

LinuxQMI SDK-Lite
SLQS04.00.02

Generated by Doxygen 1.8.6

Thu Sep 29 2016 09:54:13

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
6.2	Streaming Download Protocol (sdp)	20
6.2.1	Detailed Description	20
7	Namespace Documentation	21
7.1	Tables Namespace Reference	21
7.1.1	Detailed Description	21
8	Data Structure Documentation	23
8.1	_libSDP_FirmwareInfo_ Struct Reference	23
8.1.1	Detailed Description	23
8.1.2	Field Documentation	23

8.1.2.1	szCarrier_str	24
8.1.2.2	szCarrierPriversion_str	24
8.1.2.3	szFwversion_str	24
8.1.2.4	szModelid_str	24
8.1.2.5	szPackageid_str	24
8.1.2.6	szSku_str	24
8.2	appStats Struct Reference	24
8.2.1	Detailed Description	24
8.2.2	Field Documentation	26
8.2.2.1	aidLength	26
8.2.2.2	aidVal	26
8.2.2.3	appState	26
8.2.2.4	appType	26
8.2.2.5	persoFeature	26
8.2.2.6	persoRetries	26
8.2.2.7	persoState	26
8.2.2.8	persoUnblockRetries	27
8.2.2.9	pin1Retries	27
8.2.2.10	pin1State	27
8.2.2.11	pin2Retries	27
8.2.2.12	pin2State	27
8.2.2.13	puk1Retries	27
8.2.2.14	puk2Retries	27
8.2.2.15	univPin	27
8.3	CarrierImage_t Struct Reference	27
8.3.1	Detailed Description	27
8.3.2	Field Documentation	28
8.3.2.1	m_FwBuildId	28
8.3.2.2	m_FwImageld	28
8.3.2.3	m_nCarrierId	28
8.3.2.4	m_nFolderId	28
8.3.2.5	m_nStorage	28
8.3.2.6	m_PriBuildId	28
8.3.2.7	m_PriImageld	28
8.4	cdmaSSInfo Struct Reference	28
8.4.1	Detailed Description	28
8.4.2	Field Documentation	28
8.4.2.1	ecio	28
8.4.2.2	rss	28
8.5	connectionStatus Struct Reference	28

8.5.1	Detailed Description	28
8.5.2	Field Documentation	29
8.5.2.1	MDMCallDuration	29
8.5.2.2	MDMConnStatus	29
8.6	currNetworkInfo Struct Reference	29
8.6.1	Detailed Description	29
8.6.2	Field Documentation	29
8.6.2.1	NetworkType	29
8.6.2.2	RATMask	29
8.6.2.3	SOMask	29
8.7	dms_ActivationStatusTlv Struct Reference	29
8.7.1	Detailed Description	29
8.7.2	Field Documentation	30
8.7.2.1	activationStatus	30
8.7.2.2	TlvPresent	30
8.8	dms_OperatingModeTlv Struct Reference	30
8.8.1	Detailed Description	30
8.8.2	Field Documentation	30
8.8.2.1	operatingMode	30
8.8.2.2	TlvPresent	31
8.9	DMScustSettingInfo Struct Reference	31
8.9.1	Detailed Description	31
8.9.2	Field Documentation	31
8.9.2.1	cust_attr	31
8.9.2.2	cust_id	31
8.9.2.3	cust_value	31
8.9.2.4	id_length	31
8.9.2.5	value_length	31
8.10	DMScustSettingList Struct Reference	31
8.10.1	Detailed Description	32
8.10.2	Field Documentation	32
8.10.2.1	custSetting	32
8.10.2.2	list_type	32
8.10.2.3	num_instances	32
8.11	DMSgetCustFeatureV2 Struct Reference	32
8.11.1	Detailed Description	32
8.11.2	Field Documentation	33
8.11.2.1	pCustSettingInfo	33
8.11.2.2	pCustSettingList	33
8.11.2.3	pGetCustomInput	33

8.12 DMSgetCustomInput Struct Reference	33
8.12.1 Detailed Description	33
8.12.2 Field Documentation	33
8.12.2.1 cust_id	33
8.12.2.2 list_type	33
8.13 dunchannelRate Struct Reference	33
8.13.1 Detailed Description	33
8.13.2 Field Documentation	34
8.13.2.1 CurrChanRxRate	34
8.13.2.2 CurrChanTxRate	34
8.13.2.3 MaxChanRxRate	34
8.13.2.4 MaxChanTxRate	34
8.14 eTWSPLMNInfoTlv Struct Reference	34
8.14.1 Detailed Description	34
8.14.2 Field Documentation	34
8.14.2.1 ETWSPLMNInfo	34
8.14.2.2 TlvPresent	34
8.15 FMSImageElement 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 imageId	35
8.15.2.4 imageType	35
8.16 FMSImageIdElement Struct Reference	35
8.16.1 Detailed Description	35
8.16.2 Field Documentation	36
8.16.2.1 buildID	36
8.16.2.2 buildIDLength	36
8.16.2.3 failureCount	36
8.16.2.4 imageID	36
8.16.2.5 storageIndex	36
8.17 FMSImageIDEntries Struct Reference	36
8.17.1 Detailed Description	36
8.17.2 Field Documentation	36
8.17.2.1 executingImage	36
8.17.2.2 imageIDElement	36
8.17.2.3 imageIDSize	37
8.17.2.4 imageType	37
8.17.2.5 maxImages	37

8.18 FMSImageList Struct Reference	37
8.18.1 Detailed Description	37
8.18.2 Field Documentation	37
8.18.2.1 imageIDEntries	37
8.18.2.2 listSize	37
8.19 FMSPrefImageList Struct Reference	37
8.19.1 Detailed Description	37
8.19.2 Field Documentation	38
8.19.2.1 listEntries	38
8.19.2.2 listSize	38
8.20 hdrSSInfo Struct Reference	38
8.20.1 Detailed Description	38
8.20.2 Field Documentation	38
8.20.2.1 ecio	38
8.20.2.2 io	38
8.20.2.3 rssi	38
8.20.2.4 sinr	38
8.21 image_info_t Struct Reference	38
8.21.1 Field Documentation	38
8.21.1.1 buildID	38
8.21.1.2 buildIDLen	38
8.21.1.3 imageType	38
8.21.1.4 uniqueID	38
8.22 ipv6AddressInfo Struct Reference	39
8.22.1 Detailed Description	39
8.22.2 Field Documentation	39
8.22.2.1 IPAddressV6	39
8.22.2.2 IPV6PrefixLen	39
8.23 LibPackGPRSRequestedQoS Struct Reference	39
8.23.1 Detailed Description	39
8.23.2 Field Documentation	40
8.23.2.1 delayClass	40
8.23.2.2 meanThroughputClass	40
8.23.2.3 peakThroughputClass	40
8.23.2.4 precedenceClass	40
8.23.2.5 reliabilityClass	40
8.24 LibpackProfile3GPP Struct Reference	40
8.24.1 Detailed Description	41
8.24.2 Field Documentation	44
8.24.2.1 pAddrAllocPref	44

8.24.2.2	pAPNClass	44
8.24.2.3	pAPNDisabledFlag	44
8.24.2.4	pAPNName	44
8.24.2.5	pAPNnameSize	44
8.24.2.6	pAuthenticationPref	44
8.24.2.7	pGPRSMinimumQoS	44
8.24.2.8	pGPRSRequestedQos	44
8.24.2.9	plmCnFlag	45
8.24.2.10	pIPv4AddrPref	45
8.24.2.11	pIPv6AddPref	45
8.24.2.12	pPassword	45
8.24.2.13	pPasswordSize	45
8.24.2.14	pPcscfAddrUsingDhcp	45
8.24.2.15	pPcscfAddrUsingPCO	45
8.24.2.16	pPDNInactivTimeout	45
8.24.2.17	pDdpAccessConFlag	45
8.24.2.18	pDdpContext	45
8.24.2.19	pDdpDataCompType	45
8.24.2.20	pDdpHdrCompType	45
8.24.2.21	pPDpType	45
8.24.2.22	pPriDNSIPv4AddPref	45
8.24.2.23	pPriDNSIPv6addpref	45
8.24.2.24	pPrimaryID	45
8.24.2.25	pProfilename	45
8.24.2.26	pProfilenameSize	45
8.24.2.27	pQosClassID	45
8.24.2.28	pSecDNSIPv4AddPref	45
8.24.2.29	pSecDNSIPv6addpref	45
8.24.2.30	pSecondaryFlag	45
8.24.2.31	pTFTID1Params	45
8.24.2.32	pTFTID2Params	45
8.24.2.33	pUMTSMinQoS	45
8.24.2.34	pUMTSMinQosSigInd	45
8.24.2.35	pUMTSReqQoS	45
8.24.2.36	pUMTSReqQoSSigInd	45
8.24.2.37	pUsername	46
8.24.2.38	pUsernameSize	46
8.25	LibpackProfile3GPP2 Struct Reference	46
8.25.1	Detailed Description	46
8.25.2	Field Documentation	50

