList, [136](#)
 - wds_DomainNameList, [390](#)
 - wds_PCSCFFQDNAddressList, [393](#)
 - wds_PCSCFIPv4ServerAddressList, [393](#)
- numOpt
 - wdsDhcpv4OptionList, [398](#)
- numQosFlow
 - unpack_qos_SLQSQosSwiReadDataStats_t, [340](#)
- numRadioInterfaces
 - nas_servSystem, [139](#)
- NumRxFilters
 - unpack_qos_QosFlowInfo_t, [336](#)
- numSlot
 - uim_cardStatus, [247](#)
- NumSupUSBComps
 - unpack_dms_GetUSBComp_t, [268](#)
- NumTxFilters
 - unpack_qos_QosFlowInfo_t, [336](#)
- NxtHdrProto
 - unpack_qos_swiQosFilter_t, [346](#)
- OMADMEnabled
 - unpack_swioma_SLQSOMADMGetSettings_t, [362](#)
- OfflineReason
 - unpack_dms_GetPower_t, [267](#)
- offset
 - uim_readTransparentInfo, [251](#)
- oldPINLen
 - uim_changeUIMPIN, [248](#)
- oldPINVal
 - uim_changeUIMPIN, [248](#)
- operatingMode
 - dms_OperatingModeTlv, [29](#)
- OperatingModeTlv
 - unpack_dms_SetEventReport_ind_t, [270](#)
- OperationMode
 - unpack_dms_GetPower_t, [267](#)
- optCode
 - wdsDhcpv4Option, [398](#)
- optVal
 - wdsDhcpv4Option, [398](#)
- optValLen
 - wdsDhcpv4Option, [398](#)
- otaMsgTlv
 - NASQmiCbkNasSwiOTAMessageInd, [167](#)
- p3GPP2TimeInfo
 - unpack_nas_SLQSGetNetworkTime_t, [308](#)
- p3GPPTIMEInfo
 - unpack_nas_SLQSGetNetworkTime_t, [308](#)
- p3GppNetworkInfoInstances
 - unpack_nas_PerformNetworkScan_t, [304](#)
- p3GppNetworkInstanceSize
 - unpack_nas_PerformNetworkScan_t, [304](#)

- pAAASPI
 - pack_wds_SetMobileIPProfile_t, 221
- PACK_WDS_IPV4
 - wds.h, 567
- PACK_WDS_IPV6
 - wds.h, 567
- pAPNClass
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 54
- pAPNClass3GPP2
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAPNDisabledFlag
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 54
- pAPNEnabled3GPP2
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAPNName
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 54
- pAPNnameSize
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 54
- pAcqOrder
 - nas_acqOrderPref, 78
- pAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, 199
- pAddCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 327
- pAddGSMSSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 327
- pAddHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 327
- pAddLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 327
- pAddWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 327
- pAddrAllocPref
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 54
- pAddress
 - pack_wds_SetMobileIPProfile_t, 221
- pAllowLinger
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAltitudeAssumed
 - unpack_loc_PositionRpt_Ind_t, 294
- pAltitudeWrtEllipsoid
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 294
- pAltitudeWrtMeanSeaLevel
 - unpack_loc_BestAvailPos_Ind_t, 287
- unpack_loc_PositionRpt_Ind_t, 294
- pApnString
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pApnStringSize
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pApnname
 - pack_wds_SetDefaultProfile_t, 220
- pAppName
 - loc_LocApplicationInfo, 73
- pAppPriority
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAppProvider
 - loc_LocApplicationInfo, 73
- pAppType
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAppVersion
 - loc_LocApplicationInfo, 73
- pApplicationInfo
 - pack_loc_Start_t, 183
- pAuth
 - pack_wds_SLQSStartDataSession_t, 229
- pAuthPassword
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAuthPassword_tSize
 - LibPackprofile_3GPP2, 60
- pAuthPasswordSize
 - LibpackProfile3GPP2, 49
- pAuthProtocol
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAuthRetryCount
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAuthTimeout
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pAuthenticationPref
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 55
- pAutosdm
 - pack_swima_SLQSOMADMSSetSettings_t, 210
- pBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 199
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pBdsSVInfo
 - pack_loc_Delete_Assist_Data_t, 177
- pCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 189
- pCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pCDMAECIOThreshList
 - nas_CDMAECIOThresh, 81
- pCDMAInfo

- unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pCDMARSSIThreshList
 - nas_CDMARSSIThresh, 83
- pCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 323
- pCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- PCSCFAddrPCO
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- PCSCFFQDNAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- pCSGID
 - pack_nas_SLQSSetSysSelectionPref_t, 199
- pCardResult
 - unpack_uim_ReadTransparent_t, 365
- pCardStatus
 - unpack_uim_GetCardStatus_t, 364
 - unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 367
- pCellDb
 - pack_loc_Delete_Assist_Data_t, 177
- pChangeDuration
 - pack_nas_SLQSInitiateNetworkRegistration_t, 186
- pChgDuration
 - pack_nas_SLQSSetSysSelectionPref_t, 199
- pClkInfo
 - pack_loc_Delete_Assist_Data_t, 177
- pConfigAltitudeAssumed
 - pack_loc_Start_t, 183
- pCreateProfileOut
 - unpack_wds_SLQSCreateProfile_t, 375
- pCurProfile
 - pack_wds_SLQSCreateProfile_t, 222
- pCurrImgInfo
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 278
- pCustSettingInfo
 - DMSgetCustomFeatureV2, 32
- pCustSettingList
 - DMSgetCustomFeatureV2, 32
- pDDTMInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- PDOP
 - loc_precisionDilution, 74
- PDPTYPE
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- pDataMode
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pDataRate
 - LibpackProfile3GPP2, 49
- LibPackprofile_3GPP2, 60
- pDayItSavAdj
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 322
- pDeferTime
 - pack_swima_SLQSOMADMSendSelection_t, 209
- pDestSMSContent
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, 175
 - packgetDyingGaspCfg, 231
- pDestSMSNum
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, 175
 - packgetDyingGaspCfg, 231
- pDualStandByPrefInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pEmerMode
 - pack_nas_SLQSSetSysSelectionPref_t, 199
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pEnabled
 - pack_wds_SetMobileIPProfile_t, 221
- pEncryptData
 - pack_uim_ReadTransparent_t, 212
- pEncryptedData
 - unpack_uim_ReadTransparent_t, 365
- pEncryptedPIN1
 - pack_uim_VerifyPin_t, 217
 - unpack_uim_ChangePin_t, 363
 - unpack_uim_SetPinProtection_t, 365
 - unpack_uim_UnblockPin_t, 368
 - unpack_uim_VerifyPin_t, 369
- pErrorRateInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pExtErrCode
 - UnPackGetProfileSettingOut, 387
- pExtErrorCode
 - unpack_wds_SLQSMModifyProfile_t, 381
- pFailureReason
 - unpack_wds_SLQSSStartDataSession_t, 385
- pFixId
 - unpack_loc_PositionRpt_Ind_t, 294
- pFwAutoCheck
 - pack_swima_SLQSOMADMSSetSettings_t, 210
- pGERANInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pGPRSMInimumQoS
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 55
- pGPRSRequestedQoS
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 55
- pGSMCallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pGSMCipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pGSMRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190

- pGSMRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pGSMRSSIThreshList
 - nas_GSMRSSIThresh, 100
- pGMSSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 323
- pGMSMrvStatusInfo
 - unpack_nas_SLQSSetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pGMSMSysInfo
 - unpack_nas_SLQSSetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pGWAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, 199
 - unpack_nas_SLQSSetSysSelectionPref_t, 319
- pGetCustomInput
 - DMSgetCustomFeatureV2, 32
- pGetDyingGaspCfg
 - unpack_dms_SLQSSwiGetDyingGaspCfg_t, 277
- pGetDyingGaspStatistics
 - unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 277
- pGnssData
 - pack_loc_Delete_Assist_Data_t, 178
- pGpsTime
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 294
- pHASPI
 - pack_wds_SetMobileIPProfile_t, 221
- pHDRECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRECIOThreshList
 - nas_HDRECIOThresh, 104
- pHDRIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRIOThreshList
 - nas_HDRIOThresh, 105
- pHDRNewUATIAssInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pHDRRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRRSSIThreshList
 - nas_HDRRSSIThresh, 105
- pHDRSINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRSINRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pHDRSINRThreshList
 - nas_HDRSINRThreshold, 106
- pHDRSessionCloseInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pHDRSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 323
- pHDRSrvStatusInfo
 - unpack_nas_SLQSSetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pHDRSysInfo
 - unpack_nas_SLQSSetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pHeading
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 294
- pHeadingUnc
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorCirConf
 - unpack_loc_BestAvailPos_Ind_t, 287
- pHorConfidence
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorEllpConf
 - unpack_loc_BestAvailPos_Ind_t, 287
- pHorReliability
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorUncCircular
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorUncEllipseOrientAzimuth
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorUncEllipseSemiMajor
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorUncEllipseSemiMinor
 - unpack_loc_BestAvailPos_Ind_t, 287
 - unpack_loc_PositionRpt_Ind_t, 295
- pHorizontalAccuracyLvl
 - pack_loc_Start_t, 183
- pHotSwapStatus
 - unpack_uim_GetCardStatus_t, 364
- pHwConfig
 - unpack_wds_SLQSSetDHCPv4ClientConfig_t, 384
- pIPv4AddrPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pIPv6AddPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- plmCnFlag
 - LibpackProfile3GPP, 43
 - LibPackprofile_3GPP, 55
- plmImageList
 - pack_fms_SetImagesPreference_t, 177
 - unpack_fms_GetImagesPreference_t, 281
- plndicationToken
 - pack_uim_ChangePin_t, 211
 - pack_uim_ReadTransparent_t, 212
 - pack_uim_SetPinProtection_t, 213
 - pack_uim_UnblockPin_t, 216

- pack_uim_VerifyPin_t, 217
- unpack_uim_ChangePin_t, 363
- unpack_uim_ReadTransparent_t, 365
- unpack_uim_SetPinProtection_t, 365
- unpack_uim_UnblockPin_t, 368
- unpack_uim_VerifyPin_t, 369
- pIntermediateReportState
 - pack_loc_Start_t, 183
- plpcpAckTimeout
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- plpcpCreqRetryCount
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- plsPcscfAddressNedded
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pKeyReferenceID
 - pack_uim_ChangePin_t, 211
 - pack_uim_SetPinProtection_t, 214
 - pack_uim_UnblockPin_t, 216
 - pack_uim_VerifyPin_t, 217
- pLTEBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 199
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pLTECphyCa
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pLTEInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, 323
- pLTEInfoInterfreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pLTEInfoIntrafreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pLTEInfoNeighboringGSM
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pLTEInfoNeighboringWCDMA
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pLTERSRPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTERSRPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTERSRPThreshList
 - nas_LTERSRPThresh, 118
- pLTERSRQDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTERSRQThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTERSRQThreshList
 - nas_LTERSRQThresh, 118
- pLTERSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTERSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTERSSIThreshList
 - nas_LTERSSIThresh, 119
- pLTESNRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTESNRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTESNRThreshList
 - nas_LTESNRThresh, 120
- pLTESigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 323
- pLTESigRptConfig
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pLTESrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pLTEVoiceSupportSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pLatitude
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pLcpAckTimeout
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pLcpCreqRetryCount
 - LibpackProfile3GPP2, 49
 - LibPackprofile_3GPP2, 60
- pLeapSeconds
 - unpack_loc_PositionRpt_Ind_t, 295
- pLinktimer
 - pack_sms_SendSMS_t, 202
- pLongitude
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pMNAAB
 - pack_wds_SetMobileIPPProfile_t, 221
- pMNCIncPCSDigStat
 - pack_nas_SLQSSetSysSelectionPref_t, 200
- pMNHA
 - pack_wds_SetMobileIPPProfile_t, 221
- pMNRInfo
 - pack_nas_SLQSInitiateNetworkRegistration_t, 186
- pMagneticDeviation
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pManagedRoamingInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pMessage
 - pack_sms_SendSMS_t, 202
- pMessageIndex
 - pack_sms_SLQSDeleteSMS_t, 203
- pMessageMode
 - pack_sms_SLQSDeleteSMS_t, 204
 - pack_sms_SLQSGetSMS_t, 204
 - pack_sms_SLQSGetSMSList_t, 205
 - pack_sms_SLQSModifySMSStatus_t, 206
- pMessageTag
 - pack_sms_SLQSDeleteSMS_t, 204
- pMinIntervalTime
 - pack_loc_Start_t, 183

- pMncPcsDigitStatus
 - pack_nas_SLQSIInitiateNetworkRegistration_t, 186
- pMncPcsStatus
 - pack_nas_SLQSGetPLMNName_t, 185
- pModePref
 - pack_nas_SLQSSetSysSelectionPref_t, 200
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pNAI
 - pack_wds_SetMobileIPProfile_t, 221
- pName
 - pack_wds_SetDefaultProfile_t, 220
- pNegoDnsSrvrPref
 - LibpackProfile3GPP, 50
 - LibPackprofile_3GPP, 61
- pNetSelPref
 - pack_nas_SLQSSetSysSelectionPref_t, 200
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pNetworkTimeInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pNumberOfPhySlot
 - unpack_uim_SLQSUIGetSlotsStatus_t, 367
- pOptList
 - wdsDhcpv4OptionList, 398
- pPCSInstance
 - unpack_nas_PerformNetworkScan_t, 304
- pPCSInstanceSize
 - unpack_nas_PerformNetworkScan_t, 304
- pPDNInactivTimeout
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPDNInactivTimeout3GPP2
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pPDPTYPE
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPRLPref
 - pack_nas_SLQSSetSysSelectionPref_t, 200
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pPass
 - pack_wds_SLQSStartDataSession_t, 229
- pPassword
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
 - pack_wds_SetDefaultProfile_t, 220
- pPasswordSize
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPcscfAddrUsingDhcp
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPcscfAddrUsingPCO
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPdnType
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pPdpAccessConFlag
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPdpContext
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPdpDataCompType
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPdpHdrCompType
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPppSessCloseTimer1x
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pPppSessCloseTimerDO
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pPrecisionDilution
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pPriDNSIPv4AddPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPriDNSIPv6addpref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPriV6DnsAddress
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pPrimaryHA
 - pack_wds_SetMobileIPProfile_t, 221
- pPrimaryID
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pPrimaryV4DnsAddress
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pProfileID
 - unpack_wds_SLQSCreateProfile_t, 375
- pProfileId
 - pack_wds_SLQSCreateProfile_t, 222
 - pack_wds_SLQSModifyProfile_t, 226
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, 228
- pProfileSettings
 - unpack_wds_SLQSGetProfileSettings_t, 379
- pProfileType
 - pack_wds_SLQSCreateProfile_t, 222
 - pack_wds_SLQSModifyProfile_t, 226
- pProfileName
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pProfileNameSize
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pQosClassID
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pRAT

- pack_nas_SLQSSetSysSelectionPref_t, 200
- pRATInstance
 - unpack_nas_PerformNetworkScan_t, 304
- pRATInstanceSize
 - unpack_nas_PerformNetworkScan_t, 304
- pRATType
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- PRIStrng
 - unpack_dms_GetFirmwareRevision_t, 263
 - unpack_dms_GetFirmwareRevisions_t, 264
- PRLInd
 - unpack_nas_SLQSGetServingSystem_t, 311
- PRLPTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- PRLPref
 - NASPRLPreferenceTlv, 167
- pRadioInterface
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 322
- pRankIndicatorInd
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 195
- pReadResult
 - unpack_uim_ReadTransparent_t, 365
- pRecurrenceType
 - pack_loc_Start_t, 183
- pRejectReason
 - pack_swioama_SLQSOMADMSendSelection_t, 209
- pRemainingRetries
 - unpack_uim_ChangePin_t, 363
 - unpack_uim_SetPinProtection_t, 365
 - unpack_uim_UnblockPin_t, 368
 - unpack_uim_VerifyPin_t, 369
- pReportChannelRate
 - pack_wds_SLQSGetDUNCallInfo_t, 223
- pReportConnStatus
 - pack_wds_SLQSGetDUNCallInfo_t, 223
- pReportDataBearerTech
 - pack_wds_SLQSGetDUNCallInfo_t, 223
- pReportDormStatus
 - pack_wds_SLQSGetDUNCallInfo_t, 223
- pReqSettings
 - pack_wds_SLQSGetRuntimeSettings_t, 225
- pRequestOptionList
 - unpack_wds_SLQSSetDHCPv4ClientConfig_t, 385
- pRequestedTag
 - pack_sms_SLQSGetSMSList_t, 205
- pRevInUse
 - nas_CDMASysInfo, 87
- pRevInUseValid
 - nas_CDMASysInfo, 87
- pRevTunneling
 - pack_wds_SetMobileIPProfile_t, 221
- pRoamPref
 - pack_nas_SLQSSetSysSelectionPref_t, 200
- unpack_nas_SLQSGetSysSelectionPref_t, 319
- pRscp
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 323
- PSDomain
 - unpack_nas_GetServingNetwork_t, 302
- pSMSAttemptedFlag
 - packgetDyingGaspStatistics, 231
- pSV
 - loc_BdsSVInfo, 68
 - loc_SVInfo, 76
- pSVInfo
 - pack_loc_Delete_Assist_Data_t, 178
- pScanResult
 - unpack_nas_PerformNetworkScan_t, 304
- pSecDNSIPv4AddPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pSecDNSIPv6addpref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pSecV6DnsAddress
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pSecondaryFlag
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pSecondaryHA
 - pack_wds_SetMobileIPProfile_t, 221
- pSecondaryV4DnsAddress
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pSensorDataUsage
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pServingSystemInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pSigIndReq
 - pack_nas_SLQSSetSignalStrengthsCallback_t, 195
- pSignalStrengthInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pSourceIP
 - LibPackTFTIDParams, 64
- pSpeedHorizontal
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pSpeedUnc
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pSpeedVertical
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pSpeedVerticalUnc
 - unpack_loc_BestAvailPos_Ind_t, 288
- pSrvDomainPref
 - pack_nas_SLQSSetSysSelectionPref_t, 200
 - unpack_nas_SLQSGetSysSelectionPref_t, 319
- pSrvRegRestriction

- pack_nas_SLQSSetSysSelectionPref_t, 200
- pSubscriptionInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pSvUsedForFix
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pSysInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 193
- pSysInfoNoChange
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328
- pSystemSelectionInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 194
- pTDSCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pTDSCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pTDSCDMAECIOThreshList
 - nas_TDSCDMAECIOThresh, 146
- pTDSCDMARSCPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pTDSCDMARSCPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pTDSCDMARSCPThreshList
 - nas_TDSCDMARSCPThresh, 146
- pTDSCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pTDSCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 190
- pTDSCDMARSSIThreshList
 - nas_TDSCDMARSSIThresh, 147
- pTDSCDMASINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 191
- pTDSCDMASINRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 191
- pTDSCDMASINRThreshList
 - nas_TDSCDMASINRThresh, 147
- pTDSCDMASigInfoExt
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 323
- pTFTID1Params
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pTFTID2Params
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pTdsdmaBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 200
- pTech
 - pack_wds_SLQSSStartDataSession_t, 229
- pTechnologyMask
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pTimeSrc
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pTimeStamp
 - packgetDyingGaspStatistics, 231
- pTimeUnc
 - unpack_loc_BestAvailPos_Ind_t, 288
- unpack_loc_PositionRpt_Ind_t, 295
- pTimeZone
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 322
- pTimestampUtc
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pTransferStatInd
 - pack_wds_SLQSGetDUNCallInfo_t, 223
- pUMTSCellID
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 320
- pUMTSInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 321
- pUMTSMInQoS
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 55
- pUMTSMInQoSList
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pUMTSReqQoS
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pUMTSReqQoSSigInd
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pUimSlotsStatus
 - unpack_uim_SLQSUIMGetSlotsStatus_t, 367
- pUser
 - pack_wds_SLQSSStartDataSession_t, 230
- pUserId
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pUserIdSize
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pUsername
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
 - pack_wds_SetDefaultProfile_t, 220
- pUsernameSize
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pVerboseFailReasonType
 - unpack_wds_SLQSSStartDataSession_t, 385
- pVerboseFailureReason
 - unpack_wds_SLQSSStartDataSession_t, 385
- pVertConfidence
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pVertReliability
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pVertUnc
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_PositionRpt_Ind_t, 295
- pWCDMACallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 315
 - unpack_nas_SLQSSysInfoCallback_ind_t, 328

- pWCDMACipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [315](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [328](#)
- pWCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [191](#)
- pWCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [191](#)
- pWCDMAECIOThreshList
 - nas_WCDMAECIOThresh, [155](#)
- pWCDMAInfoLTENeighborCell
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [321](#)
- pWCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [191](#)
- pWCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [191](#)
- pWCDMARSSIThreshList
 - nas_WCDMARSSIThresh, [157](#)
- pWCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, [323](#)
- pWCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, [315](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [328](#)
- pWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [315](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [328](#)
- pXid
 - unpack_loc_BestAvailPos_Ind_t, [288](#)
- pack_dms_GetActivationState
 - dms.h, [411](#)
- pack_dms_GetBandCapability
 - dms.h, [411](#)
- pack_dms_GetCrashAction
 - dms.h, [411](#)
- pack_dms_GetCustFeature
 - dms.h, [412](#)
- pack_dms_GetCustFeaturesV2
 - dms.h, [412](#)
- pack_dms_GetCustFeaturesV2_t, [171](#)
 - cust_id, [171](#)
 - list_type, [171](#)
 - Tlvresult, [171](#)
- pack_dms_GetDeviceCap
 - dms.h, [412](#)
- pack_dms_GetDeviceCapabilities
 - dms.h, [412](#)
- pack_dms_GetDeviceHardwareRev
 - dms.h, [413](#)
- pack_dms_GetDeviceMfr
 - dms.h, [413](#)
- pack_dms_GetDeviceSerialNumbers
 - dms.h, [413](#)
- pack_dms_GetFSN
 - dms.h, [415](#)
- pack_dms_GetFirmwareInfo
 - dms.h, [414](#)
- pack_dms_GetFirmwareRevision
 - dms.h, [414](#)
- pack_dms_GetFirmwareRevisions
 - dms.h, [414](#)
- pack_dms_GetHardwareRevision
 - dms.h, [415](#)
- pack_dms_GetIMSI
 - dms.h, [415](#)
- pack_dms_GetModelID
 - dms.h, [416](#)
- pack_dms_GetNetworkTime
 - dms.h, [416](#)
- pack_dms_GetPRLVersion
 - dms.h, [417](#)
- pack_dms_GetPower
 - dms.h, [416](#)
- pack_dms_GetSerialNumbers
 - dms.h, [417](#)
- pack_dms_GetUSBComp
 - dms.h, [418](#)
- pack_dms_GetVoiceNumber
 - dms.h, [418](#)
- pack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [421](#)
- pack_dms_SLQSDmsSwiIndicationRegister
 - dms.h, [421](#)
- pack_dms_SLQSDmsSwiIndicationRegister_t, [174](#)
 - resetInfoInd, [174](#)
- pack_dms_SLQSGetBandCapability
 - dms.h, [422](#)
- pack_dms_SLQSSwiClearDyingGaspStatistics
 - dms.h, [422](#)
- pack_dms_SLQSSwiGetDyingGaspCfg
 - dms.h, [423](#)
- pack_dms_SLQSSwiGetDyingGaspStatistics
 - dms.h, [423](#)
- pack_dms_SLQSSwiGetFirmwareCurr
 - dms.h, [423](#)
- pack_dms_SLQSSwiGetFwUpdateStatus
 - dms.h, [424](#)
- pack_dms_SLQSSwiSetDyingGaspCfg
 - dms.h, [424](#)
- pack_dms_SLQSSwiSetDyingGaspCfg_t, [174](#)
 - pDestSMSContent, [175](#)
 - pDestSMSNum, [175](#)
- pack_dms_SetCrashAction
 - dms.h, [418](#)
- pack_dms_SetCrashAction_t, [171](#)
 - crashAction, [172](#)
- pack_dms_SetCustFeature
 - dms.h, [419](#)
- pack_dms_SetCustFeature_t, [172](#)
 - DHCPRelayEnabled, [172](#)
 - DisableIMSI, [172](#)
 - GPSPMP, [172](#)
 - GPSSel, [172](#)
 - GpsEnable, [172](#)
 - IPFamSupport, [172](#)
 - IsVoiceEnabled, [172](#)
 - RMAutoConnect, [172](#)
 - SMSSupport, [172](#)

- pack_dms_SetCustFeaturesV2
 - dms.h, [419](#)
- pack_dms_SetCustFeaturesV2_t, [172](#)
 - cust_id, [173](#)
 - cust_value, [173](#)
 - Tlvresult, [173](#)
 - value_length, [173](#)
- pack_dms_SetEventReport
 - dms.h, [420](#)
- pack_dms_SetEventReport_t, [173](#)
 - mode, [173](#)
- pack_dms_SetFirmwarePreference
 - dms.h, [420](#)
- pack_dms_SetPower
 - dms.h, [420](#)
- pack_dms_SetPower_t, [173](#)
 - mode, [173](#)
 - Tlvresult, [174](#)
- pack_dms_SetUSBComp
 - dms.h, [421](#)
- pack_dms_SetUSBComp_t, [174](#)
 - Tlvresult, [174](#)
 - USBComp, [174](#)
- pack_dms_UIMGetICCID
 - dms.h, [424](#)
- pack_dms_UIMGetICCID_t, [175](#)
 - Tlvresult, [175](#)
- pack_fms_GetImagesPreference
 - fms.h, [441](#)
- pack_fms_GetImagesPreference_t, [175](#)
 - Tlvresult, [175](#)
- pack_fms_GetStoredImages
 - fms.h, [441](#)
- pack_fms_GetStoredImages_t, [176](#)
 - Tlvresult, [176](#)
- pack_fms_SetImagesPreference
 - fms.h, [441](#)
- pack_fms_SetImagesPreference_t, [176](#)
 - bForceDownload, [177](#)
 - imageListSize, [177](#)
 - modemindex, [177](#)
 - pImageList, [177](#)
 - Tlvresult, [177](#)
- pack_loc_Delete_Assist_Data_t, [177](#)
 - pBdsSVInfo, [177](#)
 - pCellDb, [177](#)
 - pClkInfo, [177](#)
 - pGnssData, [178](#)
 - pSVInfo, [178](#)
 - Tlvresult, [178](#)
- pack_loc_DeleteAssistData
 - loc.h, [447](#)
- pack_loc_EventRegister
 - loc.h, [447](#)
- pack_loc_EventRegister_t, [178](#)
 - eventRegister, [180](#)
 - Tlvresult, [180](#)
- pack_loc_SLQSLOCGetBestAvailPos
 - loc.h, [448](#)
- pack_loc_SLQSLOCGetBestAvailPos_t, [181](#)
 - Tlvresult, [181](#)
 - xid, [181](#)
- pack_loc_SetExtPowerState
 - loc.h, [447](#)
- pack_loc_SetExtPowerState_t, [180](#)
 - extPowerState, [180](#)
 - Tlvresult, [180](#)
- pack_loc_SetOperationMode
 - loc.h, [448](#)
- pack_loc_SetOperationMode_t, [180](#)
 - mode, [181](#)
 - Tlvresult, [181](#)
- pack_loc_Start
 - loc.h, [448](#)
- pack_loc_Start_t, [181](#)
 - pApplicationInfo, [183](#)
 - pConfigAltitudeAssumed, [183](#)
 - pHorizontalAccuracyLvl, [183](#)
 - pIntermediateReportState, [183](#)
 - pMinIntervalTime, [183](#)
 - pRecurrenceType, [183](#)
 - SessionId, [183](#)
 - Tlvresult, [183](#)
- pack_loc_Stop
 - loc.h, [449](#)
- pack_loc_Stop_t, [183](#)
 - SessionId, [183](#)
 - Tlvresult, [183](#)
- pack_nas_GetACCOLC
 - nas.h, [459](#)
- pack_nas_GetANAAAuthenticationStatus
 - nas.h, [460](#)
- pack_nas_GetCDMANetworkParameters
 - nas.h, [460](#)
- pack_nas_GetHomeNetwork
 - nas.h, [460](#)
- pack_nas_GetNetworkPreference
 - nas.h, [461](#)
- pack_nas_GetRFInfo
 - nas.h, [461](#)
- pack_nas_GetServingNetwork
 - nas.h, [461](#)
- pack_nas_GetServingNetworkCapabilities
 - nas.h, [461](#)
- pack_nas_GetSignalStrengths
 - nas.h, [462](#)
- pack_nas_PerformNetworkScan
 - nas.h, [462](#)
- pack_nas_SLQSGetNetworkTime
 - nas.h, [464](#)
- pack_nas_SLQSGetPLMNName
 - nas.h, [464](#)
- pack_nas_SLQSGetPLMNName_t, [185](#)
 - mcc, [185](#)
 - mnc, [185](#)
 - pMncPcsStatus, [185](#)

- pack_nas_SLQSGetservingSystem
 - nas.h, [464](#)
- pack_nas_SLQSGetSignalStrength
 - nas.h, [464](#)
- pack_nas_SLQSGetSysInfo
 - nas.h, [465](#)
- pack_nas_SLQSGetSysSelectionPref
 - nas.h, [465](#)
- pack_nas_SLQSInitiateNetworkRegistration
 - nas.h, [465](#)
- pack_nas_SLQSInitiateNetworkRegistration_t, [185](#)
 - pChangeDuration, [186](#)
 - pMNRInfo, [186](#)
 - pMncPcsDigitStatus, [186](#)
 - regAction, [186](#)
- pack_nas_SLQSNasConfigSigInfo2
 - nas.h, [466](#)
- pack_nas_SLQSNasConfigSigInfo2_t, [186](#)
 - pHDRIODelta, [190](#)
 - pHDRIOTresh, [190](#)
 - pLTESigRptConfig, [190](#)
- pack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [466](#)
- pack_nas_SLQSNasGetSigInfo
 - nas.h, [466](#)
- pack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [467](#)
- pack_nas_SLQSNasIndicationRegisterExt_t, [191](#)
 - pDDTMInd, [193](#)
 - pDualStandByPrefInd, [193](#)
 - pErrorRateInd, [193](#)
 - pHDRSessionCloseInd, [193](#)
 - pLTECphyCa, [193](#)
 - pManagedRoamingInd, [193](#)
 - pNetworkTimeInd, [193](#)
 - pServingSystemInd, [193](#)
 - pSignalStrengthInd, [193](#)
 - pSubscriptionInfoInd, [193](#)
 - pSysInfoInd, [193](#)
 - pSystemSelectionInd, [194](#)
- pack_nas_SLQSNasSmiModemStatus
 - nas.h, [467](#)
- pack_nas_SLQSNasSmiOTAMessageCallback
 - nas.h, [467](#)
- pack_nas_SLQSNasSmiOTAMessageCallback_t, [194](#)
 - gsmUmtsDI, [194](#)
 - gsmUmtsUI, [195](#)
 - lteEmmDI, [195](#)
 - lteEmmUI, [195](#)
 - lteEsmDI, [195](#)
 - lteEsmUI, [195](#)
 - pRankIndicatorInd, [195](#)
- pack_nas_SLQSSetBandPreference
 - nas.h, [468](#)
- pack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [468](#)
- pack_nas_SLQSSetSignalStrengthsCallback_t, [195](#)
 - bEnable, [195](#)
- pSigIndReq, [195](#)
- pack_nas_SLQSSetSysSelectionPref
 - nas.h, [468](#)
- pack_nas_SLQSSetSysSelectionPref_t, [195](#)
 - pAcqOrderPref, [199](#)
 - pBandPref, [199](#)
 - pCSGID, [199](#)
 - pChgDuration, [199](#)
 - pEmerMode, [199](#)
 - pGWAqOrderPref, [199](#)
 - pLTEBandPref, [199](#)
 - pModePref, [200](#)
 - pNetSelPref, [200](#)
 - pPRLPref, [200](#)
 - pRAT, [200](#)
 - pRoamPref, [200](#)
 - pSrvDomainPref, [200](#)
 - pSrvRegRestriction, [200](#)
 - pTdsdmaBandPref, [200](#)
- pack_nas_SLQSSwiGetLteCQI
 - nas.h, [469](#)
- pack_nas_SetACCOLC
 - nas.h, [462](#)
- pack_nas_SetACCOLC_t, [183](#)
 - accolc, [184](#)
 - spc, [184](#)
- pack_nas_SetLURejectCallback
 - nas.h, [463](#)
- pack_nas_SetNetworkPreference
 - nas.h, [463](#)
- pack_nas_SetNetworkPreference_t, [184](#)
 - Duration, [184](#)
 - TechnologyPref, [185](#)
 - Tlvresult, [185](#)
- pack_nas_SetRFInfoCallback
 - nas.h, [463](#)
- pack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [463](#)
- pack_qmi_t, [200](#)
 - msgid, [200](#)
 - svc, [200](#)
 - timeout, [200](#)
 - xid, [200](#)
- pack_qos_SLQSQosGetNetworkStatus
 - qos.h, [527](#)
- pack_qos_SLQSQosSmiReadApnExtraParams
 - qos.h, [527](#)
- pack_qos_SLQSQosSmiReadApnExtraParams_t, [200](#)
 - apnId, [201](#)
- pack_qos_SLQSQosSmiReadDataStats
 - qos.h, [528](#)
- pack_qos_SLQSQosSmiReadDataStats_t, [201](#)
 - apnId, [201](#)
- pack_qos_SLQSSetQosEventCallback
 - qos.h, [529](#)
- pack_qos_SLQSSetQosEventCallback_t, [201](#)
 - enable, [202](#)
- pack_sms_SLQSDeleteSMS

- sms.h, 538
- pack_sms_SLQSDDeleteSMS_t, 203
 - pMessageIndex, 203
 - pMessageMode, 204
 - pMessageTag, 204
 - storageType, 204
- pack_sms_SLQSGetSMS
 - sms.h, 538
- pack_sms_SLQSGetSMS_t, 204
 - messageIndex, 204
 - pMessageMode, 204
 - storageType, 204
- pack_sms_SLQSGetSMSList
 - sms.h, 538
- pack_sms_SLQSGetSMSList_t, 204
 - pMessageMode, 205
 - pRequestedTag, 205
 - storageType, 205
- pack_sms_SLQSModifySMSStatus
 - sms.h, 539
- pack_sms_SLQSModifySMSStatus_t, 205
 - messageIndex, 206
 - messageTag, 206
 - pMessageMode, 206
 - storageType, 206
- pack_sms_SendSMS
 - sms.h, 537
- pack_sms_SendSMS_t, 202
 - messageFormat, 202
 - messageSize, 202
 - pLinktimer, 202
 - pMessage, 202
- pack_sms_SetNewSMSCallback
 - sms.h, 537
- pack_sms_SetNewSMSCallback_t, 202
 - status, 203
- pack_swiloc_SwiLocGetAutoStart
 - swiloc.h, 544
- pack_swiloc_SwiLocSetAutoStart
 - swiloc.h, 544
- pack_swiloc_SwiLocSetAutoStart_t, 206
 - fix_rate, 207
 - fix_type, 207
 - function, 207
 - max_dist, 207
 - max_time, 207
 - set_fix_rate, 207
 - set_fix_type, 207
 - set_function, 207
 - set_max_dist, 207
 - set_max_time, 207
- pack_swioma_SLQSOMADMAAlertCallback
 - swioma.h, 546
- pack_swioma_SLQSOMADMCancelSession
 - swioma.h, 547
- pack_swioma_SLQSOMADMCancelSession_t, 207
 - sessionType, 208
- pack_swioma_SLQSOMADMGetSessionInfo
 - swioma.h, 547
- pack_swioma_SLQSOMADMGetSessionInfo_t, 208
 - SessionType, 208
- pack_swioma_SLQSOMADMGetSettings
 - swioma.h, 548
- pack_swioma_SLQSOMADMSelectSelection
 - swioma.h, 548
- pack_swioma_SLQSOMADMSelectSelection_t, 208
 - pDeferTime, 209
 - pRejectReason, 209
 - selection, 209
- pack_swioma_SLQSOMADMSetSettings
 - swioma.h, 549
- pack_swioma_SLQSOMADMSetSettings_t, 209
 - FOTAdownload, 210
 - pAutosdm, 210
 - pFwAutoCheck, 210
- pack_swioma_SLQSOMADMStartSession
 - swioma.h, 550
- pack_swioma_SLQSOMADMStartSession_t, 210
 - sessionType, 210
- pack_uim_ChangePin
 - uim.h, 556
- pack_uim_ChangePin_t, 211
 - changePIN, 211
 - EncryptedPIN1, 211
 - pIndicationToken, 211
 - pKeyReferenceID, 211
 - sessionInfo, 211
 - Tlvresult, 211
- pack_uim_GetCardStatus
 - uim.h, 556
- pack_uim_ReadTransparent
 - uim.h, 557
- pack_uim_ReadTransparent_t, 212
 - fileIndex, 212
 - pEncryptData, 212
 - pIndicationToken, 212
 - readTransparent, 212
 - sessionInfo, 213
 - Tlvresult, 213
- pack_uim_SLQSUIEventRegister
 - uim.h, 557
- pack_uim_SLQSUIEventRegister_t, 214
 - eventMask, 214
- pack_uim_SLQSUIMGetSlotsStatus
 - uim.h, 558
- pack_uim_SLQSUIMSwitchSlot
 - uim.h, 558
- pack_uim_SLQSUIMSwitchSlot_t, 214
 - bLogicalSlot, 215
 - ulPhysicalSlot, 215
- pack_uim_SetPinProtection
 - uim.h, 557
- pack_uim_SetPinProtection_t, 213
 - EncryptedPIN1, 213
 - pIndicationToken, 213
 - pKeyReferenceID, 214