8.25.2.1	pAllowLinger	50
8.25.2.2	pAPNClass3GPP2	50
8.25.2.3	pAPNEnabled3GPP2	50
8.25.2.4	pApnString	50
8.25.2.5	pApnStringSize	50
8.25.2.6	pAppPriority	50
8.25.2.7	pAppType	50
8.25.2.8	pAuthPassword	50
8.25.2.9	pAuthPasswordSize	50
8.25.2.10	pAuthProtocol	50
8.25.2.11	pAuthRetryCount	50
8.25.2.12	pAuthTimeout	50
8.25.2.13	pDataMode	50
8.25.2.14	pDataRate	50
8.25.2.15	pIpcpAckTimeout	50
8.25.2.16	pIpcpCreqRetryCount	50
8.25.2.17	pIscPcscfAddressNedded	50
8.25.2.18	pLcpAckTimeout	50
8.25.2.19	pLcpCreqRetryCount	51
8.25.2.20	pNegoDnsSrvrPref	51
8.25.2.21	pPDNInactivTimeout3GPP2	51
8.25.2.22	pPdnType	51
8.25.2.23	pPppSessCloseTimer1x	51
8.25.2.24	pPppSessCloseTimerDO	51
8.25.2.25	pPrimaryV4DnsAddress	51
8.25.2.26	pPriV6DnsAddress	51
8.25.2.27	pRATType	51
8.25.2.28	pSecondaryV4DnsAddress	51
8.25.2.29	pSecV6DnsAddress	51
8.25.2.30	pUserId	51
8.25.2.31	pUserIdSize	51
8.26	LibPackprofile_3GPP Struct Reference	51
8.26.1	Detailed Description	52
8.26.2	Field Documentation	55
8.26.2.1	pAddrAllocPref	55
8.26.2.2	pAPNClass	55
8.26.2.3	pAPNDisabledFlag	55
8.26.2.4	pAPNName	55
8.26.2.5	pAPNnameSize	56
8.26.2.6	pAuthenticationPref	56

8.26.2.7	pGPRMinimumQoS	56
8.26.2.8	pGPRSRequestedQos	56
8.26.2.9	plmCnFlag	56
8.26.2.10	pIPv4AddrPref	56
8.26.2.11	pIPv6AddPref	56
8.26.2.12	pPassword	56
8.26.2.13	pPasswordSize	56
8.26.2.14	pPcscfAddrUsingDhcp	56
8.26.2.15	pPcscfAddrUsingPCO	56
8.26.2.16	pPDNInactivTimeout	56
8.26.2.17	pPdpAccessConFlag	56
8.26.2.18	pPdpContext	56
8.26.2.19	pPdpDataCompType	56
8.26.2.20	pPdpHdrCompType	56
8.26.2.21	pPDType	56
8.26.2.22	pPriDNSIPv4AddPref	56
8.26.2.23	pPriDNSIPv6addpref	56
8.26.2.24	pPrimaryID	56
8.26.2.25	pProfilename	56
8.26.2.26	pProfilenameSize	56
8.26.2.27	pQosClassID	56
8.26.2.28	pSecDNSIPv4AddPref	56
8.26.2.29	pSecDNSIPv6addpref	56
8.26.2.30	pSecondaryFlag	56
8.26.2.31	pTFTID1Params	56
8.26.2.32	pTFTID2Params	56
8.26.2.33	pUMTSMinQoS	57
8.26.2.34	pUMTSMinQoSSigInd	57
8.26.2.35	pUMTSReqQoS	57
8.26.2.36	pUMTSReqQoSSigInd	57
8.26.2.37	pUsername	57
8.26.2.38	pUsernameSize	57
8.27	LibPackprofile_3GPP2 Struct Reference	57
8.27.1	Detailed Description	57
8.27.2	Field Documentation	61
8.27.2.1	pAllowLinger	61
8.27.2.2	pAPNClass3GPP2	61
8.27.2.3	pAPNEnabled3GPP2	61
8.27.2.4	pApnString	61
8.27.2.5	pApnStringSize	61

8.27.2.6	pAppPriority	61
8.27.2.7	pAppType	61
8.27.2.8	pAuthPassword	61
8.27.2.9	pAuthPassword_tSize	61
8.27.2.10	pAuthProtocol	61
8.27.2.11	pAuthRetryCount	61
8.27.2.12	pAuthTimeout	61
8.27.2.13	pDataMode	61
8.27.2.14	pDataRate	61
8.27.2.15	plpcpAckTimeout	61
8.27.2.16	plpcpCreqRetryCount	61
8.27.2.17	plsPcscfAddressNedded	61
8.27.2.18	pLcpAckTimeout	61
8.27.2.19	pLcpCreqRetryCount	62
8.27.2.20	pNegoDnsSrvrPref	62
8.27.2.21	pPDNInactivTimeout3GPP2	62
8.27.2.22	pPdnType	62
8.27.2.23	pPppSessCloseTimer1x	62
8.27.2.24	pPppSessCloseTimerDO	62
8.27.2.25	pPrimaryV4DnsAddress	62
8.27.2.26	pPriV6DnsAddress	62
8.27.2.27	pRATType	62
8.27.2.28	pSecondaryV4DnsAddress	62
8.27.2.29	pSecV6DnsAddress	62
8.27.2.30	pUserId	62
8.27.2.31	pUserIdSize	62
8.28	LibPackQosClassID Struct Reference	62
8.28.1	Detailed Description	62
8.28.2	Field Documentation	63
8.28.2.1	gDIBitRate	63
8.28.2.2	gUIBitRate	63
8.28.2.3	maxDIBitRate	63
8.28.2.4	maxUIBitRate	63
8.28.2.5	QCI	63
8.29	LibPackTFTIDParams Struct Reference	63
8.29.1	Detailed Description	63
8.29.2	Field Documentation	64
8.29.2.1	destPortRangeEnd	64
8.29.2.2	destPortRangeStart	64
8.29.2.3	eValid	64

8.29.2.4	filterId	64
8.29.2.5	flowLabel	64
8.29.2.6	IPSECSPi	64
8.29.2.7	ipVersion	65
8.29.2.8	nextHeader	65
8.29.2.9	pSourceIP	65
8.29.2.10	sourceIPMask	65
8.29.2.11	srcPortRangeEnd	65
8.29.2.12	srcPortRangeStart	65
8.29.2.13	tosMask	65
8.30	LibPackUMTSQoS Struct Reference	65
8.30.1	Detailed Description	65
8.30.2	Field Documentation	67
8.30.2.1	deliveryErrSDU	67
8.30.2.2	grntDownlinkBitrate	67
8.30.2.3	grntUplinkBitrate	67
8.30.2.4	maxDownlinkBitrate	67
8.30.2.5	maxSDUSize	67
8.30.2.6	maxUplinkBitrate	67
8.30.2.7	qosDeliveryOrder	67
8.30.2.8	resBerRatio	67
8.30.2.9	sduErrorRatio	67
8.30.2.10	trafficClass	67
8.30.2.11	trafficPriority	67
8.30.2.12	transferDelay	67
8.31	LibPackUMTSReqQoSsigInd Struct Reference	67
8.31.1	Detailed Description	67
8.31.2	Field Documentation	68
8.31.2.1	SigInd	68
8.31.2.2	UMTSReqQoS	68
8.32	loc_BdsSV Struct Reference	68
8.32.1	Detailed Description	68
8.32.2	Field Documentation	68
8.32.2.1	id	68
8.32.2.2	mask	68
8.33	loc_BdsSVInfo Struct Reference	68
8.33.1	Detailed Description	69
8.33.2	Field Documentation	69
8.33.2.1	len	69
8.33.2.2	pSV	69

8.34	loc_CellDb Struct Reference	69
8.34.1	Detailed Description	69
8.34.2	Field Documentation	69
8.34.2.1	mask	69
8.35	loc_ClkInfo Struct Reference	70
8.35.1	Detailed Description	70
8.35.2	Field Documentation	71
8.35.2.1	mask	71
8.36	loc_GnssData Struct Reference	71
8.36.1	Detailed Description	71
8.36.2	Field Documentation	72
8.36.2.1	mask	72
8.37	loc_gpsTime Struct Reference	72
8.37.1	Detailed Description	73
8.37.2	Field Documentation	73
8.37.2.1	gpsTimeOfWeekMs	73
8.37.2.2	gpsWeek	73
8.38	loc_LocApplicationInfo Struct Reference	73
8.38.1	Detailed Description	73
8.38.2	Field Documentation	74
8.38.2.1	appNameLength	74
8.38.2.2	appProviderLength	74
8.38.2.3	appVersionLength	74
8.38.2.4	appVersionValid	74
8.38.2.5	pAppName	74
8.38.2.6	pAppProvider	74
8.38.2.7	pAppVersion	74
8.39	loc_precisionDilution Struct Reference	74
8.39.1	Detailed Description	74
8.39.2	Field Documentation	75
8.39.2.1	HDOP	75
8.39.2.2	PDOP	75
8.39.2.3	VDOP	75
8.40	loc_sensorDataUsage Struct Reference	75
8.40.1	Detailed Description	75
8.40.2	Field Documentation	75
8.40.2.1	aidingIndicatorMask	76
8.40.2.2	usageMask	76
8.41	loc_SV Struct Reference	76
8.41.1	Detailed Description	76

8.41.2	Field Documentation	76
8.41.2.1	id	76
8.41.2.2	mask	76
8.41.2.3	system	76
8.42	loc_SVInfo Struct Reference	76
8.42.1	Detailed Description	77
8.42.2	Field Documentation	77
8.42.2.1	len	77
8.42.2.2	pSV	77
8.43	loc_svUsedforFix Struct Reference	77
8.43.1	Detailed Description	77
8.43.2	Field Documentation	78
8.43.2.1	gnssSvUsedList	78
8.43.2.2	gnssSvUsedList_len	78
8.44	IteSSInfo Struct Reference	78
8.44.1	Detailed Description	78
8.44.2	Field Documentation	78
8.44.2.1	rsrp	78
8.44.2.2	rsrq	78
8.44.2.3	rssi	78
8.44.2.4	snr	78
8.45	messageModeTlv Struct Reference	78
8.45.1	Detailed Description	78
8.45.2	Field Documentation	79
8.45.2.1	MessageModelInfo	79
8.45.2.2	TlvPresent	79
8.46	nas_acqOrderPref Struct Reference	79
8.46.1	Detailed Description	79
8.46.2	Field Documentation	79
8.46.2.1	acqOrdeLen	79
8.46.2.2	pAcqOrder	79
8.47	nas_AddCDMASysInfo Struct Reference	79
8.47.1	Detailed Description	79
8.47.2	Field Documentation	80
8.47.2.1	geoSysIdx	80
8.47.2.2	regPrd	80
8.48	nas_AddSysInfo Struct Reference	80
8.48.1	Detailed Description	80
8.48.2	Field Documentation	80
8.48.2.1	cellBroadcastCap	80

8.48.2.2	geoSysIdx	80
8.49	nas_CallBarringSysInfo Struct Reference	80
8.49.1	Detailed Description	81
8.49.2	Field Documentation	81
8.49.2.1	csBarStatus	81
8.49.2.2	psBarStatus	81
8.50	nas_callBarStatus Struct Reference	81
8.50.1	Detailed Description	81
8.50.2	Field Documentation	82
8.50.2.1	csBarStatus	82
8.50.2.2	psBarStatus	82
8.51	nas_CDMAECIOThresh Struct Reference	82
8.51.1	Detailed Description	82
8.51.2	Field Documentation	82
8.51.2.1	CDMAECIOThreshListLen	82
8.51.2.2	pCDMAECIOThreshList	83
8.52	nas_CDMAInfo Struct Reference	83
8.52.1	Detailed Description	83
8.52.2	Field Documentation	83
8.52.2.1	baseId	83
8.52.2.2	baseLat	83
8.52.2.3	baseLong	83
8.52.2.4	nid	84
8.52.2.5	refpn	84
8.52.2.6	sid	84
8.53	nas_CDMARSSIThresh Struct Reference	84
8.53.1	Detailed Description	84
8.53.2	Field Documentation	84
8.53.2.1	CDMARSSIThreshListLen	84
8.53.2.2	pCDMARSSIThreshList	84
8.54	nas_CDMA SysInfo Struct Reference	84
8.54.1	Detailed Description	85
8.54.2	Field Documentation	87
8.54.2.1	baseId	87
8.54.2.2	baseLat	87
8.54.2.3	baseLong	87
8.54.2.4	bsInfoValid	87
8.54.2.5	bsPRev	87
8.54.2.6	bsPRevValid	87
8.54.2.7	ccsSupported	87

8.54.2.8	ccsSupportedValid	87
8.54.2.9	cdmaSysIdValid	87
8.54.2.10	isSysPrIMatch	87
8.54.2.11	isSysPrIMatchValid	87
8.54.2.12	MCC	87
8.54.2.13	MNC	87
8.54.2.14	networkID	87
8.54.2.15	networkIdValid	88
8.54.2.16	packetZone	88
8.54.2.17	packetZoneValid	88
8.54.2.18	pRevInUse	88
8.54.2.19	pRevInUseValid	88
8.54.2.20	sysInfoCDMA	88
8.54.2.21	systemID	88
8.55	nas_CDMASysInfoExt Struct Reference	88
8.55.1	Detailed Description	88
8.55.2	Field Documentation	88
8.55.2.1	imsi_11_12	88
8.55.2.2	MCC	88
8.56	nas_cellParams Struct Reference	88
8.56.1	Detailed Description	89
8.56.2	Field Documentation	89
8.56.2.1	pci	89
8.56.2.2	rsrp	89
8.56.2.3	rsrq	89
8.56.2.4	rssI	89
8.56.2.5	srxlev	89
8.57	nas_CommInfo Struct Reference	89
8.57.1	Detailed Description	89
8.57.2	Field Documentation	91
8.57.2.1	imsRegState	91
8.57.2.2	modemMode	91
8.57.2.3	psState	91
8.57.2.4	systemMode	91
8.57.2.5	temperature	91
8.58	nas_CSGID Struct Reference	91
8.58.1	Detailed Description	91
8.58.2	Field Documentation	91
8.58.2.1	id	92
8.58.2.2	mcc	92

8.58.2.3	mnc	92
8.58.2.4	mncPcsDigits	92
8.58.2.5	rat	92
8.59	nas_currentPLMN Struct Reference	92
8.59.1	Detailed Description	92
8.59.2	Field Documentation	92
8.59.2.1	MCC	92
8.59.2.2	MNC	92
8.59.2.3	netDescr	93
8.59.2.4	netDescrLength	93
8.60	nas_dataSrvCapabilities Struct Reference	93
8.60.1	Detailed Description	93
8.60.2	Field Documentation	93
8.60.2.1	dataCapabilities	93
8.60.2.2	dataCapabilitiesLen	93
8.61	nas_detailSvcInfo Struct Reference	93
8.61.1	Detailed Description	94
8.61.2	Field Documentation	95
8.61.2.1	hdrHybrid	95
8.61.2.2	hdrSrvStatus	95
8.61.2.3	isSysForbidden	95
8.61.2.4	srvCapability	95
8.61.2.5	srvStatus	95
8.62	nas_ecioListElement Struct Reference	95
8.62.1	Detailed Description	95
8.62.2	Field Documentation	95
8.62.2.1	ecio	95
8.62.2.2	radiolf	95
8.63	nas_errorRateListElement Struct Reference	96
8.63.1	Detailed Description	96
8.63.2	Field Documentation	96
8.63.2.1	errorRate	96
8.63.2.2	radiolf	96
8.64	nas_GERANInfo Struct Reference	96
8.64.1	Detailed Description	97
8.64.2	Field Documentation	98
8.64.2.1	arfcn	98
8.64.2.2	bsic	98
8.64.2.3	cellID	98
8.64.2.4	insNmrCellInfo	98

8.64.2.5	lac	98
8.64.2.6	nmrInst	98
8.64.2.7	plmn	98
8.64.2.8	rxLev	98
8.64.2.9	timingAdvance	98
8.65	nas_geranInstInfo Struct Reference	98
8.65.1	Detailed Description	99
8.65.2	Field Documentation	99
8.65.2.1	geranArfcn	99
8.65.2.2	geranBsicBcc	99
8.65.2.3	geranBsicNcc	99
8.65.2.4	geranRssi	99
8.66	nas_gsmCellInfo Struct Reference	99
8.66.1	Detailed Description	99
8.66.2	Field Documentation	100
8.66.2.1	arfcn	100
8.66.2.2	band1900	100
8.66.2.3	bsicId	100
8.66.2.4	cellIdValid	100
8.66.2.5	rssi	100
8.66.2.6	srxlev	100
8.67	nas_GSMRSSIThresh Struct Reference	100
8.67.1	Detailed Description	100
8.67.2	Field Documentation	101
8.67.2.1	GSMRSSIThreshListLen	101
8.67.2.2	pGSMRSSIThreshList	101
8.68	nas_GSMSrvStatusInfo Struct Reference	101
8.68.1	Detailed Description	101
8.68.2	Field Documentation	102
8.68.2.1	isPrefDataPath	102
8.68.2.2	srvStatus	102
8.68.2.3	trueSrvStatus	102
8.69	nas_GSMSysInfo Struct Reference	102
8.69.1	Detailed Description	102
8.69.2	Field Documentation	104
8.69.2.1	cellId	104
8.69.2.2	cellIdValid	104
8.69.2.3	dtmSupp	104
8.69.2.4	dtmSuppValid	104
8.69.2.5	egprsSupp	104