- pinProtection, [213](#)
- sessionInfo, [214](#)
- Tlvresult, [214](#)
- pack_uim_UnblockPin
 - uim.h, [558](#)
- pack_uim_UnblockPin_t, [215](#)
 - EncryptedPIN1, [216](#)
 - pIndicationToken, [216](#)
 - pKeyReferenceID, [216](#)
 - pinProtection, [216](#)
 - sessionInfo, [216](#)
 - Tlvresult, [216](#)
- pack_uim_VerifyPin
 - uim.h, [559](#)
- pack_uim_VerifyPin_t, [216](#)
 - pEncryptedPIN1, [217](#)
 - pIndicationToken, [217](#)
 - pKeyReferenceID, [217](#)
 - sessionInfo, [217](#)
 - Tlvresult, [217](#)
 - verifyPIN, [217](#)
- pack_wds_GetConnectionRate
 - wds.h, [567](#)
- pack_wds_GetDefaultProfile
 - wds.h, [568](#)
- pack_wds_GetDefaultProfile_t, [217](#)
 - profiletype, [217](#)
- pack_wds_GetDefaultProfileNum
 - wds.h, [568](#)
- pack_wds_GetDefaultProfileNum_t, [217](#)
 - family, [218](#)
 - type, [218](#)
- pack_wds_GetDormancyState
 - wds.h, [569](#)
- pack_wds_GetDormancyState_t, [218](#)
- pack_wds_GetLastMobileIPError
 - wds.h, [569](#)
- pack_wds_GetLastMobileIPError_t, [218](#)
- pack_wds_GetMobileIP
 - wds.h, [569](#)
- pack_wds_GetMobileIP_t, [218](#)
- pack_wds_GetMobileIPProfile
 - wds.h, [570](#)
- pack_wds_GetMobileIPProfile_t, [218](#)
 - index, [218](#)
- pack_wds_GetPacketStatus
 - wds.h, [570](#)
- pack_wds_GetPacketStatus_t, [218](#)
 - statmask, [219](#)
- pack_wds_GetSessionDuration
 - wds.h, [571](#)
- pack_wds_GetSessionDuration_t, [219](#)
- pack_wds_GetSessionState
 - wds.h, [571](#)
- pack_wds_RMSetTransferStatistics
 - wds.h, [572](#)
- pack_wds_RMSetTransferStatistics_t, [219](#)
 - RmTrasnferStaticsReq, [219](#)
- pack_wds_SLQSCreateProfile
 - wds.h, [573](#)
- pack_wds_SLQSCreateProfile_t, [222](#)
 - pCurProfile, [222](#)
 - pProfileId, [222](#)
 - pProfileType, [222](#)
- pack_wds_SLQSDeleteProfile
 - wds.h, [573](#)
- pack_wds_SLQSDeleteProfile_t, [222](#)
 - profileIndex, [223](#)
 - profileType, [223](#)
- pack_wds_SLQSGet3GPPConfigItem
 - wds.h, [574](#)
- pack_wds_SLQSGetCurrDataSystemStat
 - wds.h, [574](#)
- pack_wds_SLQSGetCurrDataSystemStat_t, [223](#)
- pack_wds_SLQSGetDUNCallInfo
 - wds.h, [575](#)
- pack_wds_SLQSGetDUNCallInfo_t, [223](#)
 - Mask, [223](#)
 - pReportChannelRate, [223](#)
 - pReportConnStatus, [223](#)
 - pReportDataBearerTech, [223](#)
 - pReportDormStatus, [223](#)
 - pTransferStatInd, [223](#)
- pack_wds_SLQSGetDataBearerTechnology
 - wds.h, [575](#)
- pack_wds_SLQSGetDataBearerTechnology_t, [223](#)
- pack_wds_SLQSGetProfileSettings
 - wds.h, [576](#)
- pack_wds_SLQSGetProfileSettings_t, [224](#)
 - ProfileId, [224](#)
 - ProfileType, [224](#)
- pack_wds_SLQSGetRuntimeSettings
 - wds.h, [576](#)
- pack_wds_SLQSGetRuntimeSettings_t, [224](#)
 - pReqSettings, [225](#)
- pack_wds_SLQSModifyProfile
 - wds.h, [576](#)
- pack_wds_SLQSModifyProfile_t, [225](#)
 - curProfile, [226](#)
 - pProfileId, [226](#)
 - pProfileType, [226](#)
- pack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [578](#)
- pack_wds_SLQSSetDHCPv4ClientConfig_t, [228](#)
 - pProfileId, [228](#)
- pack_wds_SLQSSet3GPPConfigItem
 - wds.h, [577](#)
- pack_wds_SLQSSet3GPPConfigItem_t, [226](#)
 - _3gppRelease, [227](#)
 - defaultPDNEnabled, [227](#)
 - profileList, [227](#)
- pack_wds_SLQSSetIPFamilyPreference
 - wds.h, [577](#)
- pack_wds_SLQSSetIPFamilyPreference_t, [227](#)
 - IPFamilyPreference, [227](#)
- pack_wds_SLQSSetWdsEventCallback

- wds.h, 578
- pack_wds_SLQSSetWdsEventCallback_t, 227
 - currentDataBearer, 228
 - dataBearer, 228
 - dataSystemStatus, 228
 - dormancyStatus, 228
 - interval, 228
 - mobileIP, 228
 - transferStats, 228
- pack_wds_SLQSStartDataSession
 - wds.h, 578
- pack_wds_SLQSStartDataSession_t, 228
 - pAuth, 229
 - pPass, 229
 - pTech, 229
 - pUser, 230
 - pprofileid3gpp, 229
 - pprofileid3gpp2, 229
- pack_wds_SLQSStopDataSession
 - wds.h, 579
- pack_wds_SLQSStopDataSession_t, 230
 - psid, 230
- pack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, 579
- pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 230
 - contextId, 230
 - contextType, 230
- pack_wds_SetDefaultProfile
 - wds.h, 572
- pack_wds_SetDefaultProfile_t, 219
 - authentication, 220
 - ipAddress, 220
 - pApnname, 220
 - pName, 220
 - pPassword, 220
 - pUsername, 220
 - pdpType, 220
 - primaryDNS, 220
 - profileType, 220
 - secondaryDNS, 220
- pack_wds_SetDefaultProfileNum
 - wds.h, 572
- pack_wds_SetDefaultProfileNum_t, 220
 - family, 220
 - index, 220
 - type, 220
- pack_wds_SetMobileIPProfile
 - wds.h, 573
- pack_wds_SetMobileIPProfile_t, 220
 - index, 221
 - pAAASPI, 221
 - pAddress, 221
 - pEnabled, 221
 - pHASPI, 221
 - pMNAAA, 221
 - pMNHA, 221
 - pNAI, 221
 - pPrimaryHA, 221
 - pRevTunneling, 221
 - pSecondaryHA, 221
 - spc, 222
- PackCreateProfileOut, 230
 - ExtErrorCode, 230
 - ProfileIndex, 230
 - ProfileType, 231
- package_name
 - unpack_omaDmFotaTlv_t, 331
- packageid_str
 - unpack_dms_GetFirmwareInfo_t, 263
- packetZone
 - nas_CDMASysInfo, 87
- packetZoneValid
 - nas_CDMASysInfo, 87
- packgetDyingGaspCfg, 231
 - pDestSMSContent, 231
 - pDestSMSNum, 231
- packgetDyingGaspStatistics, 231
 - pSMSAttemptedFlag, 231
 - pTimeStamp, 231
- path
 - uim_fileInfo, 249
- pathLen
 - uim_fileInfo, 249
- pci
 - nas_cellParams, 88
 - nas_PhyCaAggPcellInfo, 127
 - nas_PhyCaAggScellIndType, 129
 - nas_PhyCaAggScellInfo, 132
 - nas_umtsLTENbrCell, 153
 - NASPhyCaAggPcellInfo, 164
 - NASPhyCaAggScellIndType, 166
 - NASPhyCaAggScellInfo, 167
- pcsfFQDNAddress
 - wds_PCSCFFQDNAddressList, 393
- pdpType
 - pack_wds_SetDefaultProfile_t, 220
- pdptype
 - unpack_wds_GetDefaultProfile_t, 370
- peakRate
 - unpack_qos_tokenBucket_t, 350
- peakThroughputClass
 - LibPackGPRSRequestedQoS, 39
 - wds_GPRSQoS, 391
- PersistentTechPref
 - unpack_nas_GetNetworkPreference_t, 301
- persoFeature
 - appStats, 25
 - uim_appStatus, 245
- persoRetries
 - appStats, 25
 - uim_appStatus, 245
- persoState
 - appStats, 25
 - uim_appStatus, 245
- persoUnblockRetries
 - appStats, 25

- uim_appStatus, 245
- PhyCaAggPcellInfo
 - NasGetLTECphyCaInfo, 161
- PhyCaAggScellIDBw
 - NasGetLTECphyCaInfo, 161
- PhyCaAggScellIndType
 - NasGetLTECphyCaInfo, 161
- PhyCaAggScellIndex
 - NasGetLTECphyCaInfo, 161
- PhyCaAggScellInfo
 - NasGetLTECphyCaInfo, 161
- pin1Len
 - uim_encryptedPIN1, 249
- pin1Retries
 - appStats, 26
 - uim_appStatus, 245
- pin1State
 - appStats, 26
 - uim_appStatus, 245
- pin1Val
 - uim_encryptedPIN1, 249
- pin2Retries
 - appStats, 26
 - uim_appStatus, 245
- pin2State
 - appStats, 26
 - uim_appStatus, 245
- pinID
 - uim_changeUIMPIN, 248
 - uim_setPINProtection, 253
 - uim_unblockUIMPIN, 257
 - uim_verifyUIMPIN, 257
- pinLen
 - uim_changeUIMPIN, 248
 - uim_verifyUIMPIN, 257
- pinLength
 - uim_setPINProtection, 253
- pinOperation
 - uim_setPINProtection, 253
- pinProtection
 - pack_uim_SetPinProtection_t, 213
 - pack_uim_UnblockPin_t, 216
- pinVal
 - uim_changeUIMPIN, 248
 - uim_verifyUIMPIN, 257
- pinValue
 - uim_setPINProtection, 253
- PkgDescLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- PkgDescription
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- PkgName
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- PkgNameLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- pkgver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 278
- PktErrRate
 - unpack_qos_swiQosFlow_t, 350
- plmn
 - nas_GERANInfo, 97
 - nas_LTEInfoIntrafreq, 115
 - nas_UMTSInfo, 151
- port
 - unpack_qos_Port_t, 335
- pprofileid3gpp
 - pack_wds_SLQSStartDataSession_t, 229
- pprofileid3gpp2
 - pack_wds_SLQSStartDataSession_t, 229
- prDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 386
- prDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 386
- prPCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- prPCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- Precedence
 - unpack_qos_swiQosFilter_t, 346
- precedenceClass
 - LibPackGPRSRequestedQoS, 39
 - wds_GPRSQoS, 391
- prefNetwork
 - unpack_wds_SLQSGetCurrDataSystemStat_t, 377
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- Preferred
 - nas_QmiNas3GppNetworkInfo, 133
- prefixLen
 - unpack_qos_IPv6Addr_t, 333
- priSize
 - unpack_dms_GetFirmwareRevisions_t, 264
- pridns
 - unpack_wds_GetDefaultProfile_t, 370
- pridnsv6
 - unpack_wds_GetDefaultProfile_t, 370
- primaryDNS
 - pack_wds_SetDefaultProfile_t, 220
- PrimaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- PrimaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- primaryHA
 - unpack_wds_GetMobileIPProfile_t, 373
- priver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 278

- priversion_str
 - unpack_dms_GetFirmwareInfo_t, 263
- ProfileID
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- ProfileId
 - pack_wds_SLQSGetProfileSettings_t, 224
- profileId
 - wdsDhcpv4ProfileId, 399
- ProfileId3GPP2
 - unpack_qos_swiQosFlow_t, 350
- ProfileIndex
 - PackCreateProfileOut, 230
- profileIndex
 - pack_wds_SLQSDeleteProfile_t, 223
 - wds_ProfileIdentifier, 394
- profileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 227
 - unpack_wds_SLQSGet3GPPConfigItem_t, 377
- ProfileName
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- ProfileType
 - pack_wds_SLQSGetProfileSettings_t, 224
 - PackCreateProfileOut, 231
 - unpack_wds_SLQSGetProfileSettings_t, 379
- profileType
 - pack_wds_SetDefaultProfile_t, 220
 - pack_wds_SLQSDeleteProfile_t, 223
 - wds_ProfileIdentifier, 394
 - wdsDhcpv4ProfileId, 399
- profiletype
 - pack_wds_GetDefaultProfile_t, 217
- Protocol
 - unpack_nas_GetCDMANetworkParameters_t, 299
- psAttachState
 - nas_servSystem, 139
 - NASServingSystemInfo, 170
- psBarStatus
 - nas_CallBarringSysInfo, 80
 - nas_callBarStatus, 81
- psState
 - nas_CommInfo, 90
- psc
 - nas_UMTSInfo, 151
 - nas_wcdmaCellInfo, 155
 - nas_WCDMASysInfo, 160
- pscValid
 - nas_WCDMASysInfo, 160
- pscsfIPv4Addr
 - wds_PCSCFIPv4ServerAddressList, 393
- psid
 - pack_wds_SLQSStopDataSession_t, 230
 - unpack_wds_SLQSGetDataSession_t, 385
- puk1Retries
 - appStats, 26
 - uim_appStatus, 245
- puk2Retries
 - appStats, 26
 - uim_appStatus, 245
- pukLen
 - uim_unblockUIMPIN, 257
- pukVal
 - uim_unblockUIMPIN, 257
- QCI
 - LibPackQosClassID, 62
- QFlowState
 - unpack_qos_QosFlowInfo_t, 336
- QMI pack/unpack (pack), 19
- qaGobiApiTableBandClasses.h, 481
- qaGobiApiTableCallControlReturnReasons.h, 484
- qaGobiApiTableCallEndReasons.h, 484
- qaGobiApiTableCarrierCodes.h, 500
- qaGobiApiTableCodingScheme.h, 502
- qaGobiApiTableGpsCapabilityCodes.h, 504
- qaGobiApiTablePowerModes.h, 505
- qaGobiApiTableRadioInterfaces.h, 506
- qaGobiApiTableRegionCodes.h, 506
- qaGobiApiTableServiceOptions.h, 507
- qaGobiApiTableSupServiceInfoClasses.h, 509
- qaGobiApiTableSwiAudio.h, 509
- qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 510
- qaGobiApiTableVoiceCallEndReasons.h, 511
- qm_wds_ds_profile_extended_err_codes
 - qmerrno.h, 524
- qmerrno.h
 - eQCWWAN_ERR_API_MUTEX_TIMEOUT, 521
 - eQCWWAN_ERR_BUFFER_SZ, 520
 - eQCWWAN_ERR_CANCEL_OP, 521
 - eQCWWAN_ERR_DRIVER, 521
 - eQCWWAN_ERR_ENUM_BEGIN, 520
 - eQCWWAN_ERR_ENUM_END, 521
 - eQCWWAN_ERR_FILE_COPY, 520
 - eQCWWAN_ERR_FILE_OPEN, 520
 - eQCWWAN_ERR_GENERAL, 520
 - eQCWWAN_ERR_INTERNAL, 520
 - eQCWWAN_ERR_INVALID_ARG, 520
 - eQCWWAN_ERR_INVALID_DEVID, 520
 - eQCWWAN_ERR_INVALID_FILE, 520
 - eQCWWAN_ERR_INVALID_QMI_RSP, 520
 - eQCWWAN_ERR_INVALID_XID, 521
 - eQCWWAN_ERR_MALFORMED_QMI_RSP, 520
 - eQCWWAN_ERR_MEMORY, 520
 - eQCWWAN_ERR_MULTIPLE_DEVICES, 521
 - eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPOR-
TED, 521
 - eQCWWAN_ERR_NO_CANCELABLE_OP, 521
 - eQCWWAN_ERR_NO_CONNECTION, 520
 - eQCWWAN_ERR_NO_DEVICE, 520
 - eQCWWAN_ERR_NO_SIGNAL, 521
 - eQCWWAN_ERR_NONE, 520
 - eQCWWAN_ERR_NULL_TLV, 524
 - eQCWWAN_ERR_OFFLINE, 521
 - eQCWWAN_ERR_PDU_GENERATION, 521
 - eQCWWAN_ERR_QMI_ABORTED, 521
 - eQCWWAN_ERR_QMI_ACCESS_DENIED, 523
 - eQCWWAN_ERR_QMI_ACK_NOT_SENT, 523

- eQCWWAN_ERR_QMI_ARG_TOO_LONG, [521](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, [522](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, [522](#)
- eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, [523](#)
- eQCWWAN_ERR_QMI_CALL_FAILED, [521](#)
- eQCWWAN_ERR_QMI_CARD_BUSY_RSP, [524](#)
- eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, [523](#)
- eQCWWAN_ERR_QMI_CAT_END, [524](#)
- eQCWWAN_ERR_QMI_CAT_START, [524](#)
- eQCWWAN_ERR_QMI_CAUSE_CODE, [522](#)
- eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, [521](#)
- eQCWWAN_ERR_QMI_CONNECT, [520](#)
- eQCWWAN_ERR_QMI_DEVICE_IN_USE, [521](#)
- eQCWWAN_ERR_QMI_DEVICE_NOT_READY, [522](#)
- eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, [522](#)
- eQCWWAN_ERR_QMI_DISABLED, [522](#)
- eQCWWAN_ERR_QMI_ENCODING, [522](#)
- eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, [524](#)
- eQCWWAN_ERR_QMI_EVENT_REG_FAILED, [524](#)
- eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, [523](#)
- eQCWWAN_ERR_QMI_FDN_RESTRICT, [523](#)
- eQCWWAN_ERR_QMI_FLOW_SUSPENDED, [522](#)
- eQCWWAN_ERR_QMI_GENERAL, [522](#)
- eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, [523](#)
- eQCWWAN_ERR_QMI_IFACE, [520](#)
- eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, [523](#)
- eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, [522](#)
- eQCWWAN_ERR_QMI_INCORRECT_PIN, [521](#)
- eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, [523](#)
- eQCWWAN_ERR_QMI_INJECT_TIMEOUT, [523](#)
- eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, [522](#)
- eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, [522](#)
- eQCWWAN_ERR_QMI_INTERNAL, [521](#)
- eQCWWAN_ERR_QMI_INVALID_ARG, [522](#)
- eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, [521](#)
- eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, [522](#)
- eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, [524](#)
- eQCWWAN_ERR_QMI_INVALID_HANDLE, [521](#)
- eQCWWAN_ERR_QMI_INVALID_ID, [522](#)
- eQCWWAN_ERR_QMI_INVALID_INDEX, [522](#)
- eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, [522](#)
- eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, [522](#)
- eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, [522](#)
- eQCWWAN_ERR_QMI_INVALID_OPERATION, [523](#)
- eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PINID, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, [521](#)
- eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, [522](#)
- eQCWWAN_ERR_QMI_INVALID_QMI_CMD, [523](#)
- eQCWWAN_ERR_QMI_INVALID_QOS_ID, [522](#)
- eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, [522](#)
- eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, [521](#)
- eQCWWAN_ERR_QMI_INVALID_TECH_PREF, [521](#)
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, [524](#)
- eQCWWAN_ERR_QMI_INVALID_TRANSITION, [522](#)
- eQCWWAN_ERR_QMI_INVALID_TX_ID, [521](#)
- eQCWWAN_ERR_QMI_MALFORMED_MSG, [521](#)
- eQCWWAN_ERR_QMI_MAX, [523](#)
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, [522](#)
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, [522](#)
- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, [522](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, [522](#)
- eQCWWAN_ERR_QMI_MISSING_ARG, [521](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED, [523](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED, [523](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY, [522](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAVAILABLE, [522](#)
- eQCWWAN_ERR_QMI_NO_EFFECT, [521](#)
- eQCWWAN_ERR_QMI_NO_ENTRY, [522](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE, [521](#)
- eQCWWAN_ERR_QMI_NO_MEMORY, [521](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, [521](#)
- eQCWWAN_ERR_QMI_NO_RADIO, [523](#)

- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, [523](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS, [521](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, [522](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED, [521](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED, [523](#)
- eQCWWAN_ERR_QMI_OFFSET, [521](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUP-
PORTED, [521](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUP-
PORTED, [521](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, [523](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL, [521](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED, [522](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, [522](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH, [523](#)
- eQCWWAN_ERR_QMI_REQ, [520](#)
- eQCWWAN_ERR_QMI_REQ_SCH, [520](#)
- eQCWWAN_ERR_QMI_REQ_TO, [520](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UN-
SUPPORTED, [522](#)
- eQCWWAN_ERR_QMI_RSP, [520](#)
- eQCWWAN_ERR_QMI_RSP_TO, [520](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER, [523](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, [523](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE, [522](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID, [522](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, [522](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, [523](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, [522](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR, [523](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUS-
E, [523](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE, [523](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRAN-
SACTION, [521](#)
- eQCWWAN_ERR_QMI_UNKNOWN, [522](#)
- eQCWWAN_ERR_QMI_WIDTH, [524](#)
- eQCWWAN_ERR_RESET, [521](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR, [523](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRES-
S, [523](#)
- eQCWWAN_ERR_SWICM_END, [524](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_S-
DK_PROCESS, [523](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_
ID, [523](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSI-
ON_ID, [523](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSI-
ON_ID, [523](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, [523](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_S-
UPPORTED, [523](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SU-
PPORTED, [523](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_
SESSIONS, [523](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE, [523](#)
- eQCWWAN_ERR_SWICM_START, [523](#)
- eQCWWAN_ERR_SWICM_TIMEOUT, [523](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN, [523](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP, [523](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN, [523](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP, [523](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, [524](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_F-
OUND, [524](#)
- eQCWWAN_ERR_SWIDCS_END, [524](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR, [524](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR, [524](#)
- eQCWWAN_ERR_SWIDCS_START, [524](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_I-
MAGE, [524](#)
- eQCWWAN_ERR_SWIIM_END, [524](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, [524](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_D-
OWNLOADED, [524](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNL-
OAD_MODE, [524](#)
- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPL-
ETE, [524](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_
MISMATCH, [524](#)
- eQCWWAN_ERR_SWIIM_FW_SAME_AS_CUR-
RENT_ACTIVE_IMAGE, [524](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, [524](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCC-
ESS, [524](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REB-
OOT, [524](#)
- eQCWWAN_ERR_SWIIM_INVALID_CRASH_ST-
ATE, [524](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH, [524](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR, [524](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE, [524](#)
- eQCWWAN_ERR_SWIIM_START, [524](#)
- eQCWWAN_ERR_SWISM_END, [524](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_N-
OT_FOUND, [524](#)

- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, [524](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, [524](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, [524](#)
- eQCWWAN_ERR_SWISMS_START, [524](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, [525](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, [525](#)
- eWDS_ERR_PROFILE_REG_END, [525](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL, [525](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END, [525](#)
- qmerrno.h, [518](#)
 - eQCWWANError, [520](#)
 - qm_wds_ds_profile_extended_err_codes, [524](#)
- qmiSmsMessageList, [232](#)
 - messageIndex, [232](#)
 - messageTag, [232](#)
- qmiWDSDataBearerTechnology, [232](#)
 - currentNetwork, [232](#)
 - ratMask, [232](#)
 - soMask, [232](#)
- qos.h, [525](#)
 - pack_qos_SLQSQosGetNetworkStatus, [527](#)
 - pack_qos_SLQSQosSmiReadApnExtraParams, [527](#)
 - pack_qos_SLQSQosSmiReadDataStats, [528](#)
 - pack_qos_SLQSSetQosEventCallback, [529](#)
 - unpack_qos_SLQSQosGetNetworkStatus, [529](#)
 - unpack_qos_SLQSQosSmiReadApnExtraParams, [530](#)
 - unpack_qos_SLQSQosSmiReadDataStats, [530](#)
 - unpack_qos_SLQSSetQosEventCallback, [531](#)
 - unpack_qos_SLQSSetQosEventCallback_ind, [531](#)
 - unpack_qos_SLQSSetQosNWStatusCallback_ind, [531](#)
 - unpack_qos_SLQSSetQosPriEventCallback_ind, [532](#)
 - unpack_qos_SLQSSetQosStatusCallback_ind, [533](#)
- qosDeliveryOrder
 - LibPackUMTSQoS, [66](#)
 - wds_UMTSMinQoS, [396](#)
- qosFlow
 - unpack_qos_SLQSQosSmiReadDataStats_t, [340](#)
- QosFlowInfo
 - unpack_qos_SLQSSetQosEventCallback_ind_t, [340](#)
- RAN
 - unpack_nas_GetServingNetwork_t, [302](#)
- RAT
 - nas_QmiNas3GppNetworkRAT, [134](#)
- RATMask
 - currNetworkInfo, [28](#)
 - wds_currNetworkInfo, [389](#)
- RFBandInfoElements, [232](#)
 - activeBandClass, [233](#)
 - activeChannel, [233](#)
 - radioInterface, [233](#)
 - unpack_nas_GetRFInfo_t, [301](#)
- RFTlv
 - unpack_nas_SetEventReportInd_t, [305](#)
- RMAutoConnect
 - pack_dms_SetCustFeature_t, [172](#)
 - unpack_dms_GetCustFeature_t, [259](#)
- RPTlv
 - NASQmiCbkNasSystemSelPrefInd, [168](#)
- RRTlv
 - unpack_nas_SetEventReportInd_t, [305](#)
- RXChan
 - nas_LTEInfo, [112](#)
- rXDroppedCount
 - unpack_wds_GetPacketStatus_t, [374](#)
- rXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, [374](#)
- rXOkBytesCount
 - unpack_wds_GetPacketStatus_t, [374](#)
- rXPacketErrors
 - unpack_wds_GetPacketStatus_t, [374](#)
- rXPacketOverflows
 - unpack_wds_GetPacketStatus_t, [374](#)
- rXPacketSuccesses

- unpack_wds_GetPacketStatus_t, 374
- radio
 - unpack_nas_GetSignalStrengths_t, 303
- radiolf
 - nas_ecioListElement, 94
 - nas_errorRateListElement, 95
 - nas_rsrqInformation, 137
 - nas_rxSignalStrengthListElement, 138
- Radiolfaces
 - unpack_dms_GetDeviceCap_t, 260
 - unpack_dms_GetDeviceCapabilities_t, 261
 - unpack_nas_GetServingNetwork_t, 302
- RadiolfacesSize
 - unpack_dms_GetDeviceCap_t, 260
 - unpack_nas_GetServingNetwork_t, 302
- radiolfacesSize
 - unpack_dms_GetDeviceCapabilities_t, 261
- radiolInterface
 - nas_RFInfoTlv, 136
 - nas_roamIndList, 136
 - nas_servSystem, 139
 - nas_SignalStrengthTlv, 140
 - nas_timeInfo, 149
 - RFBandInfoElements, 233
- radiolInterfaceList
 - NASServingSystemInfo, 170
- radiolInterfaceNo
 - NASServingSystemInfo, 170
- radiolInterfaceSize
 - nas_RFInfoTlv, 136
- range
 - unpack_qos_Port_t, 335
- rat
 - nas_CSGID, 91
 - nas_MNRInfo, 125
- ratMask
 - qmiWSDDataBearerTechnology, 232
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- readTransparent
 - pack_uim_ReadTransparent_t, 212
- reason
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 342
- reconfigReqd
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 383
- refData
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 279
- refString
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 279
- refpn
 - nas_CDMAInfo, 83
- regAction
 - pack_nas_SLQSInitiateNetworkRegistration_t, 186
- RegForeignNID
 - unpack_nas_GetCDMANetworkParameters_t, 299
- RegForeignSID
 - unpack_nas_GetCDMANetworkParameters_t, 299
- RegHomeSID
 - unpack_nas_GetCDMANetworkParameters_t, 299
- regPrd
 - nas_AddCDMASysInfo, 79
- regRejectInfoValid
 - nas_GSMSysInfo, 103
 - nas_LTESysInfo, 123
 - nas_WCDMASysInfo, 160
- regState
 - nas_servSystem, 139
- RegistrationState
 - unpack_nas_GetServingNetwork_t, 302
- registrationState
 - NASServingSystemInfo, 170
- rejCause
 - nas_GSMSysInfo, 103
 - nas_LTESysInfo, 123
 - nas_WCDMASysInfo, 160
- rejectCause
 - nas_RejectReasonTlv, 135
- rejectSrvDomain
 - nas_GSMSysInfo, 104
 - nas_LTESysInfo, 123
 - nas_WCDMASysInfo, 160
- reliabilityClass
 - LibPackGPRSRequestedQoS, 39
 - wds_GPRSQoS, 391
- resBerRatio
 - LibPackUMTSQoS, 66
 - wds_UMTSMInQoS, 396
- ResCode
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 279
- resetInfoInd
 - pack_dms_SLQSDmsSwiIndicationRegister_t, 174
- RetryCount
 - unpack_swioama_SLQSOMADMGetSessionInfo_t, 360
- revTunneling
 - unpack_wds_GetMobileIPProfile_t, 373
- RmTrasnferStaticsReq
 - pack_wds_RMSetTransferStatistics_t, 219
- rmTrasnferStaticsReq, 233
 - bResetStatistics, 233
 - ulMask, 233
- roamIndicator
 - nas_roamIndList, 136
- RoamIndicatorVal
 - unpack_nas_SLQSGetServingSystem_t, 311
- RoamPref
 - NASRoamPreferenceTlv, 168
- roamStatus
 - nas_sysInfoCommon, 145
- roamStatusValid
 - nas_sysInfoCommon, 145
- Roaming
 - nas_QmiNas3GppNetworkInfo, 133
 - unpack_nas_GetCDMANetworkParameters_t, 299

- unpack_nas_GetServingNetwork_t, 302
- roaming
 - unpack_nas_SetRoamingIndicatorCallback_ind_t, 307
- RoamingIndicatorList
 - unpack_nas_SLQSGetServingSystem_t, 311
- rptRate
 - nas_LTESigRptConfig, 119
- rscp
 - nas_UMTSInfo, 151
 - tdscdmaSigInfoExt, 241
- rsrp
 - lteSSInfo, 77
 - nas_cellParams, 88
 - nas_umtsLTENbrCell, 153
- rsrplevel
 - nas_lteRsrpinformation, 117
- rsrq
 - lteSSInfo, 77
 - nas_cellParams, 88
 - nas_rsrqInformation, 137
 - nas_umtsLTENbrCell, 153
- rsrqDelta
 - nas_SLQSSignalStrengthsIndReq, 141
- rsrqInfo
 - nas_SLQSSignalStrengthsInformation, 142
 - unpack_nas_SLQSGetSignalStrength_t, 312
- rssti
 - cdmaSSInfo, 27
 - hdrSSInfo, 37
 - lteSSInfo, 77
 - nas_cellParams, 88
 - nas_gsmCellInfo, 99
 - tdscdmaSigInfoExt, 241
 - unpack_nas_GetSignalStrengths_t, 303
- rx_bytes
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- rx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- rxLev
 - nas_GERANInfo, 97
- rxOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, 379
- RxQFilter
 - unpack_qos_QosFlowInfo_t, 336
- RxQFlowGranted
 - unpack_qos_QosFlowInfo_t, 336
- rxSignalStrength
 - nas_rxSignalStrengthListElement, 138
- rxSignalStrengthDelta
 - nas_SLQSSignalStrengthsIndReq, 141
- rxSignalStrengthInfo
 - nas_SLQSSignalStrengthsInformation, 142
- rxSignalStrengthList
 - unpack_nas_SLQSGetSignalStrength_t, 312
- rxSignalStrengthListLen
 - unpack_nas_SLQSGetSignalStrength_t, 312
- SCI
 - unpack_nas_GetCDMANetworkParameters_t, 299
- SCM
 - unpack_nas_GetCDMANetworkParameters_t, 299
- SDPTlv
 - NASQmiCbkNasSystemSelPrefInd, 168
- SDU_HDR_LEN
 - common.h, 403
- SHORT
 - SwiDataTypes.h, 543
- sIntraSearch
 - nas_LTEInfoIntrafreq, 115
- SLQSSSTlv
 - unpack_nas_SetEventReportInd_t, 305
- sMSCAddress, 236
 - data, 236
 - length, 237
- sMSCAddressInfo
 - sms.h, 535
- sMSCAddressTlv, 237
 - SMSCInfo, 237
 - TlvPresent, 237
- SMSCInfo
 - sMSCAddressTlv, 237
- SMSCtlv
 - unpack_sms_SetNewSMSCallback_ind_t, 353
- sMSEtwsMessage, 237
 - data, 237
 - length, 238
 - notificationType, 238
- sMSEtwsMessageInfo
 - sms.h, 535
- sMSEtwsMessageTlv, 238
 - EtwsMessageInfo, 238
 - TlvPresent, 238
- sMSEtwsPlmn, 238
 - mobileCountryCode, 238
 - mobileNetworkCode, 238
- sMSEtwsPlmnInfo
 - sms.h, 536
- sMSMTMessage, 239
 - messageIndex, 239
 - storageType, 239
- sMSMTMessageInfo
 - sms.h, 536
- sMSMessageMode, 239
 - messageMode, 239
- sMSMessageModelInfo
 - sms.h, 536
- sMSOnIMS, 239
 - smsOnIMS, 239
- sMSOnIMSInfo
 - sms.h, 536
- sMSOnIMSTlv, 240
 - IMSInfo, 240
 - TlvPresent, 240
- SMSSupport

- pack_dms_SetCustFeature_t, 172
- unpack_dms_GetCustFeature_t, 259
- sMSTransferRouteMTMessage, 240
 - ackIndicator, 241
 - data, 241
 - format, 241
 - length, 241
 - transactionID, 241
- sMSTransferRouteMTMessageInfo
 - sms.h, 536
- sNonIntraSearch
 - nas_LTEInfoIntrafreq, 115
- SOMask
 - currNetworkInfo, 28
 - wds_currNetworkInfo, 389
- sPhyCaAggPcellInfo
 - unpack_nas_SetNasLTECphyCalndCallback_ind_t, 305
- sPhyCaAggScellDIBw
 - unpack_nas_SetNasLTECphyCalndCallback_ind_t, 305
- sPhyCaAggScellIndType
 - unpack_nas_SetNasLTECphyCalndCallback_ind_t, 306
- sPhyCaAggScellIndex
 - unpack_nas_SetNasLTECphyCalndCallback_ind_t, 306
- sPhyCaAggScellInfo
 - unpack_nas_SetNasLTECphyCalndCallback_ind_t, 306
- SSInfo
 - unpack_nas_SetServingSystemCallback_ind_t, 307
- sSLQSSignalStrengthsInfo
 - nas_SLQSSignalStrengthsTlv, 142
- SSTlv
 - unpack_nas_SetEventReportInd_t, 305
- SWI_API
 - SwiDataTypes.h, 543
- SWIWWANCMAPI.h, 554
- scell_idx
 - nas_PhyCaAggScellIndex, 128
 - NASPhyCaAggScellIndex, 165
- scell_state
 - nas_PhyCaAggScellIndType, 129
 - nas_PhyCaAggScellInfo, 132
 - NASPhyCaAggScellIndType, 166
 - NASPhyCaAggScellInfo, 167
- sduErrorRatio
 - LibPackUMTSQoS, 66
 - wds_UMTSMInQoS, 397
- seDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- seDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- sePCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- sePCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- sePCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 387
- secdns
 - unpack_wds_GetDefaultProfile_t, 370
- secdnsV6
 - unpack_wds_GetDefaultProfile_t, 370
- second
 - nas_timeInfo, 149
 - nas_UniversalTime, 154
- secondaryDNS
 - pack_wds_SetDefaultProfile_t, 220
- SecondaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- SecondaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- secondaryHA
 - unpack_wds_GetMobileIPProfile_t, 373
- selNetwork
 - nas_servSystem, 139
- selectedNetwork
 - NASServingSystemInfo, 170
- selection
 - pack_swioama_SLQSOMADMSendSelection_t, 209
- ServerAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- serviceDomain
 - nas_RejectReasonTlv, 135
- servingCellId
 - nas_LTEInfoIntrafreq, 115
- ServingSystem
 - unpack_nas_SLQSGetServingSystem_t, 311
- sessionEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 383
- SessionId
 - pack_loc_Start_t, 183
 - pack_loc_Stop_t, 183
- sessionId
 - unpack_loc_PositionRpt_Ind_t, 295
- sessionInfo
 - pack_uim_ChangePin_t, 211
 - pack_uim_ReadTransparent_t, 213
 - pack_uim_SetPinProtection_t, 214
 - pack_uim_UnblockPin_t, 216
 - pack_uim_VerifyPin_t, 217
- SessionInfoConfig
 - unpack_swioama_SLQSOMADMAAlertCallback_ind_t, 358
- SessionInfoFota
 - unpack_swioama_SLQSOMADMAAlertCallback_ind_t, 358
- SessionInfoNotification
 - unpack_swioama_SLQSOMADMAAlertCallback_ind_t, 358

- SessionState
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- sessionStatus
 - unpack_loc_PositionRpt_Ind_t, 295
 - unpack_omaDmNotificationsTlv_t, 331
- SessionType
 - pack_swima_SLQSOMADMGetSessionInfo_t, 208
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- sessionType
 - pack_swima_SLQSOMADMCancelSession_t, 208
 - pack_swima_SLQSOMADMStartSession_t, 210
 - uim_sessionInformation, 252
 - uim_UIMSessionInformation, 256
 - unpack_omaDmFotaTlv_t, 331
- set_fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, 207
- set_fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, 207
- set_function
 - pack_swiloc_SwiLocSetAutoStart_t, 207
- set_max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, 207
- set_max_time
 - pack_swiloc_SwiLocSetAutoStart_t, 207
- Severity
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- severity
 - unpack_omaDmFotaTlv_t, 331
- shortName
 - unpack_nas_SLQSGetPLMNName_t, 309
- shortNameCI
 - unpack_nas_SLQSGetPLMNName_t, 309
- shortNameEn
 - unpack_nas_SLQSGetPLMNName_t, 309
- shortNameLen
 - unpack_nas_SLQSGetPLMNName_t, 309
- shortNameSB
 - unpack_nas_SLQSGetPLMNName_t, 309
- sid
 - nas_CDMAInfo, 83
 - unpack_nas_GetHomeNetwork_t, 300
- SigInd
 - LibPackUMTSReqQoSsigInd, 67
- signalStrength
 - nas_SignalStrengthTlv, 140
- signalStrengthReqMask
 - unpack_nas_SLQSGetSignalStrength_t, 312
- SimCapability
 - unpack_dms_GetDeviceCap_t, 260
- simCapability
 - unpack_dms_GetDeviceCapabilities_t, 261
- sinr
 - hdrSSInfo, 37
 - nas_SLQSSignalStrengthsInformation, 142
 - tdscdmaSigInfoExt, 241
 - unpack_nas_SLQSGetSignalStrength_t, 312
- sinrDelta
 - nas_SLQSSignalStrengthsIndReq, 141
- sinrThresholdList
 - nas_SLQSSignalStrengthsIndReq, 141
- sinrThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, 141
- sku_str
 - unpack_dms_GetFirmwareInfo_t, 263
- slot_t, 233
 - bICCID, 234
 - bICCIDLength, 234
 - bLogicalSlot, 234
 - uPhyCardStatus, 234
 - uPhySlotStatus, 234
- slotInf, 234
 - AppStatus, 236
 - cardState, 236
 - errorState, 236
 - numApp, 236
 - upinRetries, 236
 - upinState, 236
 - upukRetries, 236
- SlotInfo
 - uim_cardStatus, 247
- slots_t, 236
 - uimSlotStatus, 236
- slotsstatusChange
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 366
- SlqsProfile3GPP
 - unpackWdsProfileParam, 387
 - wds_profileInfo, 394
- SlqsProfile3GPP2
 - unpackWdsProfileParam, 387
 - wds_profileInfo, 394
- sms.h
 - LIBPACK_QMI_CBK_PARAM_NOCHANGE, 537
 - LIBPACK_QMI_CBK_PARAM_RESET, 537
 - LIBPACK_QMI_CBK_PARAM_SET, 537
- sms.h, 533
 - eqmiCbkSetStatus, 537
 - MAX_MSE_TWS_MSG, 535
 - MAX_SMS_LIST_SIZE, 535
 - pack_sms_SLQSDeleteSMS, 538
 - pack_sms_SLQSGetSMS, 538
 - pack_sms_SLQSGetSMSList, 538
 - pack_sms_SLQSModifySMSStatus, 539
 - pack_sms_SendSMS, 537
 - pack_sms_SetNewSMSCallback, 537
 - sMSCAddressInfo, 535
 - sMSEtwMessageInfo, 535
 - sMSEtwPlmnInfo, 536
 - sMSMTMessageInfo, 536
 - sMSMessageModelInfo, 536
 - sMSOnIMSInfo, 536