8.69.2.6	egprsSuppValid	104
8.69.2.7	lac	104
8.69.2.8	lacValid	104
8.69.2.9	MCC	104
8.69.2.10	MNC	104
8.69.2.11	networkIdValid	104
8.69.2.12	regRejectInfoValid	104
8.69.2.13	rejCause	105
8.69.2.14	rejectSrvDomain	105
8.69.2.15	sysInfoGSM	105
8.70	nas_HDRECIOThresh Struct Reference	105
8.70.1	Detailed Description	105
8.70.2	Field Documentation	105
8.70.2.1	HDRECIOThreshListLen	105
8.70.2.2	pHDRECIOThreshList	105
8.71	nas_HDRIOTThresh Struct Reference	105
8.71.1	Detailed Description	105
8.71.2	Field Documentation	106
8.71.2.1	HDRIOTThreshListLen	106
8.71.2.2	pHDRIOTThreshList	106
8.72	nas_HDRRSSIThresh Struct Reference	106
8.72.1	Detailed Description	106
8.72.2	Field Documentation	106
8.72.2.1	HDRRSSIThreshListLen	106
8.72.2.2	pHRRSSIThreshList	106
8.73	nas_HDRSINRThreshold Struct Reference	106
8.73.1	Detailed Description	107
8.73.2	Field Documentation	107
8.73.2.1	HDRSINRThreshListLen	107
8.73.2.2	pHDRSINRThreshList	107
8.74	nas_HDRSysInfo Struct Reference	107
8.74.1	Detailed Description	107
8.74.2	Field Documentation	109
8.74.2.1	hdrActiveProt	109
8.74.2.2	hdrActiveProtValid	109
8.74.2.3	hdrPersonality	109
8.74.2.4	hdrPersonalityValid	109
8.74.2.5	is856SysId	109
8.74.2.6	is856SysIdValid	109
8.74.2.7	isSysPrIMatch	109

8.74.2.8	isSysPrIMatchValid	109
8.74.2.9	sysInfoHDR	109
8.75	nas_infoInterFreq Struct Reference	109
8.75.1	Detailed Description	109
8.75.2	Field Documentation	110
8.75.2.1	cell_resel_priority	110
8.75.2.2	cellInterFreqParams	110
8.75.2.3	cells_len	110
8.75.2.4	earfcn	110
8.75.2.5	threshXHigh	110
8.75.2.6	threshXLow	110
8.76	nas_lteGsmCellInfo Struct Reference	110
8.76.1	Detailed Description	110
8.76.2	Field Documentation	111
8.76.2.1	cellReselPriority	111
8.76.2.2	cells_len	111
8.76.2.3	GsmCellInfo	111
8.76.2.4	nccPermitted	111
8.76.2.5	threshGsmHigh	111
8.76.2.6	threshGsmLow	111
8.77	nas_LTEInfo Struct Reference	111
8.77.1	Detailed Description	112
8.77.2	Field Documentation	113
8.77.2.1	band	113
8.77.2.2	bandwidth	113
8.77.2.3	emmConnState	113
8.77.2.4	emmState	113
8.77.2.5	emmSubState	113
8.77.2.6	RXChan	113
8.77.2.7	TXChan	113
8.78	nas_LTEInfoInterfreq Struct Reference	114
8.78.1	Detailed Description	114
8.78.2	Field Documentation	114
8.78.2.1	freqsLen	114
8.78.2.2	InfoInterfreq	114
8.78.2.3	ueInIdle	114
8.79	nas_LTEInfoIntrafreq Struct Reference	114
8.79.1	Detailed Description	115
8.79.2	Field Documentation	116
8.79.2.1	CellParams	116

8.79.2.2	cellReselPriority	116
8.79.2.3	cellsLen	116
8.79.2.4	earfcn	116
8.79.2.5	globalCellId	116
8.79.2.6	plmn	116
8.79.2.7	servingCellId	116
8.79.2.8	sIntraSearch	116
8.79.2.9	sNonIntraSearch	116
8.79.2.10	tac	116
8.79.2.11	threshServingLow	116
8.79.2.12	ueInIdle	116
8.80	nas_LTEInfoNeighboringGSM Struct Reference	116
8.80.1	Detailed Description	117
8.80.2	Field Documentation	117
8.80.2.1	freqsLen	117
8.80.2.2	LteGsmCellInfo	117
8.80.2.3	ueInIdle	117
8.81	nas_LTEInfoNeighboringWCDMA Struct Reference	117
8.81.1	Detailed Description	117
8.81.2	Field Documentation	118
8.81.2.1	freqsLen	118
8.81.2.2	LTEWCDMACellInfo	118
8.81.2.3	ueInIdle	118
8.82	nas_lteRsrpInformation Struct Reference	118
8.82.1	Detailed Description	118
8.82.2	Field Documentation	118
8.82.2.1	rsrplevel	118
8.83	nas_LTERS RPThresh Struct Reference	118
8.83.1	Detailed Description	118
8.83.2	Field Documentation	118
8.83.2.1	LTERS RPThreshListLen	119
8.83.2.2	pLTERS RPThreshList	119
8.84	nas_LTERS RQThresh Struct Reference	119
8.84.1	Detailed Description	119
8.84.2	Field Documentation	119
8.84.2.1	LTERS RQThreshListLen	119
8.84.2.2	pLTERS RQThreshList	119
8.85	nas_LTERSS IThresh Struct Reference	119
8.85.1	Detailed Description	119
8.85.2	Field Documentation	120

8.85.2.1	LTERSSIThreshListLen	120
8.85.2.2	pLTERSSIThreshList	120
8.86	nas_LTESigRptConfig Struct Reference	120
8.86.1	Detailed Description	120
8.86.2	Field Documentation	120
8.86.2.1	avgPeriod	120
8.86.2.2	rptRate	120
8.87	nas_lteSnrinformation Struct Reference	120
8.87.1	Detailed Description	121
8.87.2	Field Documentation	121
8.87.2.1	snrlevel	121
8.88	nas_LTESNRThreshold Struct Reference	121
8.88.1	Detailed Description	121
8.88.2	Field Documentation	121
8.88.2.1	LTESNRThreshListLen	121
8.88.2.2	pLTESNRThreshList	121
8.89	nas_LTESysInfo Struct Reference	121
8.89.1	Detailed Description	122
8.89.2	Field Documentation	123
8.89.2.1	cellId	123
8.89.2.2	cellIdValid	123
8.89.2.3	lac	123
8.89.2.4	lacValid	123
8.89.2.5	MCC	124
8.89.2.6	MNC	124
8.89.2.7	networkIdValid	124
8.89.2.8	regRejectInfoValid	124
8.89.2.9	rejCause	124
8.89.2.10	rejectSrvDomain	124
8.89.2.11	sysInfoLTE	124
8.89.2.12	tac	124
8.89.2.13	tacValid	124
8.90	nas_lteWcdmaCellInfo Struct Reference	124
8.90.1	Detailed Description	124
8.90.2	Field Documentation	125
8.90.2.1	cellReselPriority	125
8.90.2.2	cellsLen	125
8.90.2.3	threshXhigh	125
8.90.2.4	threshXlow	125
8.90.2.5	uarfcn	125

8.90.2.6	WCDMACellInfo	125
8.91	nas_MNRInfo Struct Reference	125
8.91.1	Detailed Description	125
8.91.2	Field Documentation	125
8.91.2.1	mcc	126
8.91.2.2	mnc	126
8.91.2.3	rat	126
8.92	nas_netSelectionPref Struct Reference	126
8.92.1	Detailed Description	126
8.92.2	Field Documentation	126
8.92.2.1	mcc	126
8.92.2.2	mnc	126
8.92.2.3	netReg	126
8.93	nas_nmrCellInfo Struct Reference	126
8.93.1	Detailed Description	127
8.93.2	Field Documentation	127
8.93.2.1	nmrArfcn	127
8.93.2.2	nmrBsic	127
8.93.2.3	nmrCellID	127
8.93.2.4	nmrLac	127
8.93.2.5	nmrPlmn	127
8.93.2.6	nmrRxLev	127
8.94	nas_PhyCaAggPcellInfo Struct Reference	128
8.94.1	Detailed Description	128
8.94.2	Field Documentation	128
8.94.2.1	dl_bw_value	128
8.94.2.2	freq	128
8.94.2.3	iLTEbandValue	128
8.94.2.4	pci	128
8.94.2.5	TlvPresent	128
8.95	nas_PhyCaAggScellDIBw Struct Reference	128
8.95.1	Detailed Description	129
8.95.2	Field Documentation	129
8.95.2.1	dl_bw_value	129
8.95.2.2	TlvPresent	129
8.96	nas_PhyCaAggScellIndex Struct Reference	129
8.96.1	Detailed Description	129
8.96.2	Field Documentation	129
8.96.2.1	scell_idx	129
8.96.2.2	TlvPresent	129