- SMSTransferRouteMTMessageInfo, 536
 - unpack_sms_SLQSDDeleteSMS, 540
 - unpack_sms_SLQSGetSMS, 541
 - unpack_sms_SLQSGetSMSList, 541
 - unpack_sms_SLQSModifySMSStatus, 541
 - unpack_sms_SLQSWmsMemoryFullCallback_ind, 542
 - unpack_sms_SendSMS, 539
 - unpack_sms_SetNewSMSCallback, 540
 - unpack_sms_SetNewSMSCallback_ind, 540
- smsOnIMS
 - sMSOnIMS, 239
- snr
 - lteSSInfo, 77
- snrlevel
 - nas_lteSnrinformation, 120
- soMask
 - qmiWSDDataBearerTechnology, 232
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 384
- Source
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- source
 - unpack_dms_GetNetworkTime_t, 266
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 271
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 272
- sourceIPMask
 - LibPackTFTIDParams, 64
- SourceLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- spc
 - pack_nas_SetACCOLC_t, 184
 - pack_wds_SetMobileIPPProfile_t, 222
- spn
 - unpack_nas_SLQSGetPLMNName_t, 309
- spnEncoding
 - unpack_nas_SLQSGetPLMNName_t, 309
- spnLength
 - unpack_nas_SLQSGetPLMNName_t, 309
- srcPortRangeEnd
 - LibPackTFTIDParams, 64
- srcPortRangeStart
 - LibPackTFTIDParams, 64
- srvCapability
 - nas_detailSvcInfo, 94
 - nas_sysInfoCommon, 145
- srvCapabilityValid
 - nas_sysInfoCommon, 145
- srvDomain
 - nas_sysInfoCommon, 145
- SrvDomainPref
 - NASServDomainPrefTlv, 168
- srvDomainValid
 - nas_sysInfoCommon, 145
- srvStatus
 - nas_detailSvcInfo, 94
 - nas_GSMSrvStatusInfo, 101
 - nas_SrvStatusInfo, 143
- srxlev
 - nas_cellParams, 88
 - nas_gsmCellInfo, 99
 - nas_umtsLTENbrCell, 153
 - nas_wcdmaCellInfo, 155
- state
 - unpack_dms_GetActivationState_t, 258
 - unpack_omaDmConfigTlv_t, 329
 - unpack_omaDmFotaTlv_t, 331
 - unpack_qos_QosFlowInfoState_t, 337
- statmask
 - pack_wds_GetPacketStatus_t, 219
- StatsMask
 - transferStatInd, 242
- StatsPeriod
 - transferStatInd, 242
- Status
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 360
- status
 - pack_sms_SetNewSMSCallback_t, 203
 - unpack_loc_BestAvailPos_Ind_t, 288
 - unpack_loc_SetExtPowerConfig_Ind_t, 296
 - unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 341
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 342
- storageIndex
 - FMSImageldElement, 35
- storageType
 - pack_sms_SLQSDDeleteSMS_t, 204
 - pack_sms_SLQSGetSMS_t, 204
 - pack_sms_SLQSGetSMSList_t, 205
 - pack_sms_SLQSModifySMSStatus_t, 206
 - sMSMTMessage, 239
 - unpack_sms_SLQSWmsMemoryFullCallback_ind_t, 355
- String
 - unpack_dms_GetDeviceHardwareRev_t, 261
 - unpack_dms_GetDeviceMfr_t, 261
 - unpack_dms_GetFSN_t, 264
 - unpack_dms_UIMGetICCID_t, 280
- stringSize
 - unpack_dms_GetDeviceHardwareRev_t, 261
 - unpack_dms_GetDeviceMfr_t, 261
 - unpack_dms_UIMGetICCID_t, 280
- subnetMask
 - unpack_qos_IPv4Addr_t, 333
- SubnetMaskV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- SupUSBComps
 - unpack_dms_GetUSBComp_t, 268
- svc
 - pack_qmi_t, 200
- sw1

- uim_cardResult, 246
- sw2
 - uim_cardResult, 246
- SwiDataTypes.h, 542
 - BOOL, 543
 - BYTE, 543
 - CHAR, 543
 - FLOAT, 543
 - INT32, 543
 - INT8, 543
 - LPCSTR, 543
 - SHORT, 543
 - SWI_API, 543
 - ULONG, 543
 - ULONGLONG, 543
 - UNUSEDPARAM, 543
 - USHORT, 543
 - WORD, 543
- swiloc.h, 543
 - pack_swiloc_SwiLocGetAutoStart, 544
 - pack_swiloc_SwiLocSetAutoStart, 544
 - unpack_swiloc_SwiLocGetAutoStart, 544
 - unpack_swiloc_SwiLocSetAutoStart, 545
- swioma.h, 545
 - pack_swioma_SLQSOMADMAAlertCallback, 546
 - pack_swioma_SLQSOMADMCancelSession, 547
 - pack_swioma_SLQSOMADMGetSessionInfo, 547
 - pack_swioma_SLQSOMADMGetSettings, 548
 - pack_swioma_SLQSOMADMSelectSend, 548
 - pack_swioma_SLQSOMADMSetSettings, 549
 - pack_swioma_SLQSOMADMStartSession, 550
 - unpack_swioma_SLQSOMADMAAlertCallback, 550
 - unpack_swioma_SLQSOMADMAAlertCallback_ind, 551
 - unpack_swioma_SLQSOMADMCancelSession, 551
 - unpack_swioma_SLQSOMADMGetSessionInfo, 552
 - unpack_swioma_SLQSOMADMGetSettings, 552
 - unpack_swioma_SLQSOMADMSelectSend, 553
 - unpack_swioma_SLQSOMADMSetSettings, 553
 - unpack_swioma_SLQSOMADMStartSession, 553
- sysInfoCDMA
 - nas_CDMASysInfo, 87
- sysInfoGSM
 - nas_GSMSysInfo, 104
- sysInfoHDR
 - nas_HDRSysInfo, 108
- sysInfoLTE
 - nas_LTESysInfo, 123
- sysInfoWCDMA
 - nas_WCDMASysInfo, 160
- system
 - loc_SV, 75
- SystemID
 - unpack_nas_SLQSGetServingSystem_t, 311
- systemID
 - nas_CDMASysInfo, 87
- systemMode
 - nas_CommInfo, 90
- TCPDstPort
 - unpack_qos_swiQosFilter_t, 346
- TCPsrcPort
 - unpack_qos_swiQosFilter_t, 346
- TDSCDMAECIOThreshListLen
 - nas_TDSCDMAECIOThresh, 146
- TDSCDMARSCPTThreshListLen
 - nas_TDSCDMARSCPTThresh, 146
- TDSCDMARSSIThreshListLen
 - nas_TDSCDMARSSIThresh, 147
- TDSCDMASINRThreshListLen
 - nas_TDSCDMASINRThresh, 148
- TRMessageTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 353
- TXChan
 - nas_LTEInfo, 112
- tXDroppedCount
 - unpack_wds_GetPacketStatus_t, 374
- tXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, 374
- tXOkBytesCount
 - unpack_wds_GetPacketStatus_t, 374
- tXPacketErrors
 - unpack_wds_GetPacketStatus_t, 374
- tXPacketOverflows
 - unpack_wds_GetPacketStatus_t, 374
- tXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, 374
- Tables, 21
- tac
 - nas_LTEInfoIntrafreq, 115
 - nas_LTESysInfo, 123
- tacValid
 - nas_LTESysInfo, 123
- TdsBandCapability
 - unpack_dms_SLQSGetBandCapability_t, 276
- tdscdmaSigInfoExt, 241
 - ecio, 241
 - rscp, 241
 - rsi, 241
 - sinr, 241
- techName
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 383
- Technology
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- TechnologyPref
 - pack_nas_SetNetworkPreference_t, 185
- temperature
 - nas_CommInfo, 90
- threshGsmHigh
 - nas_lteGsmCellInfo, 110
- threshGsmLow
 - nas_lteGsmCellInfo, 110
- threshServingLow

- nas_LTEInfoIntraFreq, 115
- threshXHigh
 - nas_infoInterFreq, 109
- threshXLow
 - nas_infoInterFreq, 109
- threshXhigh
 - nas_lteWcdmaCellInfo, 124
- threshXlow
 - nas_lteWcdmaCellInfo, 124
- Time
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 361
- time
 - NASTimeInfoTlv, 170
- TimeLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 361
- timeTlv
 - NASQmiCbkNasSwiOTAMessageInd, 167
- timeZone
 - nas_timeInfo, 149
- timeout
 - pack_qmi_t, 200
- timestamp
 - unpack_dms_GetNetworkTime_t, 266
- timingAdvance
 - nas_GERANInfo, 97
- TlvPresent
 - dms_ActivationStatusTlv, 29
 - dms_OperatingModeTlv, 29
 - eTWSPLMNInfoTlv, 33
 - messageModeTlv, 78
 - nas_PhyCaAggPcellInfo, 127
 - nas_PhyCaAggScellIDIBw, 128
 - nas_PhyCaAggScellIndex, 128
 - nas_PhyCaAggScellIndType, 129
 - nas_PhyCaAggScellInfo, 132
 - nas_RejectReasonTlv, 135
 - nas_RFInfoTlv, 136
 - nas_SignalStrengthTlv, 140
 - nas_SLQSSignalStrengthsTlv, 142
 - nas_timeInfo, 149
 - NASBandPreferenceTlv, 160
 - NASEmergencyModeTlv, 161
 - NASGWAcqOrderPrefTlv, 161
 - NASLTEBandPreferenceTlv, 162
 - NASLteNasReleaseInfoTlv, 162
 - NASModePreferenceTlv, 162
 - NASNetSelPreferenceTlv, 163
 - NASOTAMessageTlv, 163
 - NASPhyCaAggPcellInfo, 164
 - NASPhyCaAggScellIDIBw, 164
 - NASPhyCaAggScellIndex, 165
 - NASPhyCaAggScellIndType, 166
 - NASPhyCaAggScellInfo, 167
 - NASPRLPreferenceTlv, 167
 - NASRoamPreferenceTlv, 168
 - NASServDomainPrefTlv, 168
 - NASTimeInfoTlv, 170
 - newMTMessageTlv, 171
 - sMSCAddressTlv, 237
 - sMSEtwSMessageTlv, 238
 - sMSOnIMSTlv, 240
 - transferRouteMessageTlv, 242
- Tlvresult
 - pack_dms_GetCustFeaturesV2_t, 171
 - pack_dms_SetCustFeaturesV2_t, 173
 - pack_dms_SetPower_t, 174
 - pack_dms_SetUSBComp_t, 174
 - pack_dms_UIMGetICCID_t, 175
 - pack_fms_GetImagesPreference_t, 175
 - pack_fms_GetStoredImages_t, 176
 - pack_fms_SetImagesPreference_t, 177
 - pack_loc_Delete_Assist_Data_t, 178
 - pack_loc_EventRegister_t, 180
 - pack_loc_SetExtPowerState_t, 180
 - pack_loc_SetOperationMode_t, 181
 - pack_loc_SLQSLOCGetBestAvailPos_t, 181
 - pack_loc_Start_t, 183
 - pack_loc_Stop_t, 183
 - pack_nas_SetNetworkPreference_t, 185
 - pack_uim_ChangePin_t, 211
 - pack_uim_ReadTransparent_t, 213
 - pack_uim_SetPinProtection_t, 214
 - pack_uim_UnblockPin_t, 216
 - pack_uim_VerifyPin_t, 217
 - unpack_dms_GetBandCapability_t, 258
 - unpack_dms_GetCrashAction_t, 258
 - unpack_dms_GetCustFeature_t, 259
 - unpack_dms_GetCustFeaturesV2_t, 260
 - unpack_dms_GetDeviceCap_t, 260
 - unpack_dms_GetDeviceHardwareRev_t, 261
 - unpack_dms_GetDeviceMfr_t, 261
 - unpack_dms_GetDeviceSerialNumbers_t, 262
 - unpack_dms_GetFirmwareInfo_t, 263
 - unpack_dms_GetFirmwareRevision_t, 264
 - unpack_dms_GetFirmwareRevisions_t, 264
 - unpack_dms_GetFSN_t, 264
 - unpack_dms_GetIMSI_t, 265
 - unpack_dms_GetModelID_t, 265
 - unpack_dms_GetNetworkTime_t, 266
 - unpack_dms_GetPower_t, 267
 - unpack_dms_GetPRLVersion_t, 267
 - unpack_dms_GetUSBComp_t, 268
 - unpack_dms_GetVoiceNumber_t, 268
 - unpack_dms_SetCustFeature_t, 269
 - unpack_dms_SetCustFeaturesV2_t, 269
 - unpack_dms_SetEventReport_ind_t, 270
 - unpack_dms_SetEventReport_t, 270
 - unpack_dms_SetFirmwarePreference_t, 270
 - unpack_dms_SetPower_t, 270
 - unpack_dms_SetUSBComp_t, 271
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 271
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 272

- unpack_dms_SLQSDmsSwiIndicationRegister_t, 273
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 276
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, 277
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 277
- unpack_dms_SLQSSwiGetFwUpdateStatus_t, 279
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, 280
- unpack_dms_UIMGetICCID_t, 280
- unpack_fms_GetImagesPreference_t, 281
- unpack_fms_GetStoredImages_t, 282
- unpack_fms_SetImagesPreference_t, 282
- unpack_loc_BestAvailPos_Ind_t, 288
- unpack_loc_Delete_Assist_Data_t, 289
- unpack_loc_EngineState_Ind_t, 289
- unpack_loc_EventRegister_t, 290
- unpack_loc_PositionRpt_Ind_t, 295
- unpack_loc_SetExtPowerConfig_Ind_t, 296
- unpack_loc_SetExtPowerState_t, 297
- unpack_loc_SetOperationMode_t, 297
- unpack_loc_SLQSLOCGetBestAvailPos_t, 297
- unpack_loc_Start_t, 298
- unpack_loc_Stop_t, 298
- unpack_nas_GetNetworkPreference_t, 301
- unpack_nas_SetNetworkPreference_t, 306
- unpack_nas_SetServingSystemCallback_ind_t, 307
- unpack_nas_SlqsGetLTECphyCAInfo_t, 308
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 324
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 324
- unpack_uim_ChangePin_t, 363
- unpack_uim_GetCardStatus_t, 364
- unpack_uim_ReadTransparent_t, 365
- unpack_uim_SetPinProtection_t, 365
- unpack_uim_UnblockPin_t, 368
- unpack_uim_VerifyPin_t, 369
- unpack_wds_SLQSCreateProfile_t, 375
- unpack_wds_SLQSGetProfileSettings_t, 379
- unpack_wds_SLQSSetIPFamilyPreference_t, 382
- TokenBucket
 - unpack_qos_swiQosFlow_t, 350
- tokenRate
 - unpack_qos_tokenBucket_t, 350
- tosMask
 - LibPackTFTIDParams, 64
- total_rx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_rx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_bytes_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 340
- total_tx_pkt_drp_drp_drp_d

- unpack_qmi_t, 332
- u16PRLVersion
 - unpack_dms_GetPRLVersion_t, 267
- u8PRLPreference
 - unpack_dms_GetPRLVersion_t, 267
- UDPDstPort
 - unpack_qos_swiQosFilter_t, 346
- UDPSrcPort
 - unpack_qos_swiQosFilter_t, 346
- ULONG
 - SwiDataTypes.h, 543
- ULONGLONG
 - SwiDataTypes.h, 543
- UMTSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, 381
- UMTSInstInfo
 - nas_UMTSInfo, 151
- UMTSLTENbrCell
 - nas_WCDMAInfoLTENeighborCell, 156
- UMTSReqQoS
 - LibPackUMTSReqQoSSigInd, 67
- UNIQUE_ID_LEN
 - dms.h, 410
- UNUSEDPARAM
 - common.h, 403
 - SwiDataTypes.h, 543
- uPhyCardStatus
 - slot_t, 234
- uPhySlotStatus
 - slot_t, 234
- USBComp
 - pack_dms_SetUSBComp_t, 174
 - unpack_dms_GetUSBComp_t, 268
- USHORT
 - SwiDataTypes.h, 543
- uarfcn
 - nas_lteWcdmaCellInfo, 124
 - nas_UMTSInfo, 151
- ueInIdle
 - nas_LTEInfoInterfreq, 113
 - nas_LTEInfoIntrafreq, 115
 - nas_LTEInfoNeighboringGSM, 116
 - nas_LTEInfoNeighboringWCDMA, 117
- uim.h, 554
 - MAX_ICCID_LENGTH, 556
 - MAX_NO_OF_SLOTS, 556
 - MAX_SLOTS_STATUS, 556
 - pack_uim_ChangePin, 556
 - pack_uim_GetCardStatus, 556
 - pack_uim_ReadTransparent, 557
 - pack_uim_SLQSUIEventRegister, 557
 - pack_uim_SLQSUIGetSlotsStatus, 558
 - pack_uim_SLQSUISwitchSlot, 558
 - pack_uim_SetPinProtection, 557
 - pack_uim_UnblockPin, 558
 - pack_uim_VerifyPin, 559
 - unpack_uim_ChangePin, 559
 - unpack_uim_GetCardStatus, 560
 - unpack_uim_ReadTransparent, 560
 - unpack_uim_SLQSUIEventRegister, 561
 - unpack_uim_SLQSUIGetSlotsStatus, 561
 - unpack_uim_SLQSUISetStatusChangeCall-
Back_ind, 562
 - unpack_uim_SLQSUISwitchSlot, 562
 - unpack_uim_SetPinProtection, 560
 - unpack_uim_SetUimSlotStatusChangeCallback_-
ind, 561
 - unpack_uim_UnblockPin, 562
 - unpack_uim_VerifyPin, 563
- uim_UIMSessionInformation, 255
 - aid, 256
 - aidLength, 256
 - sessionType, 256
- uim_appStatus, 242
 - aidLength, 245
 - aidVal, 245
 - appState, 245
 - appType, 245
 - persoFeature, 245
 - persoRetries, 245
 - persoState, 245
 - persoUnblockRetries, 245
 - pin1Retries, 245
 - pin1State, 245
 - pin2Retries, 245
 - pin2State, 245
 - puk1Retries, 245
 - puk2Retries, 245
 - univPin, 245
- uim_cardResult, 245
 - sw1, 246
 - sw2, 246
- uim_cardStatus, 246
 - index1xPri, 247
 - index1xSec, 247
 - indexGwPri, 247
 - indexGwSec, 247
 - numSlot, 247
 - SlotInfo, 247
- uim_changeUIMPIN, 247
 - oldPINLen, 248
 - oldPINVal, 248
 - pinID, 248
 - pinLen, 248
 - pinVal, 248
- uim_encryptedPIN1, 248
 - pin1Len, 249
 - pin1Val, 249
- uim_fileInfo, 249
 - fileID, 249
 - path, 249
 - pathLen, 249
- uim_hotSwapStatus, 249
 - hotSwap, 250
 - hotSwapLength, 250
- uim_readResult, 250