8.97 nas_PhyCaAggScellIndType Struct Reference	129
8.97.1 Detailed Description	130
8.97.2 Field Documentation	130
8.97.2.1 freq	130
8.97.2.2 pci	130
8.97.2.3 scell_state	130
8.97.2.4 TlvPresent	130
8.98 nas_PhyCaAggScellInfo Struct Reference	130
8.98.1 Detailed Description	130
8.98.2 Field Documentation	133
8.98.2.1 dl_bw_value	133
8.98.2.2 freq	133
8.98.2.3 iLTEbandValue	133
8.98.2.4 pci	133
8.98.2.5 scell_state	133
8.98.2.6 TlvPresent	133
8.99 nas_qaQmi3Gpp2TimeZone Struct Reference	133
8.99.1 Detailed Description	133
8.99.2 Field Documentation	133
8.99.2.1 daylightSavings	133
8.99.2.2 leapSeconds	133
8.99.2.3 localTimeOffset	134
8.100 nas_QmiNas3GppNetworkInfo Struct Reference	134
8.100.1 Detailed Description	134
8.100.2 Field Documentation	134
8.100.2.1 Description	134
8.100.2.2 Forbidden	134
8.100.2.3 InUse	134
8.100.2.4 MCC	134
8.100.2.5 MNC	134
8.100.2.6 Preferred	134
8.100.2.7 Roaming	134
8.101 nas_QmiNas3GppNetworkRAT Struct Reference	134
8.101.1 Detailed Description	134
8.101.2 Field Documentation	135
8.101.2.1 MCC	135
8.101.2.2 MNC	135
8.101.2.3 RAT	135
8.102 nas_QmiNasPcsDigit Struct Reference	135
8.102.1 Detailed Description	135

8.102.2 Field Documentation	135
8.102.2.1 includes_pcs_digit	136
8.102.2.2 MCC	136
8.102.2.3 MNC	136
8.103nas_RejectReasonTlv Struct Reference	136
8.103.1 Detailed Description	136
8.103.2 Field Documentation	136
8.103.2.1 rejectCause	136
8.103.2.2 serviceDomain	136
8.103.2.3 TlvPresent	136
8.104nas_RFInfoTlv Struct Reference	136
8.104.1 Detailed Description	136
8.104.2 Field Documentation	136
8.104.2.1 activeBandClass	136
8.104.2.2 activeChannel	137
8.104.2.3 radiolInterface	137
8.104.2.4 radiolInterfaceSize	137
8.104.2.5 TlvPresent	137
8.105nas_roamIndList Struct Reference	137
8.105.1 Detailed Description	137
8.105.2 Field Documentation	137
8.105.2.1 numInstances	137
8.105.2.2 radiolInterface	137
8.105.2.3 roamIndicator	138
8.106nas_rsrqInformation Struct Reference	138
8.106.1 Detailed Description	138
8.106.2 Field Documentation	138
8.106.2.1 radiolf	138
8.106.2.2 rsrq	138
8.107nas_RxSigInfo Struct Reference	138
8.107.1 Detailed Description	138
8.107.2 Field Documentation	139
8.107.2.1 isRadioTuned	139
8.107.2.2 rsrp	139
8.107.2.3 rxChainIndex	139
8.107.2.4 rxPower	139
8.108nas_rxSignalStrengthListElement Struct Reference	139
8.108.1 Detailed Description	139
8.108.2 Field Documentation	140
8.108.2.1 radiolf	140

8.108.2.2 rxSignalStrength	140
8.109nas_SccRxInfo Struct Reference	140
8.109.1 Detailed Description	140
8.109.2 Field Documentation	141
8.109.2.1 numInstances	141
8.109.2.2 rsrq	141
8.109.2.3 sigInfo	141
8.109.2.4 snr	141
8.109.2.5 TlvPresent	141
8.110nas_servSystem Struct Reference	141
8.110.1 Detailed Description	141
8.110.2 Field Documentation	142
8.110.2.1 csAttachState	142
8.110.2.2 numRadioInterfaces	142
8.110.2.3 psAttachState	142
8.110.2.4 radiolInterface	142
8.110.2.5 regState	142
8.110.2.6 selNetwork	142
8.111nas_SignalStrengthTlv Struct Reference	143
8.111.1 Detailed Description	143
8.111.2 Field Documentation	143
8.111.2.1 radiolInterface	143
8.111.2.2 signalStrength	143
8.111.2.3 TlvPresent	143
8.112nas_SLQSSignalStrengthsIndReq Struct Reference	143
8.112.1 Detailed Description	143
8.112.2 Field Documentation	144
8.112.2.1 ecioDelta	144
8.112.2.2 ecioThresholdList	144
8.112.2.3 ecioThresholdListLen	144
8.112.2.4 ioDelta	144
8.112.2.5 lteRsrpDelta	144
8.112.2.6 lteSnrDelta	144
8.112.2.7 rsrqDelta	144
8.112.2.8 rxSignalStrengthDelta	144
8.112.2.9 sinrDelta	144
8.112.2.10sinrThresholdList	144
8.112.2.11sinrThresholdListLen	144
8.113nas_SLQSSignalStrengthsInformation Struct Reference	144
8.113.1 Detailed Description	144

8.113.2 Field Documentation	145
8.113.2.1 eciInfo	145
8.113.2.2 errorRateInfo	145
8.113.2.3 io	145
8.113.2.4 lteRsrpinfo	145
8.113.2.5 lteSnrinfo	145
8.113.2.6 rsrqInfo	145
8.113.2.7 rxSignalStrengthInfo	145
8.113.2.8 sinr	145
8.114nas_SLQSSignalStrengthsTlv Struct Reference	145
8.114.1 Detailed Description	145
8.114.2 Field Documentation	145
8.114.2.1 sSLQSSignalStrengthsInfo	145
8.114.2.2 TlvPresent	145
8.115nas_SrvStatusInfo Struct Reference	145
8.115.1 Detailed Description	146
8.115.2 Field Documentation	146
8.115.2.1 isPrefDataPath	146
8.115.2.2 srvStatus	146
8.116nas_sysInfoCommon Struct Reference	146
8.116.1 Detailed Description	146
8.116.2 Field Documentation	148
8.116.2.1 isSysForbidden	148
8.116.2.2 isSysForbiddenValid	148
8.116.2.3 roamStatus	148
8.116.2.4 roamStatusValid	148
8.116.2.5 srvCapability	148
8.116.2.6 srvCapabilityValid	148
8.116.2.7 srvDomain	148
8.116.2.8 srvDomainValid	148
8.117nas_TDSCDMAECIOThresh Struct Reference	149
8.117.1 Detailed Description	149
8.117.2 Field Documentation	149
8.117.2.1 pTDSCDMAECIOThreshList	149
8.117.2.2 TDSCDMAECIOThreshListLen	149
8.118nas_TDSCDMARSCPThresh Struct Reference	149
8.118.1 Detailed Description	149
8.118.2 Field Documentation	149
8.118.2.1 pTDSCDMARSCPThreshList	149
8.118.2.2 TDSCDMARSCPThreshListLen	150

8.119nas_TDSCDMARSSIThresh Struct Reference	150
8.119.1 Detailed Description	150
8.119.2 Field Documentation	150
8.119.2.1 pTDSCDMARSSIThreshList	150
8.119.2.2 TDSCDMARSSIThreshListLen	150
8.120nas_TDSCDMASINRThresh Struct Reference	150
8.120.1 Detailed Description	150
8.120.2 Field Documentation	150
8.120.2.1 pTDSCDMASINRThreshList	151
8.120.2.2 TDSCDMASINRThreshListLen	151
8.121nas_timeInfo Struct Reference	151
8.121.1 Detailed Description	151
8.121.2 Field Documentation	152
8.121.2.1 day	152
8.121.2.2 dayLtSavingAdj	152
8.121.2.3 dayOfWeek	152
8.121.2.4 hour	152
8.121.2.5 minute	152
8.121.2.6 month	152
8.121.2.7 radiolInterface	152
8.121.2.8 second	152
8.121.2.9 timeZone	152
8.121.2.10TlvPresent	152
8.121.2.11year	152
8.122nas_UMTSInfo Struct Reference	152
8.122.1 Detailed Description	153
8.122.2 Field Documentation	154
8.122.2.1 cellID	154
8.122.2.2 ecio	154
8.122.2.3 geranInst	154
8.122.2.4 GeranInstInfo	154
8.122.2.5 lac	154
8.122.2.6 plmn	154
8.122.2.7 psc	154
8.122.2.8 rscp	154
8.122.2.9 uarfcn	154
8.122.2.10umtsInst	154
8.122.2.11UMTSInstInfo	154
8.123nas_UMTSinstInfo Struct Reference	154
8.123.1 Detailed Description	154