- content, 250
- contentLen, 250
- uim_readTransparentInfo, 251
 - length, 251
 - offset, 251
- uim_remainingRetries, 251
 - unblockLeft, 252
 - verifyLeft, 252
- uim_sessionInformation, 252
 - aid, 252
 - aidLength, 252
 - sessionType, 252
- uim_setPINProtection, 253
 - pinID, 253
 - pinLength, 253
 - pinOperation, 253
 - pinValue, 253
- uim_slotInfo, 253
 - AppStatus, 255
 - cardState, 255
 - errorState, 255
 - numApp, 255
 - upinRetries, 255
 - upinState, 255
 - upukRetries, 255
- uim_unblockUIMPIN, 256
 - newPINLen, 257
 - newPINVal, 257
 - pinID, 257
 - pukLen, 257
 - pukVal, 257
- uim_verifyUIMPIN, 257
 - pinID, 257
 - pinLen, 257
 - pinVal, 257
- uimSlotStatus
 - slots_t, 236
- ulMask
 - rmTrasnferStaticsReq, 233
- ulPhysicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, 215
- umtsEcio
 - nas_UMTSinstInfo, 152
- umtsInst
 - nas_UMTSInfo, 151
- umtsLTENbrCellLen
 - nas_WCDMAInfoLTENeighborCell, 156
- umtsPsc
 - nas_UMTSinstInfo, 152
- umtsRscp
 - nas_UMTSinstInfo, 152
- umtsUarfcn
 - nas_UMTSinstInfo, 152
- UnPackGetProfileSettingOut, 387
 - curProfile, 387
 - pExtErrCode, 387
- unblockLeft
 - uim_remainingRetries, 252
- uniqueID
 - image_info_t, 37
- univPin
 - appStats, 26
 - uim_appStatus, 245
- universalTime
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind-
_t, 322
- unpack_QosFlowStat_t, 351
 - bearerId, 352
 - tx_bytes, 352
 - tx_bytes_drp, 352
 - tx_pkt, 352
 - tx_pkt_drp, 352
- unpack_dms_GetActivationState
 - dms.h, 425
- unpack_dms_GetActivationState_t, 257
 - state, 258
- unpack_dms_GetBandCapability
 - dms.h, 425
- unpack_dms_GetBandCapability_t, 258
 - BandCapability, 258
 - Tlvresult, 258
- unpack_dms_GetCrashAction
 - dms.h, 425
- unpack_dms_GetCrashAction_t, 258
 - DevCrashState, 258
 - Tlvresult, 258
- unpack_dms_GetCustFeature
 - dms.h, 426
- unpack_dms_GetCustFeature_t, 259
 - DHCPRelayEnabled, 259
 - DisableIMSI, 259
 - GPSPMP, 259
 - GPSSel, 259
 - GpsEnable, 259
 - IPFamSupport, 259
 - IsVoiceEnabled, 259
 - RMAutoConnect, 259
 - SMSSupport, 259
 - Tlvresult, 259
- unpack_dms_GetCustFeaturesV2
 - dms.h, 426
- unpack_dms_GetCustFeaturesV2_t, 259
 - GetCustomFeatureV2, 260
 - Tlvresult, 260
- unpack_dms_GetDeviceCap
 - dms.h, 426
- unpack_dms_GetDeviceCap_t, 260
 - DataServiceCapability, 260
 - MaxRXChannelRate, 260
 - MaxTXChannelRate, 260
 - Radiolfaces, 260
 - RadiolfacesSize, 260
 - SimCapability, 260
 - Tlvresult, 260
- unpack_dms_GetDeviceCapabilities
 - dms.h, 427

- unpack_dms_GetDeviceCapabilities_t, 260
 - dataServiceCaCapability, 261
 - maxRxChannelRate, 261
 - maxTxChannelRate, 261
 - Radiofaces, 261
 - radiofacesSize, 261
 - simCapability, 261
- unpack_dms_GetDeviceHardwareRev
 - dms.h, 427
- unpack_dms_GetDeviceHardwareRev_t, 261
 - String, 261
 - stringSize, 261
 - Tlvresult, 261
- unpack_dms_GetDeviceMfr
 - dms.h, 427
- unpack_dms_GetDeviceMfr_t, 261
 - String, 261
 - stringSize, 261
 - Tlvresult, 261
- unpack_dms_GetDeviceSerialNumbers
 - dms.h, 428
- unpack_dms_GetDeviceSerialNumbers_t, 262
 - ESNString, 262
 - esnSize, 262
 - IMEIString, 262
 - imeiSize, 262
 - imeiSvnSize, 262
 - ImeiSvnString, 262
 - MEIDString, 262
 - meidSize, 262
 - Tlvresult, 262
- unpack_dms_GetFSN
 - dms.h, 429
- unpack_dms_GetFSN_t, 264
 - String, 264
 - Tlvresult, 264
- unpack_dms_GetFirmwareInfo
 - dms.h, 428
- unpack_dms_GetFirmwareInfo_t, 262
 - appversion_str, 263
 - bootversion_str, 263
 - carrier_str, 263
 - cur_carr_name, 263
 - cur_carr_rev, 263
 - modelid_str, 263
 - packageid_str, 263
 - priversion_str, 263
 - sku_str, 263
 - Tlvresult, 263
- unpack_dms_GetFirmwareRevision
 - dms.h, 428
- unpack_dms_GetFirmwareRevision_t, 263
 - AMSSString, 263
 - amssSize, 263
 - PRISString, 263
 - Tlvresult, 264
- unpack_dms_GetFirmwareRevisions
 - dms.h, 429
- unpack_dms_GetFirmwareRevisions_t, 264
 - AMSSString, 264
 - amssSize, 264
 - bootSize, 264
 - BootString, 264
 - PRISString, 264
 - priSize, 264
 - Tlvresult, 264
- unpack_dms_GetHardwareRevision
 - dms.h, 429
- unpack_dms_GetHardwareRevision_t, 265
 - hwVer, 265
- unpack_dms_GetIMSI
 - dms.h, 430
- unpack_dms_GetIMSI_t, 265
 - imsi, 265
 - Tlvresult, 265
- unpack_dms_GetModelID
 - dms.h, 430
- unpack_dms_GetModelID_t, 265
 - modelid, 265
 - Tlvresult, 265
- unpack_dms_GetNetworkTime
 - dms.h, 430
- unpack_dms_GetNetworkTime_t, 265
 - source, 266
 - timestamp, 266
 - Tlvresult, 266
- unpack_dms_GetPRLVersion
 - dms.h, 431
- unpack_dms_GetPRLVersion_t, 267
 - Tlvresult, 267
 - u16PRLVersion, 267
 - u8PRLPreference, 267
- unpack_dms_GetPower
 - dms.h, 431
- unpack_dms_GetPower_t, 266
 - HardwareControlledMode, 267
 - OfflineReason, 267
 - OperationMode, 267
 - Tlvresult, 267
- unpack_dms_GetSerialNumbers
 - dms.h, 431
- unpack_dms_GetSerialNumbers_t, 267
 - esn, 267
 - imei_no, 267
 - imeisv_svn, 267
 - meid, 267
- unpack_dms_GetUSBComp
 - dms.h, 432
- unpack_dms_GetUSBComp_t, 268
 - NumSupUSBComps, 268
 - SupUSBComps, 268
 - Tlvresult, 268
 - USBComp, 268
- unpack_dms_GetVoiceNumber
 - dms.h, 432
- unpack_dms_GetVoiceNumber_t, 268

- MIN, [268](#)
- minSize, [268](#)
- Tlvresult, [268](#)
- VoiceNumber, [268](#)
- voiceNumberSize, [268](#)
- unpack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [435](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind
 - dms.h, [435](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, [271](#)
 - source, [271](#)
 - Tlvresult, [271](#)
 - type, [272](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_t, [272](#)
 - source, [272](#)
 - Tlvresult, [272](#)
 - type, [272](#)
- unpack_dms_SLQSDmsSwiIndicationRegister
 - dms.h, [436](#)
- unpack_dms_SLQSDmsSwiIndicationRegister_t, [273](#)
 - Tlvresult, [273](#)
- unpack_dms_SLQSGetBandCapability
 - dms.h, [436](#)
- unpack_dms_SLQSGetBandCapability_t, [273](#)
 - bandCapability, [276](#)
 - LteBandCapability, [276](#)
 - TdsBandCapability, [276](#)
- unpack_dms_SLQSSwiClearDyingGaspStatistics
 - dms.h, [437](#)
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, [276](#)
 - Tlvresult, [276](#)
- unpack_dms_SLQSSwiGetDyingGaspCfg
 - dms.h, [437](#)
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, [276](#)
 - pGetDyingGaspCfg, [277](#)
 - Tlvresult, [277](#)
- unpack_dms_SLQSSwiGetDyingGaspStatistics
 - dms.h, [437](#)
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, [277](#)
 - pGetDyingGaspStatistics, [277](#)
 - Tlvresult, [277](#)
- unpack_dms_SLQSSwiGetFirmwareCurr
 - dms.h, [438](#)
- unpack_dms_SLQSSwiGetFirmwareCurr_t, [277](#)
 - carrier, [278](#)
 - fwvers, [278](#)
 - numEntries, [278](#)
 - pCurrImgInfo, [278](#)
 - pkgver, [278](#)
 - priver, [278](#)
- unpack_dms_SLQSSwiGetFwUpdateStatus
 - dms.h, [438](#)
- unpack_dms_SLQSSwiGetFwUpdateStatus_t, [278](#)
 - imgType, [279](#)
 - logString, [279](#)
 - refData, [279](#)
 - refString, [279](#)
 - ResCode, [279](#)
 - Tlvresult, [279](#)
- unpack_dms_SLQSSwiSetDyingGaspCfg
 - dms.h, [438](#)
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, [280](#)
 - Tlvresult, [280](#)
- unpack_dms_SetCrashAction
 - dms.h, [432](#)
- unpack_dms_SetCrashAction_t, [268](#)
 - notused, [269](#)
- unpack_dms_SetCustFeature
 - dms.h, [433](#)
- unpack_dms_SetCustFeature_t, [269](#)
 - Tlvresult, [269](#)
- unpack_dms_SetCustFeaturesV2
 - dms.h, [433](#)
- unpack_dms_SetCustFeaturesV2_t, [269](#)
 - Tlvresult, [269](#)
- unpack_dms_SetEventReport
 - dms.h, [433](#)
- unpack_dms_SetEventReport_ind
 - dms.h, [434](#)
- unpack_dms_SetEventReport_ind_t, [269](#)
 - ActivationStatusTlv, [270](#)
 - OperatingModeTlv, [270](#)
 - Tlvresult, [270](#)
- unpack_dms_SetEventReport_t, [270](#)
 - Tlvresult, [270](#)
- unpack_dms_SetFirmwarePreference
 - dms.h, [434](#)
- unpack_dms_SetFirmwarePreference_t, [270](#)
 - Tlvresult, [270](#)
- unpack_dms_SetPower
 - dms.h, [434](#)
- unpack_dms_SetPower_t, [270](#)
 - Tlvresult, [270](#)
- unpack_dms_SetUSBComp
 - dms.h, [435](#)
- unpack_dms_SetUSBComp_t, [270](#)
 - Tlvresult, [271](#)
- unpack_dms_UIMGetICCID
 - dms.h, [439](#)
- unpack_dms_UIMGetICCID_t, [280](#)
 - String, [280](#)
 - stringSize, [280](#)
 - Tlvresult, [280](#)
- unpack_fms_GetImagesPreference
 - fms.h, [441](#)
- unpack_fms_GetImagesPreference_t, [281](#)
 - ImageListSize, [281](#)
 - plmageList, [281](#)
 - Tlvresult, [281](#)
- unpack_fms_GetStoredImages
 - fms.h, [442](#)
- unpack_fms_GetStoredImages_t, [281](#)
 - imageList, [282](#)
 - imagelistSize, [282](#)
 - Tlvresult, [282](#)

- unpack_fms_SetImagesPreference
 - fms.h, [442](#)
- unpack_fms_SetImagesPreference_t, [282](#)
 - ImageTypes, [282](#)
 - ImageTypesSize, [282](#)
 - Tlvresult, [282](#)
- unpack_loc_BestAvailPos_Ind
 - loc.h, [449](#)
- unpack_loc_BestAvailPos_Ind_t, [282](#)
 - pAltitudeWrtEllipsoid, [287](#)
 - pAltitudeWrtMeanSeaLevel, [287](#)
 - pGpsTime, [287](#)
 - pHeading, [287](#)
 - pHeadingUnc, [287](#)
 - pHorCirConf, [287](#)
 - pHorEllpConf, [287](#)
 - pHorReliability, [287](#)
 - pHorUncCircular, [287](#)
 - pHorUncEllipseOrientAzimuth, [287](#)
 - pHorUncEllipseSemiMajor, [287](#)
 - pHorUncEllipseSemiMinor, [287](#)
 - pLatitude, [288](#)
 - pLongitude, [288](#)
 - pMagneticDeviation, [288](#)
 - pPrecisionDilution, [288](#)
 - pSensorDataUsage, [288](#)
 - pSpeedHorizontal, [288](#)
 - pSpeedUnc, [288](#)
 - pSpeedVertical, [288](#)
 - pSpeedVerticalUnc, [288](#)
 - pSvUsedforFix, [288](#)
 - pTechnologyMask, [288](#)
 - pTimeSrc, [288](#)
 - pTimeUnc, [288](#)
 - pTimestampUtc, [288](#)
 - pVertConfidence, [288](#)
 - pVertReliability, [288](#)
 - pVertUnc, [288](#)
 - pXid, [288](#)
 - status, [288](#)
 - Tlvresult, [288](#)
- unpack_loc_Delete_Assist_Data_t, [288](#)
 - Tlvresult, [289](#)
- unpack_loc_DeleteAssistData
 - loc.h, [449](#)
- unpack_loc_EngineState_Ind
 - loc.h, [450](#)
- unpack_loc_EngineState_Ind_t, [289](#)
 - engineState, [289](#)
 - Tlvresult, [289](#)
- unpack_loc_EventRegister
 - loc.h, [450](#)
- unpack_loc_EventRegister_t, [289](#)
 - Tlvresult, [290](#)
- unpack_loc_PositionRpt_Ind
 - loc.h, [450](#)
- unpack_loc_PositionRpt_Ind_t, [290](#)
 - pAltitudeAssumed, [294](#)
 - pAltitudeWrtEllipsoid, [294](#)
 - pAltitudeWrtMeanSeaLevel, [294](#)
 - pFixId, [294](#)
 - pGpsTime, [294](#)
 - pHeading, [294](#)
 - pHeadingUnc, [295](#)
 - pHorConfidence, [295](#)
 - pHorReliability, [295](#)
 - pHorUncCircular, [295](#)
 - pHorUncEllipseOrientAzimuth, [295](#)
 - pHorUncEllipseSemiMajor, [295](#)
 - pHorUncEllipseSemiMinor, [295](#)
 - pLatitude, [295](#)
 - pLeapSeconds, [295](#)
 - pLongitude, [295](#)
 - pMagneticDeviation, [295](#)
 - pPrecisionDilution, [295](#)
 - pSensorDataUsage, [295](#)
 - pSpeedHorizontal, [295](#)
 - pSpeedUnc, [295](#)
 - pSpeedVertical, [295](#)
 - pSvUsedforFix, [295](#)
 - pTechnologyMask, [295](#)
 - pTimeSrc, [295](#)
 - pTimeUnc, [295](#)
 - pTimestampUtc, [295](#)
 - pVertConfidence, [295](#)
 - pVertReliability, [295](#)
 - pVertUnc, [295](#)
 - sessionId, [295](#)
 - sessionStatus, [295](#)
 - Tlvresult, [295](#)
- unpack_loc_SLQSLOCGetBestAvailPos
 - loc.h, [452](#)
- unpack_loc_SLQSLOCGetBestAvailPos_t, [297](#)
 - Tlvresult, [297](#)
- unpack_loc_SetExtPowerConfig_Ind
 - loc.h, [451](#)
- unpack_loc_SetExtPowerConfig_Ind_t, [296](#)
 - status, [296](#)
 - Tlvresult, [296](#)
- unpack_loc_SetExtPowerState
 - loc.h, [451](#)
- unpack_loc_SetExtPowerState_t, [296](#)
 - Tlvresult, [297](#)
- unpack_loc_SetOperationMode
 - loc.h, [451](#)
- unpack_loc_SetOperationMode_t, [297](#)
 - Tlvresult, [297](#)
- unpack_loc_Start
 - loc.h, [452](#)
- unpack_loc_Start_t, [297](#)
 - Tlvresult, [298](#)
- unpack_loc_Stop
 - loc.h, [452](#)
- unpack_loc_Stop_t, [298](#)
 - Tlvresult, [298](#)
- unpack_nas_GetACCOLC

- nas.h, [469](#)
- unpack_nas_GetANAAAAAuthenticationStatus
 - nas.h, [469](#)
- unpack_nas_GetCDMANetworkParameters
 - nas.h, [470](#)
- unpack_nas_GetCDMANetworkParameters_t, [298](#)
 - Application, [299](#)
 - Broadcast, [299](#)
 - CustomSCP, [299](#)
 - ForceRev0, [299](#)
 - Protocol, [299](#)
 - RegForeignNID, [299](#)
 - RegForeignSID, [299](#)
 - RegHomeSID, [299](#)
 - Roaming, [299](#)
 - SCI, [299](#)
 - SCM, [299](#)
- unpack_nas_GetHomeNetwork
 - nas.h, [470](#)
- unpack_nas_GetHomeNetwork_t, [299](#)
 - mcc, [300](#)
 - mnc, [300](#)
 - name, [300](#)
 - nid, [300](#)
 - sid, [300](#)
- unpack_nas_GetNetworkPreference
 - nas.h, [470](#)
- unpack_nas_GetNetworkPreference_t, [300](#)
 - ActiveTechPref, [301](#)
 - Duration, [301](#)
 - PersistentTechPref, [301](#)
 - Tlvresult, [301](#)
- unpack_nas_GetRFInfo
 - nas.h, [470](#)
- unpack_nas_GetRFInfo_t, [301](#)
 - instancesSize, [301](#)
 - RFBandInfoElements, [301](#)
- unpack_nas_GetServingNetwork
 - nas.h, [471](#)
- unpack_nas_GetServingNetwork_t, [301](#)
 - CSDomain, [302](#)
 - DataCaps, [302](#)
 - DataCapsLen, [302](#)
 - MCC, [302](#)
 - MNC, [302](#)
 - Name, [302](#)
 - nameSize, [302](#)
 - PSDomain, [302](#)
 - RAN, [302](#)
 - Radiofaces, [302](#)
 - RadiofacesSize, [302](#)
 - RegistrationState, [302](#)
 - Roaming, [302](#)
- unpack_nas_GetServingNetworkCapabilities
 - nas.h, [471](#)
- unpack_nas_GetServingNetworkCapabilities_t, [302](#)
 - DataCaps, [303](#)
 - DataCapsLen, [303](#)
- unpack_nas_GetSignalStrengths
 - nas.h, [471](#)
- unpack_nas_GetSignalStrengths_t, [303](#)
 - len, [303](#)
 - radio, [303](#)
 - rsi, [303](#)
- unpack_nas_PerformNetworkScan
 - nas.h, [472](#)
- unpack_nas_PerformNetworkScan_t, [303](#)
 - p3GppNetworkInfoInstances, [304](#)
 - p3GppNetworkInstanceSize, [304](#)
 - pPCSInstance, [304](#)
 - pPCSInstanceSize, [304](#)
 - pRATInstance, [304](#)
 - pRATInstanceSize, [304](#)
 - pScanResult, [304](#)
- unpack_nas_SLQSGetNetworkTime
 - nas.h, [474](#)
- unpack_nas_SLQSGetNetworkTime_t, [308](#)
 - p3GPP2TimeInfo, [308](#)
 - p3GPPTIMEInfo, [308](#)
- unpack_nas_SLQSGetPLMNName
 - nas.h, [474](#)
- unpack_nas_SLQSGetPLMNName_t, [308](#)
 - longName, [309](#)
 - longNameCI, [309](#)
 - longNameEn, [309](#)
 - longNameLen, [309](#)
 - longNameSB, [309](#)
 - shortName, [309](#)
 - shortNameCI, [309](#)
 - shortNameEn, [309](#)
 - shortNameLen, [309](#)
 - shortNameSB, [309](#)
 - spn, [309](#)
 - spnEncoding, [309](#)
 - spnLength, [309](#)
- unpack_nas_SLQSGetServingSystem
 - nas.h, [475](#)
- unpack_nas_SLQSGetServingSystem_t, [309](#)
 - BasestationID, [310](#)
 - BasestationLatitude, [311](#)
 - BasestationLongitude, [311](#)
 - CDMASystemInfoExt, [311](#)
 - CallBarStatus, [311](#)
 - CellID, [311](#)
 - ConcSvcInfo, [311](#)
 - CurrentPLMN, [311](#)
 - DTMInd, [311](#)
 - DataSrvCapabilities, [311](#)
 - DefaultRoamInd, [311](#)
 - DetailedSvcInfo, [311](#)
 - Gpp2TimeZone, [311](#)
 - GppNetworkDSTAdjustment, [311](#)
 - GppTimeZone, [311](#)
 - HdrPersonality, [311](#)
 - Lac, [311](#)
 - NetworkID, [311](#)