8.123.2 Field Documentation	155
8.123.2.1 umtsEcio	155
8.123.2.2 umtsPsc	155
8.123.2.3 umtsRscp	155
8.123.2.4 umtsUarfcn	155
8.124nas_umtsLTENbrCell Struct Reference	155
8.124.1 Detailed Description	155
8.124.2 Field Documentation	156
8.124.2.1 cellsTDD	156
8.124.2.2 earfcn	156
8.124.2.3 pci	156
8.124.2.4 rsrp	156
8.124.2.5 rsrq	156
8.124.2.6 srxlev	156
8.125nas_UniversalTime Struct Reference	156
8.125.1 Detailed Description	156
8.125.2 Field Documentation	157
8.125.2.1 day	157
8.125.2.2 dayOfWeek	157
8.125.2.3 hour	157
8.125.2.4 minute	157
8.125.2.5 month	157
8.125.2.6 second	157
8.125.2.7 year	157
8.126nas_wcdmaCellInfo Struct Reference	157
8.126.1 Detailed Description	157
8.126.2 Field Documentation	158
8.126.2.1 cpich_ecno	158
8.126.2.2 cpich_rscp	158
8.126.2.3 psc	158
8.126.2.4 srxlev	158
8.127nas_WCDMAECIOThresh Struct Reference	158
8.127.1 Detailed Description	158
8.127.2 Field Documentation	158
8.127.2.1 pWCDMAECIOThreshList	158
8.127.2.2 WCDMAECIOThreshListLen	159
8.128nas_WCDMAInfoLTENeighborCell Struct Reference	159
8.128.1 Detailed Description	159
8.128.2 Field Documentation	159
8.128.2.1 UMTSLTENbrCell	159

8.128.2.2 umtsLTENbrCellLen	159
8.128.2.3 wcdmaRRCState	159
8.129nas_WCDMARSSIThresh Struct Reference	159
8.129.1 Detailed Description	160
8.129.2 Field Documentation	160
8.129.2.1 pWCDMARSSIThreshList	160
8.129.2.2 WCDMARSSIThreshListLen	160
8.130nas_WCDMASysInfo Struct Reference	160
8.130.1 Detailed Description	160
8.130.2 Field Documentation	163
8.130.2.1 cellId	163
8.130.2.2 cellIdValid	163
8.130.2.3 hsCallStatus	163
8.130.2.4 hsCallStatusValid	163
8.130.2.5 hsInd	163
8.130.2.6 hsIndValid	163
8.130.2.7 lac	163
8.130.2.8 lacValid	163
8.130.2.9 MCC	163
8.130.2.10MNC	163
8.130.2.11networkIdValid	163
8.130.2.12psc	163
8.130.2.13pscValid	163
8.130.2.14regRejectInfoValid	163
8.130.2.15rejCause	163
8.130.2.16rejectSrvDomain	163
8.130.2.17sysInfoWCDMA	163
8.131NASBandPreferenceTlv Struct Reference	163
8.131.1 Field Documentation	163
8.131.1.1 band_pref	163
8.131.1.2 TlvPresent	164
8.132NASEmergencyModeTlv Struct Reference	164
8.132.1 Field Documentation	164
8.132.1.1 EmerMode	164
8.132.1.2 TlvPresent	164
8.133NasGetLTECphyCaInfo Struct Reference	164
8.133.1 Field Documentation	164
8.133.1.1 PhyCaAggPcellInfo	164
8.133.1.2 PhyCaAggScellIDBw	164
8.133.1.3 PhyCaAggScellIndex	164

8.133.1.4 PhyCaAggScellIndType	164
8.133.1.5 PhyCaAggScellInfo	164
8.134NASGWAcqOrderPrefTlv Struct Reference	164
8.134.1 Field Documentation	164
8.134.1.1 GWAcqOrderPref	164
8.134.1.2 TlvPresent	164
8.135NASLTEBandPreferenceTlv Struct Reference	165
8.135.1 Field Documentation	165
8.135.1.1 LTEBandPref	165
8.135.1.2 TlvPresent	165
8.136NASLteNasReleaseInfoTlv Struct Reference	165
8.136.1 Field Documentation	165
8.136.1.1 nas_major	165
8.136.1.2 nas_minor	165
8.136.1.3 nas_release	165
8.136.1.4 TlvPresent	165
8.137NASModePreferenceTlv Struct Reference	165
8.137.1 Field Documentation	165
8.137.1.1 ModePref	165
8.137.1.2 TlvPresent	165
8.138NASNetSelPreferenceTlv Struct Reference	165
8.138.1 Field Documentation	166
8.138.1.1 NetSelPref	166
8.138.1.2 TlvPresent	166
8.139NASOTAMessageTlv Struct Reference	166
8.139.1 Field Documentation	166
8.139.1.1 data_buf	166
8.139.1.2 data_len	166
8.139.1.3 message_type	166
8.139.1.4 TlvPresent	166
8.140NASPhyCaAggPcellInfo Struct Reference	166
8.140.1 Detailed Description	166
8.140.2 Field Documentation	167
8.140.2.1 dl_bw_value	167
8.140.2.2 freq	167
8.140.2.3 iLTEbandValue	167
8.140.2.4 pci	167
8.140.2.5 TlvPresent	167
8.141NASPhyCaAggScellIDBw Struct Reference	167
8.141.1 Detailed Description	167

8.141.2 Field Documentation	167
8.141.2.1 dl_bw_value	167
8.141.2.2 TlvPresent	167
8.142NASPhyCaAggScellIndex Struct Reference	167
8.142.1 Detailed Description	168
8.142.2 Field Documentation	168
8.142.2.1 scell_idx	168
8.142.2.2 TlvPresent	168
8.143NASPhyCaAggScellIndType Struct Reference	168
8.143.1 Detailed Description	168
8.143.2 Field Documentation	168
8.143.2.1 freq	169
8.143.2.2 pci	169
8.143.2.3 scell_state	169
8.143.2.4 TlvPresent	169
8.144NASPhyCaAggScellInfo Struct Reference	169
8.144.1 Detailed Description	169
8.144.2 Field Documentation	169
8.144.2.1 dl_bw_value	169
8.144.2.2 freq	170
8.144.2.3 iLTEbandValue	170
8.144.2.4 pci	170
8.144.2.5 scell_state	170
8.144.2.6 TlvPresent	170
8.145NASPRLPreferenceTlv Struct Reference	170
8.145.1 Field Documentation	170
8.145.1.1 PRLPref	170
8.145.1.2 TlvPresent	170
8.146NASQmiCbkNasSwiOTAMessageInd Struct Reference	170
8.146.1 Field Documentation	170
8.146.1.1 nasRelInfoTlv	170
8.146.1.2 otaMsgTlv	170
8.146.1.3 timeTlv	170
8.147NASQmiCbkNasSystemSelPrefInd Struct Reference	170
8.147.1 Field Documentation	171
8.147.1.1 BPTlv	171
8.147.1.2 EMTlv	171
8.147.1.3 GWAOPTlv	171
8.147.1.4 LBPTlv	171
8.147.1.5 MPTlv	171

8.147.1.6 NSPTlv	171
8.147.1.7 PRLPTlv	171
8.147.1.8 RPTlv	171
8.147.1.9 SDPTlv	171
8.148NASRoamPreferenceTlv Struct Reference	171
8.148.1 Field Documentation	171
8.148.1.1 RoamPref	171
8.148.1.2 TlvPresent	171
8.149NASServDomainPrefTlv Struct Reference	171
8.149.1 Field Documentation	171
8.149.1.1 SrvDomainPref	171
8.149.1.2 TlvPresent	171
8.150NASServingSystemInfo Struct Reference	171
8.150.1 Detailed Description	172
8.150.2 Field Documentation	173
8.150.2.1 csAttachState	173
8.150.2.2 hdrPersonality	173
8.150.2.3 psAttachState	173
8.150.2.4 radiolInterfaceList	173
8.150.2.5 radiolInterfaceNo	173
8.150.2.6 registrationState	173
8.150.2.7 selectedNetwork	173
8.151NASTimeInfoTlv Struct Reference	173
8.151.1 Field Documentation	173
8.151.1.1 time	173
8.151.1.2 TlvPresent	173
8.152newMTMessageTlv Struct Reference	173
8.152.1 Detailed Description	173
8.152.2 Field Documentation	173
8.152.2.1 MTMessageInfo	174
8.152.2.2 TlvPresent	174
8.153pack_dms_GetCustFeaturesV2_t Struct Reference	174
8.153.1 Detailed Description	174
8.153.2 Field Documentation	174
8.153.2.1 cust_id	174
8.153.2.2 list_type	174
8.153.2.3 Tlvresult	174
8.154pack_dms_SetCrashAction_t Struct Reference	174
8.154.1 Detailed Description	174
8.154.2 Field Documentation	175