- PRLInd, [311](#)
- RoamIndicatorVal, [311](#)
- RoamingIndicatorList, [311](#)
- ServingSystem, [311](#)
- SystemID, [311](#)
- TrackAreaCode, [311](#)
- unpack_nas_SLQSGetSignalStrength
 - nas.h, [475](#)
- unpack_nas_SLQSGetSignalStrength_t, [311](#)
 - ecioList, [312](#)
 - ecioListLen, [312](#)
 - errorRateList, [312](#)
 - errorRateListLen, [312](#)
 - lo, [312](#)
 - ltsr, [312](#)
 - ltsnr, [312](#)
 - rsrqInfo, [312](#)
 - rxSignalStrengthList, [312](#)
 - rxSignalStrengthListLen, [312](#)
 - signalStrengthReqMask, [312](#)
 - sinr, [312](#)
- unpack_nas_SLQSGetSysInfo
 - nas.h, [475](#)
- unpack_nas_SLQSGetSysInfo_t, [313](#)
 - pAddCDMASysInfo, [315](#)
 - pAddGSMSysInfo, [315](#)
 - pAddHDRSysInfo, [315](#)
 - pAddLTESysInfo, [315](#)
 - pCDMASrvStatusInfo, [315](#)
 - pCDMASysInfo, [315](#)
 - pGSMCallBarringSysInfo, [315](#)
 - pGSMCipherDomainSysInfo, [315](#)
 - pGSMStatusInfo, [315](#)
 - pGSMStatusInfo, [315](#)
 - pHRSrvStatusInfo, [315](#)
 - pHRSysInfo, [315](#)
 - pLTESrvStatusInfo, [315](#)
 - pLTESysInfo, [315](#)
 - pLTEVoiceSupportSysInfo, [315](#)
 - pWCDMASysInfo, [315](#)
- unpack_nas_SLQSGetSysSelectionPref
 - nas.h, [476](#)
- unpack_nas_SLQSGetSysSelectionPref_t, [315](#)
 - pBandPref, [319](#)
 - pEmerMode, [319](#)
 - pGWAqOrderPref, [319](#)
 - pLTEBandPref, [319](#)
 - pModePref, [319](#)
 - pNetSelPref, [319](#)
 - pPRLPref, [319](#)
 - pRoamPref, [319](#)
 - pSrvDomainPref, [319](#)
- unpack_nas_SLQSIInitiateNetworkRegistration
 - nas.h, [476](#)
- unpack_nas_SLQSNasConfigSigInfo2
 - nas.h, [476](#)
- unpack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [476](#)
- unpack_nas_SLQSNasGetCellLocationInfo_t, [319](#)
 - pCDMAInfo, [320](#)
 - pLTEInfoInterfreq, [320](#)
 - pLTEInfoIntrafreq, [320](#)
 - pUMTSInfo, [321](#)
- unpack_nas_SLQSNasGetSigInfo
 - nas.h, [477](#)
- unpack_nas_SLQSNasGetSigInfo_t, [321](#)
 - CDMAInfo, [321](#)
 - GSMInfo, [321](#)
 - HDRInfo, [321](#)
 - LTESInfo, [321](#)
- unpack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [477](#)
- unpack_nas_SLQSNasNetworkTimeCallBack_ind
 - nas.h, [477](#)
- unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, [321](#)
 - pDayltSavAdj, [322](#)
 - pRadioInterface, [322](#)
 - pTimeZone, [322](#)
 - universalTime, [322](#)
- unpack_nas_SLQSNasSigInfoCallback_ind
 - nas.h, [478](#)
- unpack_nas_SLQSNasSigInfoCallback_ind_t, [322](#)
 - pRscp, [323](#)
- unpack_nas_SLQSNasSwiModemStatus
 - nas.h, [478](#)
- unpack_nas_SLQSNasSwiModemStatus_t, [323](#)
 - commonInfo, [323](#)
 - pLTEInfo, [323](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback
 - nas.h, [478](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind
 - nas.h, [479](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, [324](#)
 - Info, [324](#)
 - Tlvresult, [324](#)
- unpack_nas_SLQSNasSysInfoCallback_ind
 - nas.h, [479](#)
- unpack_nas_SLQSSetBandPreference
 - nas.h, [479](#)
- unpack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [479](#)
- unpack_nas_SLQSSetSysSelectionPref
 - nas.h, [480](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
 - nas.h, [480](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, [324](#)
 - Info, [324](#)
 - Tlvresult, [324](#)
- unpack_nas_SLQSSwiGetLteCQI
 - nas.h, [480](#)
- unpack_nas_SLQSSwiGetLteCQI_t, [324](#)
 - ValidityCW0, [325](#)
 - ValidityCW1, [325](#)

- unpack_nas_SLQSSysInfoCallback_ind_t, 325
 - pGSMSysInfo, 328
 - pHDRSysInfo, 328
 - pLTERSysInfo, 328
 - pSysInfoNoChange, 328
- unpack_nas_SetACCOLC
 - nas.h, 472
- unpack_nas_SetDataCapabilitiesCallback_ind
 - nas.h, 472
- unpack_nas_SetDataCapabilitiesCallback_ind_t, 304
 - dataCaps, 304
 - dataCapsSize, 304
- unpack_nas_SetEventReportInd
 - nas.h, 473
- unpack_nas_SetEventReportInd_t, 304
 - RFTlv, 305
 - RRTlv, 305
 - SLQSSSTlv, 305
 - SSTlv, 305
- unpack_nas_SetLURejectCallback
 - nas.h, 473
- unpack_nas_SetNasLTECphyCalIndCallback_ind
 - nas.h, 473
- unpack_nas_SetNasLTECphyCalIndCallback_ind_t, 305
- unpack_nas_SetNetworkPreference
 - nas.h, 473
- unpack_nas_SetNetworkPreference_t, 306
 - Tlvresult, 306
- unpack_nas_SetRFInfoCallback
 - nas.h, 473
- unpack_nas_SetRoamingIndicatorCallback_ind
 - nas.h, 474
- unpack_nas_SetRoamingIndicatorCallback_ind_t, 307
 - roaming, 307
- unpack_nas_SetServingSystemCallback_ind
 - nas.h, 474
- unpack_nas_SetServingSystemCallback_ind_t, 307
 - SSInfo, 307
 - Tlvresult, 307
- unpack_nas_SlqsGetLTECphyCAInfo
 - nas.h, 474
- unpack_nas_SlqsGetLTECphyCAInfo_t, 307
 - LTECphyCAInfo, 308
 - Tlvresult, 308
- unpack_omaDmConfigTlv_t, 328
 - alertmsg, 329
 - alertmsglength, 329
 - state, 329
 - userInputReq, 329
 - userInputTimeout, 329
- unpack_omaDmFotaTlv_t, 329
 - description, 331
 - descriptionlength, 331
 - fwloadsize, 331
 - fwloadComplete, 331
 - namelength, 331
 - package_name, 331
 - sessionType, 331
 - severity, 331
 - state, 331
 - updateCompleteStatus, 331
 - userInputReq, 331
 - userInputTimeout, 331
 - version, 331
 - versionlength, 331
- unpack_omaDmNotificationsTlv_t, 331
 - notification, 331
 - sessionStatus, 331
- unpack_qmi_t, 331
 - msgid, 332
 - type, 332
 - xid, 332
- unpack_qos_IPv4Addr_t, 332
 - addr, 333
 - subnetMask, 333
- unpack_qos_IPv6Addr_t, 333
 - addr, 333
 - prefixLen, 333
- unpack_qos_IPv6TrafCls_t, 333
 - mask, 334
 - val, 334
- unpack_qos_Port_t, 334
 - port, 335
 - range, 335
- unpack_qos_QosFlowInfo_t, 335
 - BearerID, 336
 - is_RxQFlowGranted_Available, 336
 - is_TxQFlowGranted_Available, 336
 - NumRxFilters, 336
 - NumTxFilters, 336
 - QFlowState, 336
 - RxQFilter, 336
 - RxQFlowGranted, 336
 - TxQFilter, 336
 - TxQFlowGranted, 336
- unpack_qos_QosFlowInfoState_t, 336
 - id, 337
 - isNewFlow, 337
 - state, 337
- unpack_qos_SLQSQosGetNetworkStatus
 - qos.h, 529
- unpack_qos_SLQSQosGetNetworkStatus_t, 337
 - NWQoSStatus, 337
- unpack_qos_SLQSQosSwiReadApnExtraParams
 - qos.h, 530
- unpack_qos_SLQSQosSwiReadApnExtraParams_t, 337
 - ambr_dl, 338
 - ambr_dl_ext, 338
 - ambr_dl_ext2, 338
 - ambr_ul, 338
 - ambr_ul_ext, 338
 - ambr_ul_ext2, 338
 - apnId, 338
- unpack_qos_SLQSQosSwiReadDataStats
 - qos.h, 530

- unpack_qos_SLQSQosSwiReadDataStats_t, 339
 - apnId, 340
 - numQosFlow, 340
 - qosFlow, 340
 - total_rx_bytes, 340
 - total_rx_pkt, 340
 - total_tx_bytes, 340
 - total_tx_pkt, 340
- unpack_qos_SLQSSetQosEventCallback
 - qos.h, 531
- unpack_qos_SLQSSetQosEventCallback_ind
 - qos.h, 531
- unpack_qos_SLQSSetQosEventCallback_ind_t, 340
 - NumFlows, 340
 - QosFlowInfo, 340
- unpack_qos_SLQSSetQosNWStatusCallback_ind
 - qos.h, 531
- unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 340
 - status, 341
- unpack_qos_SLQSSetQosPriEventCallback_ind
 - qos.h, 532
- unpack_qos_SLQSSetQosPriEventCallback_ind_t, 341
 - event, 341
- unpack_qos_SLQSSetQosStatusCallback_ind
 - qos.h, 533
- unpack_qos_SLQSSetQosStatusCallback_ind_t, 341
 - event, 342
 - id, 342
 - reason, 342
 - status, 342
- unpack_qos_Tos_t, 351
 - mask, 351
 - val, 351
- unpack_qos_dataRate_t, 332
 - dataRateMax, 332
 - guaranteedRate, 332
- unpack_qos_pktErrRate_t, 334
 - exponent, 334
 - multiplier, 334
- unpack_qos_swiQosFilter_t, 343
 - EspSpi, 345
 - IPv4DstAddr, 345
 - IPv4SrcAddr, 345
 - IPv4Tos, 345
 - IPv6DstAddr, 345
 - IPv6Label, 345
 - IPv6SrcAddr, 345
 - IPv6TrafCls, 345
 - Id, 345
 - index, 345
 - is_EspSpi_Available, 345
 - is_IPv4DstAddr_Available, 345
 - is_IPv4SrcAddr_Available, 345
 - is_IPv4Tos_Available, 345
 - is_IPv6DstAddr_Available, 345
 - is_IPv6Label_Available, 345
 - is_IPv6SrcAddr_Available, 345
 - is_IPv6TrafCls_Available, 345
 - is_Id_Available, 345
 - is_NxtHdrProto_Available, 345
 - is_Precedence_Available, 345
 - is_TCPDstPort_Available, 345
 - is_TCPSrcPort_Available, 345
 - is_TransDstPort_Available, 346
 - is_TransSrcPort_Available, 346
 - is_UDPdstPort_Available, 346
 - is_UDPSrcPort_Available, 346
 - NxtHdrProto, 346
 - Precedence, 346
 - TCPDstPort, 346
 - TCPSrcPort, 346
 - TransDstPort, 346
 - TransSrcPort, 346
 - UDPDstPort, 346
 - UDPSrcPort, 346
 - version, 346
- unpack_qos_swiQosFlow_t, 346
 - DataRate, 349
 - index, 349
 - is_DataRate_Available, 349
 - is_Jitter_Available, 349
 - is_Latency_Available, 349
 - is_LteQci_Available, 349
 - is_MaxAllowedPktSz_Available, 349
 - is_MinPolicedPktSz_Available, 349
 - is_PktErrRate_Available, 349
 - is_ProfileId3GPP2_Available, 349
 - is-TokenBucket_Available, 349
 - is_TrafficClass_Available, 349
 - is_val_3GPP2Pri_Available, 349
 - is_val_3GPPImCn_Available, 349
 - is_val_3GPPSigInd_Available, 349
 - Jitter, 349
 - Latency, 349
 - LteQci, 350
 - MaxAllowedPktSz, 350
 - MinPolicedPktSz, 350
 - PktErrRate, 350
 - ProfileId3GPP2, 350
 - TokenBucket, 350
 - TrafficClass, 350
 - val_3GPP2Pri, 350
 - val_3GPPImCn, 350
 - val_3GPPResResidualBER, 350
 - val_3GPPSigInd, 350
 - val_3GPPTraHdIPri, 350
- unpack_qos_tokenBucket_t, 350
 - bucketSz, 350
 - peakRate, 350
 - tokenRate, 350
- unpack_result_code_only
 - common.h, 405
- unpack_sms_SLQSDDeleteSMS
 - sms.h, 540
- unpack_sms_SLQSDDeleteSMS_t, 354

- unpack_sms_SLQSGetSMS
 - sms.h, [541](#)
- unpack_sms_SLQSGetSMS_t, [354](#)
 - message, [354](#)
 - messageFormat, [354](#)
 - messageSize, [354](#)
 - messageTag, [354](#)
- unpack_sms_SLQSGetSMSList
 - sms.h, [541](#)
- unpack_sms_SLQSGetSMSList_t, [354](#)
 - messageList, [355](#)
 - messageListSize, [355](#)
- unpack_sms_SLQSModifySMSStatus
 - sms.h, [541](#)
- unpack_sms_SLQSModifySMSStatus_t, [355](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind
 - sms.h, [542](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, [355](#)
 - messageMode, [355](#)
 - storageType, [355](#)
- unpack_sms_SendSMS
 - sms.h, [539](#)
- unpack_sms_SendSMS_t, [352](#)
 - messageFailureCode, [352](#)
 - messageID, [352](#)
- unpack_sms_SetNewSMSCallback
 - sms.h, [540](#)
- unpack_sms_SetNewSMSCallback_ind
 - sms.h, [540](#)
- unpack_sms_SetNewSMSCallback_ind_t, [352](#)
 - ETWSTlv, [353](#)
 - IMSTlv, [353](#)
 - MMTlv, [353](#)
 - NewMMTlv, [353](#)
 - SMSCTlv, [353](#)
 - TRMessageTlv, [353](#)
- unpack_sms_SetNewSMSCallback_t, [353](#)
- unpack_swiloc_SwiLocGetAutoStart
 - swiloc.h, [544](#)
- unpack_swiloc_SwiLocGetAutoStart_t, [356](#)
 - fix_rate, [357](#)
 - fix_rate_reported, [357](#)
 - fix_type, [357](#)
 - fix_type_reported, [357](#)
 - function, [357](#)
 - function_reported, [357](#)
 - max_dist, [357](#)
 - max_dist_reported, [357](#)
 - max_time, [357](#)
 - max_time_reported, [357](#)
- unpack_swiloc_SwiLocSetAutoStart
 - swiloc.h, [545](#)
- unpack_swioima_SLQSOMADMAAlertCallback
 - swioima.h, [550](#)
- unpack_swioima_SLQSOMADMAAlertCallback_ind
 - swioima.h, [551](#)
- unpack_swioima_SLQSOMADMAAlertCallback_ind_t, [357](#)
 - eventType, [358](#)
- unpack_swioima_SLQSOMADMCancelSession
 - swioima.h, [551](#)
- unpack_swioima_SLQSOMADMGetSessionInfo
 - swioima.h, [552](#)
- unpack_swioima_SLQSOMADMGetSessionInfo_t, [358](#)
 - Date, [360](#)
 - DateLength, [360](#)
 - PkgDescLength, [360](#)
 - PkgDescription, [360](#)
 - PkgName, [360](#)
 - PkgNameLength, [360](#)
 - RetryCount, [360](#)
 - SessionState, [360](#)
 - SessionType, [360](#)
 - Severity, [360](#)
 - Source, [360](#)
 - SourceLength, [360](#)
 - Status, [360](#)
 - Time, [361](#)
 - TimeLength, [361](#)
 - UpdateCompleteStatus, [361](#)
- unpack_swioima_SLQSOMADMGetSettings
 - swioima.h, [552](#)
- unpack_swioima_SLQSOMADMGetSettings_t, [361](#)
 - Autosdm, [362](#)
 - FOTAdownload, [362](#)
 - FwAutoCheck, [362](#)
- unpack_swioima_SLQSOMADMSelect
 - swioima.h, [553](#)
- unpack_swioima_SLQSOMADMSetSettings
 - swioima.h, [553](#)
- unpack_swioima_SLQSOMADMStartSession
 - swioima.h, [553](#)
- unpack_swioima_SLQSOMADMStartSession_t, [362](#)
 - FwAvailability, [362](#)
- unpack_uim_ChangePin
 - uim.h, [559](#)
- unpack_uim_ChangePin_t, [363](#)
 - pEncryptedPIN1, [363](#)
 - pIndicationToken, [363](#)
 - pRemainingRetries, [363](#)
 - Tlvresult, [363](#)
- unpack_uim_GetCardStatus
 - uim.h, [560](#)
- unpack_uim_GetCardStatus_t, [363](#)
 - pCardStatus, [364](#)
 - pHotSwapStatus, [364](#)
 - Tlvresult, [364](#)
- unpack_uim_ReadTransparent
 - uim.h, [560](#)
- unpack_uim_ReadTransparent_t, [364](#)
 - pCardResult, [365](#)
 - pEncryptedData, [365](#)
 - pIndicationToken, [365](#)
 - pReadResult, [365](#)