8.154.2.1 crashAction	175
8.155pack_dms_SetCustFeature_t Struct Reference	175
8.155.1 Field Documentation	175
8.155.1.1 DHCPRelayEnabled	175
8.155.1.2 DisableIMSI	175
8.155.1.3 GpsEnable	175
8.155.1.4 GPSLPM	175
8.155.1.5 GPSSel	175
8.155.1.6 IPFamSupport	175
8.155.1.7 IsVoiceEnabled	175
8.155.1.8 RMAutoConnect	175
8.155.1.9 SMSSupport	175
8.156pack_dms_SetCustFeaturesV2_t Struct Reference	175
8.156.1 Detailed Description	176
8.156.2 Field Documentation	176
8.156.2.1 cust_id	176
8.156.2.2 cust_value	176
8.156.2.3 Tlvresult	176
8.156.2.4 value_length	176
8.157pack_dms_SetEventReport_t Struct Reference	176
8.157.1 Field Documentation	176
8.157.1.1 mode	176
8.158pack_dms_SetPower_t Struct Reference	176
8.158.1 Field Documentation	176
8.158.1.1 mode	177
8.158.1.2 Tlvresult	177
8.159pack_dms_SetUSBComp_t Struct Reference	177
8.159.1 Field Documentation	177
8.159.1.1 Tlvresult	177
8.159.1.2 USBComp	177
8.160pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	177
8.160.1 Detailed Description	177
8.160.2 Field Documentation	177
8.160.2.1 resetInfoInd	177
8.161pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	177
8.161.1 Detailed Description	177
8.161.2 Field Documentation	178
8.161.2.1 pDestSMSContent	178
8.161.2.2 pDestSMSNum	178
8.162pack_dms_UIMGetICCID_t Struct Reference	178

8.162.1 Detailed Description	178
8.162.2 Field Documentation	178
8.162.2.1 Tlvresult	178
8.163pack_fms_GetImagesPreference_t Struct Reference	178
8.163.1 Detailed Description	178
8.163.2 Field Documentation	178
8.163.2.1 Tlvresult	178
8.164pack_fms_GetStoredImages_t Struct Reference	179
8.164.1 Detailed Description	179
8.164.2 Field Documentation	179
8.164.2.1 Tlvresult	179
8.165pack_fms_SetImagesPreference_t Struct Reference	179
8.165.1 Detailed Description	179
8.165.2 Field Documentation	180
8.165.2.1 bForceDownload	180
8.165.2.2 imageListSize	180
8.165.2.3 modemindex	180
8.165.2.4 pImageList	180
8.165.2.5 Tlvresult	180
8.166pack_loc_Delete_Assist_Data_t Struct Reference	180
8.166.1 Detailed Description	180
8.166.2 Field Documentation	180
8.166.2.1 pBdsSVInfo	180
8.166.2.2 pCellDb	180
8.166.2.3 pClkInfo	181
8.166.2.4 pGnssData	181
8.166.2.5 pSVInfo	181
8.166.2.6 Tlvresult	181
8.167pack_loc_EventRegister_t Struct Reference	181
8.167.1 Detailed Description	181
8.167.2 Field Documentation	183
8.167.2.1 eventRegister	183
8.167.2.2 Tlvresult	183
8.168pack_loc_SetExtPowerState_t Struct Reference	183
8.168.1 Detailed Description	183
8.168.2 Field Documentation	183
8.168.2.1 extPowerState	183
8.168.2.2 Tlvresult	183
8.169pack_loc_SetOperationMode_t Struct Reference	183
8.169.1 Detailed Description	184

8.169.2 Field Documentation	184
8.169.2.1 mode	184
8.169.2.2 Tlvresult	184
8.170 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	184
8.170.1 Detailed Description	184
8.170.2 Field Documentation	184
8.170.2.1 Tlvresult	184
8.170.2.2 xid	184
8.171 pack_loc_Start_t Struct Reference	184
8.171.1 Detailed Description	185
8.171.2 Field Documentation	186
8.171.2.1 pApplicationInfo	186
8.171.2.2 pConfigAltitudeAssumed	186
8.171.2.3 pHorizontalAccuracyLvl	186
8.171.2.4 pIntermediateReportState	186
8.171.2.5 pMinIntervalTime	186
8.171.2.6 pRecurrenceType	186
8.171.2.7 SessionId	186
8.171.2.8 Tlvresult	186
8.172 pack_loc_Stop_t Struct Reference	186
8.172.1 Detailed Description	186
8.172.2 Field Documentation	186
8.172.2.1 SessionId	186
8.172.2.2 Tlvresult	186
8.173 pack_nas_SetACCOLC_t Struct Reference	186
8.173.1 Detailed Description	187
8.173.2 Field Documentation	187
8.173.2.1 accolc	187
8.173.2.2 spc	187
8.174 pack_nas_SetNetworkPreference_t Struct Reference	187
8.174.1 Detailed Description	187
8.174.2 Field Documentation	187
8.174.2.1 Duration	188
8.174.2.2 TechnologyPref	188
8.174.2.3 Tlvresult	188
8.175 pack_nas_SLQSGetPLMNName_t Struct Reference	188
8.175.1 Detailed Description	188
8.175.2 Field Documentation	188
8.175.2.1 mcc	188
8.175.2.2 mnc	188

8.175.2.3 pMncPcsStatus	188
8.176pack_nas_SLQSNInitiateNetworkRegistration_t Struct Reference	188
8.176.1 Detailed Description	189
8.176.2 Field Documentation	189
8.176.2.1 pChangeDuration	189
8.176.2.2 pMncPcsDigitStatus	189
8.176.2.3 pMNRInfo	189
8.176.2.4 regAction	189
8.177pack_nas_SLQSNasConfigSigInfo2_t Struct Reference	189
8.177.1 Detailed Description	190
8.177.2 Field Documentation	192
8.177.2.1 pCDMAECIODelta	193
8.177.2.2 pCDMAECIOThresh	193
8.177.2.3 pCDMARSSIDelta	193
8.177.2.4 pCDMARSSIThresh	193
8.177.2.5 pGSMRSSIDelta	193
8.177.2.6 pGSMRSSIThresh	193
8.177.2.7 pHDRECIODelta	193
8.177.2.8 pHDRECIOThresh	193
8.177.2.9 pHDRIODelta	193
8.177.2.10pHDRIOThresh	193
8.177.2.11pHDRRSSIDelta	193
8.177.2.12pHDRRSSIThresh	193
8.177.2.13pHDRSINRDelta	193
8.177.2.14pHDRSINRThresh	193
8.177.2.15pLTERSRPDelta	193
8.177.2.16pLTERSRPThresh	193
8.177.2.17pLTERSRQDelta	193
8.177.2.18pLTERSRQThresh	193
8.177.2.19pLTERSSIDelta	193
8.177.2.20pLTERSSIThresh	193
8.177.2.21pLTESigRptConfig	193
8.177.2.22pLTESNRDelta	193
8.177.2.23pLTESNRThresh	193
8.177.2.24pTDSCDMAECIODelta	193
8.177.2.25pTDSCDMAECIOThresh	193
8.177.2.26pTDSCDMARSCPDelta	193
8.177.2.27pTDSCDMARSCPThresh	193
8.177.2.28pTDSCDMARSSIDelta	193
8.177.2.29pTDSCDMARSSIThresh	194

8.177.2.30pTDSCDMASINRDelta	194
8.177.2.31pTDSCDMASINRThresh	194
8.177.2.32pWCDMAECIODelta	194
8.177.2.33pWCDMAECIOTresh	194
8.177.2.34pWCDMARSSIDelta	194
8.177.2.35pWCDMARSSITresh	194
8.178pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference	194
8.178.1 Detailed Description	194
8.178.2 Field Documentation	196
8.178.2.1 pDDTMInd	196
8.178.2.2 pDualStandByPrefInd	196
8.178.2.3 pErrorRateInd	196
8.178.2.4 pHDRNewUATIAssInd	196
8.178.2.5 pHDRSessionCloseInd	196
8.178.2.6 pLTECphyCa	196
8.178.2.7 pManagedRoamingInd	196
8.178.2.8 pNetworkTimeInd	196
8.178.2.9 pServingSystemInd	196
8.178.2.10pSignalStrengthInd	196
8.178.2.11pSubscriptionInfoInd	196
8.178.2.12pSysInfoInd	197
8.178.2.13pSystemSelectionInd	197
8.179pack_nas_SLQSNasSwiOTAMessageCallback_t Struct Reference	197
8.179.1 Detailed Description	197
8.179.2 Field Documentation	197
8.179.2.1 gsmUmtsDI	198
8.179.2.2 gsmUmtsUI	198
8.179.2.3 lteEmmDI	198
8.179.2.4 lteEmmUI	198
8.179.2.5 lteEsmDI	198
8.179.2.6 lteEsmUI	198
8.179.2.7 pRankIndicatorInd	198
8.180pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference	198
8.180.1 Detailed Description	198
8.180.2 Field Documentation	198
8.180.2.1 bEnable	198
8.180.2.2 pSigIndReq	198
8.181pack_nas_SLQSSetSysSelectionPref_t Struct Reference	198
8.181.1 Detailed Description	199
8.181.2 Field Documentation	202