- Tlvresult, [365](#)
- unpack_uim_SLQSUIEventRegister
 - uim.h, [561](#)
- unpack_uim_SLQSUIEventRegister_t, [366](#)
 - eventMask, [366](#)
- unpack_uim_SLQSUIGetSlotsStatus
 - uim.h, [561](#)
- unpack_uim_SLQSUIGetSlotsStatus_t, [366](#)
 - pNumberOfPhySlot, [367](#)
 - pUimSlotsStatus, [367](#)
- unpack_uim_SLQSUISetStatusChangeCallBack_ind
 - uim.h, [562](#)
- unpack_uim_SLQSUISetStatusChangeCallBack_ind_t, [367](#)
- unpack_uim_SLQSUISwitchSlot
 - uim.h, [562](#)
- unpack_uim_SetPinProtection
 - uim.h, [560](#)
- unpack_uim_SetPinProtection_t, [365](#)
 - pEncryptedPIN1, [365](#)
 - pIndicationToken, [365](#)
 - pRemainingRetries, [365](#)
 - Tlvresult, [365](#)
- unpack_uim_SetUimSlotStatusChangeCallback_ind
 - uim.h, [561](#)
- unpack_uim_SetUimSlotStatusChangeCallback_ind_t, [366](#)
 - bNumberOfPhySlots, [366](#)
 - slotsstatusChange, [366](#)
- unpack_uim_UnblockPin
 - uim.h, [562](#)
- unpack_uim_UnblockPin_t, [367](#)
 - pEncryptedPIN1, [368](#)
 - pIndicationToken, [368](#)
 - pRemainingRetries, [368](#)
 - Tlvresult, [368](#)
- unpack_uim_VerifyPin
 - uim.h, [563](#)
- unpack_uim_VerifyPin_t, [368](#)
 - pEncryptedPIN1, [369](#)
 - pIndicationToken, [369](#)
 - pRemainingRetries, [369](#)
 - Tlvresult, [369](#)
- unpack_wds_GetConnectionRate
 - wds.h, [580](#)
- unpack_wds_GetConnectionRate_t, [369](#)
 - currentChannelRXRate, [369](#)
 - currentChannelTXRate, [369](#)
 - maxChannelRXRate, [369](#)
 - maxChannelTXRate, [369](#)
- unpack_wds_GetDefaultProfile
 - wds.h, [580](#)
- unpack_wds_GetDefaultProfile_t, [369](#)
 - apnname, [370](#)
 - apnsize, [370](#)
 - auth, [370](#)
 - ipaddr, [370](#)
 - ipaddrv6, [370](#)
 - name, [370](#)
 - namesize, [370](#)
 - pdptype, [370](#)
 - pridns, [370](#)
 - pridnsv6, [370](#)
 - secdns, [370](#)
 - secdnsv6, [370](#)
 - username, [370](#)
 - usersize, [371](#)
- unpack_wds_GetDefaultProfileNum
 - wds.h, [580](#)
- unpack_wds_GetDefaultProfileNum_t, [371](#)
 - index, [371](#)
- unpack_wds_GetDormancyState
 - wds.h, [581](#)
- unpack_wds_GetDormancyState_t, [371](#)
 - dormancyState, [371](#)
- unpack_wds_GetLastMobileIPError
 - wds.h, [581](#)
- unpack_wds_GetLastMobileIPError_t, [371](#)
 - error, [371](#)
- unpack_wds_GetMobileIP
 - wds.h, [581](#)
- unpack_wds_GetMobileIP_t, [372](#)
 - mipMode, [372](#)
- unpack_wds_GetMobileIPProfile
 - wds.h, [582](#)
- unpack_wds_GetMobileIPProfile_t, [372](#)
 - AAASPI, [372](#)
 - AAASState, [372](#)
 - address, [373](#)
 - enabled, [373](#)
 - HASPI, [373](#)
 - HASState, [373](#)
 - NAI, [373](#)
 - naiSize, [373](#)
 - primaryHA, [373](#)
 - revTunneling, [373](#)
 - secondaryHA, [373](#)
- unpack_wds_GetPacketStatus
 - wds.h, [582](#)
- unpack_wds_GetPacketStatus_t, [373](#)
 - rXDroppedCount, [374](#)
 - rXOKBytesLastCall, [374](#)
 - rXOkBytesCount, [374](#)
 - rXPacketErrors, [374](#)
 - rXPacketOverflows, [374](#)
 - rXPacketSuccesses, [374](#)
 - tXDroppedCount, [374](#)
 - tXOKBytesLastCall, [374](#)
 - tXOkBytesCount, [374](#)
 - tXPacketErrors, [374](#)
 - tXPacketOverflows, [374](#)
 - tXPacketSuccesses, [374](#)
- unpack_wds_GetSessionDuration
 - wds.h, [582](#)
- unpack_wds_GetSessionDuration_t, [374](#)
 - callDuration, [374](#)

- unpack_wds_GetSessionState
 - wds.h, [583](#)
- unpack_wds_GetSessionState_t, [374](#)
 - connectionStatus, [375](#)
- unpack_wds_RMSetTransferStatistics
 - wds.h, [583](#)
- unpack_wds_RMSetTransferStatistics_t, [375](#)
- unpack_wds_SLQSCreateProfile
 - wds.h, [584](#)
- unpack_wds_SLQSCreateProfile_t, [375](#)
 - pCreateProfileOut, [375](#)
 - pProfileID, [375](#)
 - Tlvresult, [375](#)
- unpack_wds_SLQSDeleteProfile
 - wds.h, [585](#)
- unpack_wds_SLQSDeleteProfile_t, [375](#)
 - extendedErrorCode, [376](#)
- unpack_wds_SLQSGet3GPPConfigItem
 - wds.h, [585](#)
- unpack_wds_SLQSGet3GPPConfigItem_t, [376](#)
 - _3gppRelease, [376](#)
 - defaultPDNEnabled, [376](#)
 - profileList, [377](#)
- unpack_wds_SLQSGetCurrDataSystemStat
 - wds.h, [585](#)
- unpack_wds_SLQSGetCurrDataSystemStat_t, [377](#)
 - currNetworkInfo, [377](#)
 - networkInfoLen, [377](#)
 - prefNetwork, [377](#)
- unpack_wds_SLQSGetDUNCallInfo
 - wds.h, [586](#)
- unpack_wds_SLQSGetDUNCallInfo_t, [378](#)
 - callEndReason, [378](#)
 - channelRate, [378](#)
 - connectionStatus, [378](#)
 - dataBearerTech, [378](#)
 - dormancyStatus, [378](#)
 - lastCallDataBearerTech, [378](#)
 - mdmCallDurationActive, [379](#)
 - rxOKBytesCount, [379](#)
 - txOKBytesCount, [379](#)
- unpack_wds_SLQSGetDataBearerTechnology
 - wds.h, [586](#)
- unpack_wds_SLQSGetDataBearerTechnology_t, [377](#)
 - curDataBearerTechnology, [377](#)
 - dataBearerMask, [377](#)
 - lastCallDataBearerTechnology, [378](#)
- unpack_wds_SLQSGetProfileSettings
 - wds.h, [587](#)
- unpack_wds_SLQSGetProfileSettings_t, [379](#)
 - pProfileSettings, [379](#)
 - ProfileType, [379](#)
 - Tlvresult, [379](#)
- unpack_wds_SLQSGetRuntimeSettings
 - wds.h, [587](#)
- unpack_wds_SLQSGetRuntimeSettings_t, [379](#)
 - APNName, [380](#)
 - Authentication, [380](#)
 - DomainList, [380](#)
 - GPRSGrantedQoS, [380](#)
 - GWAddressV4, [380](#)
 - IMCNflag, [380](#)
 - IPFamilyPreference, [380](#)
 - IPv6AddrInfo, [381](#)
 - IPv6GWAddrInfo, [381](#)
 - IPv4, [380](#)
 - Mtu, [381](#)
 - PDPTType, [381](#)
 - PrimaryDNSV4, [381](#)
 - PrimaryDNSV6, [381](#)
 - ProfileID, [381](#)
 - ProfileName, [381](#)
 - SecondaryDNSV4, [381](#)
 - SecondaryDNSV6, [381](#)
 - ServerAddrList, [381](#)
 - SubnetMaskV4, [381](#)
 - Technology, [381](#)
 - UMTSGrantedQoS, [381](#)
 - Username, [381](#)
- unpack_wds_SLQSModifyProfile
 - wds.h, [587](#)
- unpack_wds_SLQSModifyProfile_t, [381](#)
 - pExtErrorCode, [381](#)
- unpack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [589](#)
- unpack_wds_SLQSSetDHCPv4ClientConfig_t, [384](#)
 - pHwConfig, [384](#)
- unpack_wds_SLQSSet3GPPConfigItem
 - wds.h, [587](#)
- unpack_wds_SLQSSetIPFamilyPreference
 - wds.h, [588](#)
- unpack_wds_SLQSSetIPFamilyPreference_t, [381](#)
 - Tlvresult, [382](#)
- unpack_wds_SLQSSetPacketSrvStatusCallback
 - wds.h, [588](#)
- unpack_wds_SLQSSetPacketSrvStatusCallback_t, [382](#)
 - bearerID, [382](#)
 - conn_status, [382](#)
 - ipFamily, [382](#)
 - reconfigReqd, [383](#)
 - sessionEndReason, [383](#)
 - techName, [383](#)
 - verboseSessnEndReason, [383](#)
 - verboseSessnEndReasonType, [383](#)
- unpack_wds_SLQSSetWdsEventCallback
 - wds.h, [588](#)
- unpack_wds_SLQSSetWdsEventCallback_ind
 - wds.h, [589](#)
- unpack_wds_SLQSSetWdsEventCallback_ind_t, [383](#)
 - currDBTechAvail, [384](#)
 - currNWInfo, [384](#)
 - dBTechAvail, [384](#)
 - dBTechnology, [384](#)
 - dataSysStatAvail, [384](#)
 - dormancyStatAvail, [384](#)
 - dormancyStatus, [384](#)

- mipStatus, [384](#)
- mipstatAvail, [384](#)
- netInfoLen, [384](#)
- prefNetwork, [384](#)
- ratMask, [384](#)
- rx_bytes, [384](#)
- rx_pkts, [384](#)
- soMask, [384](#)
- tx_bytes, [384](#)
- tx_pkts, [384](#)
- xferStatAvail, [384](#)
- unpack_wds_SLQSSStartDataSession
 - wds.h, [589](#)
- unpack_wds_SLQSSStartDataSession_t, [385](#)
 - pFailureReason, [385](#)
 - pVerboseFailReasonType, [385](#)
 - pVerboseFailureReason, [385](#)
 - psid, [385](#)
- unpack_wds_SLQSSStopDataSession
 - wds.h, [590](#)
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, [590](#)
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [385](#)
 - apnName, [386](#)
 - bearerId, [386](#)
 - contextId, [386](#)
 - ipv4Address, [386](#)
 - ipv4GWAddress, [386](#)
 - ipv6Address, [386](#)
 - ipv6GWAddress, [386](#)
- unpack_wds_SetDefaultProfile
 - wds.h, [583](#)
- unpack_wds_SetDefaultProfileNum
 - wds.h, [584](#)
- unpack_wds_SetMobileIPProfile
 - wds.h, [584](#)
- unpack_wds_SetMobileIPProfile_t, [375](#)
- UnpackQmiProfileInfo
 - wds.h, [567](#)
- unpackWdsProfileParam, [387](#)
 - SlqsProfile3GPP, [387](#)
 - SlqsProfile3GPP2, [387](#)
- UpdateCompleteStatus
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [361](#)
- updateCompleteStatus
 - unpack_omaDmFotaTlv_t, [331](#)
- upinRetries
 - slotInf, [236](#)
 - uim_slotInfo, [255](#)
- upinState
 - slotInf, [236](#)
 - uim_slotInfo, [255](#)
- upukRetries
 - slotInf, [236](#)
 - uim_slotInfo, [255](#)
- usageMask
 - loc_sensorDataUsage, [75](#)
- userInputReq
 - unpack_omaDmConfigTlv_t, [329](#)
 - unpack_omaDmFotaTlv_t, [331](#)
- userInputTimeout
 - unpack_omaDmConfigTlv_t, [329](#)
 - unpack_omaDmFotaTlv_t, [331](#)
- Username
 - unpack_wds_SLQSGetRuntimeSettings_t, [381](#)
- username
 - unpack_wds_GetDefaultProfile_t, [370](#)
- usersize
 - unpack_wds_GetDefaultProfile_t, [371](#)
- VDOP
 - loc_precisionDilution, [74](#)
- val
 - unpack_qos_IPv6TrafCls_t, [334](#)
 - unpack_qos_Tos_t, [351](#)
- val_3GPP2Pri
 - unpack_qos_swiQosFlow_t, [350](#)
- val_3GPPImCn
 - unpack_qos_swiQosFlow_t, [350](#)
- val_3GPPResResidualBER
 - unpack_qos_swiQosFlow_t, [350](#)
- val_3GPPSigInd
 - unpack_qos_swiQosFlow_t, [350](#)
- val_3GPPTraHdlPri
 - unpack_qos_swiQosFlow_t, [350](#)
- ValidityCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, [325](#)
- ValidityCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, [325](#)
- value_length
 - DMScustSettingInfo, [30](#)
 - pack_dms_SetCustFeaturesV2_t, [173](#)
- verboseSessnEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [383](#)
- verboseSessnEndReasonType
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [383](#)
- verifyLeft
 - uim_remainingRetries, [252](#)
- verifyPIN
 - pack_uim_VerifyPin_t, [217](#)
- version
 - unpack_omaDmFotaTlv_t, [331](#)
 - unpack_qos_swiQosFilter_t, [346](#)
- versionlength
 - unpack_omaDmFotaTlv_t, [331](#)
- VoiceNumber
 - unpack_dms_GetVoiceNumber_t, [268](#)
- voiceNumberSize
 - unpack_dms_GetVoiceNumber_t, [268](#)
- WCDMACellInfo
 - nas_lteWcdmaCellInfo, [124](#)
- WCDMAECIOThreshListLen

- nas_WCDMAECIOThresh, [155](#)
- WCDMARSSIThreshListLen
 - nas_WCDMARSSIThresh, [157](#)
- WCDMASSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [321](#)
- WORD
 - SwiDataTypes.h, [543](#)
- wcdmaRRCState
 - nas_WCDMAInfoLTENeighborCell, [156](#)
- wds.h, [563](#)
 - PACK_WDS_IPV4, [567](#)
 - PACK_WDS_IPV6, [567](#)
 - pack_wds_GetConnectionRate, [567](#)
 - pack_wds_GetDefaultProfile, [568](#)
 - pack_wds_GetDefaultProfileNum, [568](#)
 - pack_wds_GetDormancyState, [569](#)
 - pack_wds_GetLastMobileIPError, [569](#)
 - pack_wds_GetMobileIP, [569](#)
 - pack_wds_GetMobileIPProfile, [570](#)
 - pack_wds_GetPacketStatus, [570](#)
 - pack_wds_GetSessionDuration, [571](#)
 - pack_wds_GetSessionState, [571](#)
 - pack_wds_RMSetTransferStatistics, [572](#)
 - pack_wds_SLQSCreateProfile, [573](#)
 - pack_wds_SLQSDeleteProfile, [573](#)
 - pack_wds_SLQSGet3GPPConfigItem, [574](#)
 - pack_wds_SLQSGetCurrDataSystemStat, [574](#)
 - pack_wds_SLQSGetDUNCallInfo, [575](#)
 - pack_wds_SLQSGetDataBearerTechnology, [575](#)
 - pack_wds_SLQSGetProfileSettings, [576](#)
 - pack_wds_SLQSGetRuntimeSettings, [576](#)
 - pack_wds_SLQSModifyProfile, [576](#)
 - pack_wds_SLQSSetDHCPv4ClientConfig, [578](#)
 - pack_wds_SLQSSet3GPPConfigItem, [577](#)
 - pack_wds_SLQSSetIPFamilyPreference, [577](#)
 - pack_wds_SLQSSetWdsEventCallback, [578](#)
 - pack_wds_SLQSStartDataSession, [578](#)
 - pack_wds_SLQSStopDataSession, [579](#)
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings, [579](#)
 - pack_wds_SetDefaultProfile, [572](#)
 - pack_wds_SetDefaultProfileNum, [572](#)
 - pack_wds_SetMobileIPProfile, [573](#)
 - unpack_wds_GetConnectionRate, [580](#)
 - unpack_wds_GetDefaultProfile, [580](#)
 - unpack_wds_GetDefaultProfileNum, [580](#)
 - unpack_wds_GetDormancyState, [581](#)
 - unpack_wds_GetLastMobileIPError, [581](#)
 - unpack_wds_GetMobileIP, [581](#)
 - unpack_wds_GetMobileIPProfile, [582](#)
 - unpack_wds_GetPacketStatus, [582](#)
 - unpack_wds_GetSessionDuration, [582](#)
 - unpack_wds_GetSessionState, [583](#)
 - unpack_wds_RMSetTransferStatistics, [583](#)
 - unpack_wds_SLQSCreateProfile, [584](#)
 - unpack_wds_SLQSDeleteProfile, [585](#)
 - unpack_wds_SLQSGet3GPPConfigItem, [585](#)
 - unpack_wds_SLQSGetCurrDataSystemStat, [585](#)
 - unpack_wds_SLQSGetDUNCallInfo, [586](#)
 - unpack_wds_SLQSGetDataBearerTechnology, [586](#)
 - unpack_wds_SLQSGetProfileSettings, [587](#)
 - unpack_wds_SLQSGetRuntimeSettings, [587](#)
 - unpack_wds_SLQSModifyProfile, [587](#)
 - unpack_wds_SLQSSetDHCPv4ClientConfig, [589](#)
 - unpack_wds_SLQSSet3GPPConfigItem, [587](#)
 - unpack_wds_SLQSSetIPFamilyPreference, [588](#)
 - unpack_wds_SLQSSetPacketSrvStatusCallback, [588](#)
 - unpack_wds_SLQSSetWdsEventCallback, [588](#)
 - unpack_wds_SLQSSetWdsEventCallback_ind, [589](#)
 - unpack_wds_SLQSStartDataSession, [589](#)
 - unpack_wds_SLQSStopDataSession, [590](#)
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings, [590](#)
 - unpack_wds_SetDefaultProfile, [583](#)
 - unpack_wds_SetDefaultProfileNum, [584](#)
 - unpack_wds_SetMobileIPProfile, [584](#)
 - UnpackQmiProfileInfo, [567](#)
- wds_Domain, [389](#)
 - domainLen, [390](#)
 - domainName, [390](#)
- wds_DomainNameList, [390](#)
 - domain, [390](#)
 - numInstances, [390](#)
- wds_GPRSQoS, [390](#)
 - delayClass, [391](#)
 - meanThroughputClass, [391](#)
 - peakThroughputClass, [391](#)
 - precedenceClass, [391](#)
 - reliabilityClass, [391](#)
- wds_IPV6AddressInfo, [391](#)
 - IPAddressV6, [391](#)
 - IPV6PrefixLen, [391](#)
- wds_IPV6GWAddressInfo, [391](#)
 - gwAddressV6, [392](#)
 - gwV6PrefixLen, [392](#)
- wds_PCSCFFQDNAddress, [392](#)
 - fqdnAddr, [392](#)
 - fqdnLen, [392](#)
- wds_PCSCFFQDNAddressList, [392](#)
 - numInstances, [393](#)
 - pcsfQDNAddress, [393](#)
- wds_PCSCFIPv4ServerAddressList, [393](#)
 - numInstances, [393](#)
 - pcsfIPv4Addr, [393](#)
- wds_ProfileIdentifier, [393](#)
 - profileIndex, [394](#)
 - profileType, [394](#)
- wds_UMTSMinQoS, [394](#)
 - deliveryErrSDU, [396](#)
 - grntDownlinkBitrate, [396](#)
 - grntUplinkBitrate, [396](#)
 - maxDownlinkBitrate, [396](#)

- maxSDUSize, [396](#)
- maxUplinkBitrate, [396](#)
- qosDeliveryOrder, [396](#)
- resBerRatio, [396](#)
- sduErrorRatio, [397](#)
- trafficClass, [397](#)
- trafficPriority, [397](#)
- transferDelay, [397](#)
- wds_currNetworkInfo, [387](#)
 - NetworkType, [389](#)
 - RATMask, [389](#)
 - SOMask, [389](#)
- wds_profileInfo, [394](#)
 - SlqsProfile3GPP, [394](#)
 - SlqsProfile3GPP2, [394](#)
- wdsDhcpv4HwConfig, [397](#)
 - chaddr, [397](#)
 - chaddrLen, [397](#)
 - hwType, [397](#)
- wdsDhcpv4Option, [397](#)
 - optCode, [398](#)
 - optVal, [398](#)
 - optValLen, [398](#)
- wdsDhcpv4OptionList, [398](#)
 - numOpt, [398](#)
 - pOptList, [398](#)
- wdsDhcpv4ProfileId, [398](#)
 - profileId, [399](#)
 - profileType, [399](#)
- xferStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [384](#)
- xid
 - pack_loc_SLQSLOCGetBestAvailPos_t, [181](#)
 - pack_qmi_t, [200](#)
 - unpack_qmi_t, [332](#)
- year
 - nas_timeInfo, [149](#)
 - nas_UniversalTime, [154](#)