8.181.2.1 pAcqOrderPref	202
8.181.2.2 pBandPref	202
8.181.2.3 pChgDuration	202
8.181.2.4 pCSGID	202
8.181.2.5 pEmerMode	202
8.181.2.6 pGWAcqOrderPref	202
8.181.2.7 pLTEBandPref	203
8.181.2.8 pMNCIncPCSDigStat	203
8.181.2.9 pModePref	203
8.181.2.10pNetSelPref	203
8.181.2.11pPRLPref	203
8.181.2.12pRAT	203
8.181.2.13pRoamPref	203
8.181.2.14pSrvDomainPref	203
8.181.2.15pSrvRegRestriction	203
8.181.2.16pTdsdmaBandPref	203
8.182pack_qmi_t Struct Reference	203
8.182.1 Detailed Description	203
8.182.2 Field Documentation	203
8.182.2.1 msgid	203
8.182.2.2 svc	203
8.182.2.3 timeout	203
8.182.2.4 xid	203
8.183pack_qos_SLQSQosSviReadApnExtraParams_t Struct Reference	203
8.183.1 Detailed Description	204
8.183.2 Field Documentation	204
8.183.2.1 apnId	204
8.184pack_qos_SLQSQosSviReadDataStats_t Struct Reference	204
8.184.1 Detailed Description	204
8.184.2 Field Documentation	204
8.184.2.1 apnId	204
8.185pack_qos_SLQSSetQosEventCallback_t Struct Reference	204
8.185.1 Detailed Description	204
8.185.2 Field Documentation	205
8.185.2.1 enable	205
8.186pack_sms_SendSMS_t Struct Reference	205
8.186.1 Detailed Description	205
8.186.2 Field Documentation	205
8.186.2.1 messageFormat	205
8.186.2.2 messageSize	205

8.186.2.3 pLinktimer	205
8.186.2.4 pMessage	205
8.187pack_sms_SetNewSMSCallback_t Struct Reference	205
8.187.1 Detailed Description	206
8.187.2 Field Documentation	206
8.187.2.1 status	206
8.188pack_sms_SLQSDeleteSMS_t Struct Reference	206
8.188.1 Detailed Description	206
8.188.2 Field Documentation	206
8.188.2.1 pMessageIndex	207
8.188.2.2 pMessageMode	207
8.188.2.3 pMessageTag	207
8.188.2.4 storageType	207
8.189pack_sms_SLQSGetSMS_t Struct Reference	207
8.189.1 Detailed Description	207
8.189.2 Field Documentation	207
8.189.2.1 messageIndex	207
8.189.2.2 pMessageMode	207
8.189.2.3 storageType	207
8.190pack_sms_SLQSGetSMSList_t Struct Reference	207
8.190.1 Detailed Description	207
8.190.2 Field Documentation	208
8.190.2.1 pMessageMode	208
8.190.2.2 pRequestedTag	208
8.190.2.3 storageType	208
8.191pack_sms_SLQSModifySMSStatus_t Struct Reference	208
8.191.1 Detailed Description	208
8.191.2 Field Documentation	209
8.191.2.1 messageIndex	209
8.191.2.2 messageTag	209
8.191.2.3 pMessageMode	209
8.191.2.4 storageType	209
8.192pack_swiloc_SwiLocSetAutoStart_t Struct Reference	209
8.192.1 Detailed Description	209
8.192.2 Field Documentation	210
8.192.2.1 fix_rate	210
8.192.2.2 fix_type	210
8.192.2.3 function	210
8.192.2.4 max_dist	210
8.192.2.5 max_time	210

8.192.2.6 set_fix_rate	210
8.192.2.7 set_fix_type	210
8.192.2.8 set_function	210
8.192.2.9 set_max_dist	210
8.192.2.10set_max_time	210
8.193pack_swoma_SLQSOMADMCancelSession_t Struct Reference	210
8.193.1 Detailed Description	211
8.193.2 Field Documentation	211
8.193.2.1 sessionType	211
8.194pack_swoma_SLQSOMADMGetSessionInfo_t Struct Reference	211
8.194.1 Detailed Description	211
8.194.2 Field Documentation	211
8.194.2.1 SessionType	211
8.195pack_swoma_SLQSOMADMSendSelection_t Struct Reference	211
8.195.1 Detailed Description	212
8.195.2 Field Documentation	212
8.195.2.1 pDeferTime	212
8.195.2.2 pRejectReason	212
8.195.2.3 selection	212
8.196pack_swoma_SLQSOMADMSetSettings_t Struct Reference	212
8.196.1 Detailed Description	212
8.196.2 Field Documentation	213
8.196.2.1 FOTAdownload	213
8.196.2.2 FOTAUpdate	213
8.196.2.3 pAutosdm	213
8.196.2.4 pFwAutoCheck	213
8.197pack_swoma_SLQSOMADMStartSession_t Struct Reference	213
8.197.1 Detailed Description	213
8.197.2 Field Documentation	213
8.197.2.1 sessionType	213
8.198pack_uim_ChangePin_t Struct Reference	214
8.198.1 Detailed Description	214
8.198.2 Field Documentation	214
8.198.2.1 changePIN	214
8.198.2.2 EncryptedPIN1	214
8.198.2.3 pIndicationToken	214
8.198.2.4 pKeyReferenceID	214
8.198.2.5 sessionInfo	214
8.198.2.6 Tlvresult	215
8.199pack_uim_ReadTransparent_t Struct Reference	215

8.199.1 Detailed Description	215
8.199.2 Field Documentation	215
8.199.2.1 fileIndex	215
8.199.2.2 pEncryptData	215
8.199.2.3 pIndicationToken	215
8.199.2.4 readTransparent	216
8.199.2.5 sessionInfo	216
8.199.2.6 Tlvresult	216
8.200pack_uim_SetPinProtection_t Struct Reference	216
8.200.1 Detailed Description	216
8.200.2 Field Documentation	216
8.200.2.1 EncryptedPIN1	216
8.200.2.2 pIndicationToken	216
8.200.2.3 pinProtection	217
8.200.2.4 pKeyReferenceID	217
8.200.2.5 sessionInfo	217
8.200.2.6 Tlvresult	217
8.201pack_uim_SLQSUIMEventRegister_t Struct Reference	217
8.201.1 Detailed Description	217
8.201.2 Field Documentation	217
8.201.2.1 eventMask	217
8.202pack_uim_SLQSUIMPowerDown_t Struct Reference	217
8.202.1 Detailed Description	217
8.202.2 Field Documentation	217
8.202.2.1 slot	217
8.203pack_uim_SLQSUIMPowerUp_t Struct Reference	217
8.203.1 Detailed Description	218
8.203.2 Field Documentation	218
8.203.2.1 plgnoreHotSwapSwitch	218
8.203.2.2 slot	218
8.204pack_uim_SLQSUIMSwitchSlot_t Struct Reference	218
8.204.1 Detailed Description	218
8.204.2 Field Documentation	219
8.204.2.1 bLogicalSlot	219
8.204.2.2 ulPhysicalSlot	219
8.205pack_uim_UnblockPin_t Struct Reference	219
8.205.1 Detailed Description	219
8.205.2 Field Documentation	220
8.205.2.1 EncryptedPIN1	220
8.205.2.2 pIndicationToken	220

8.205.2.3 pinProtection	220
8.205.2.4 pKeyReferenceID	220
8.205.2.5 sessionInfo	220
8.205.2.6 Tlvresult	220
8.206pack_uim_VerifyPin_t Struct Reference	220
8.206.1 Detailed Description	220
8.206.2 Field Documentation	221
8.206.2.1 pEncryptedPIN1	221
8.206.2.2 pIndicationToken	221
8.206.2.3 pKeyReferenceID	221
8.206.2.4 sessionInfo	221
8.206.2.5 Tlvresult	221
8.206.2.6 verifyPIN	221
8.207pack_wds_GetDefaultProfile_t Struct Reference	221
8.207.1 Detailed Description	221
8.207.2 Field Documentation	221
8.207.2.1 profiletype	221
8.208pack_wds_GetDefaultProfileNum_t Struct Reference	221
8.208.1 Detailed Description	222
8.208.2 Field Documentation	222
8.208.2.1 family	222
8.208.2.2 type	222
8.209pack_wds_GetDormancyState_t Struct Reference	222
8.210pack_wds_GetLastMobileIPError_t Struct Reference	222
8.211pack_wds_GetMobileIP_t Struct Reference	222
8.212pack_wds_GetMobileIPProfile_t Struct Reference	222
8.212.1 Detailed Description	222
8.212.2 Field Documentation	222
8.212.2.1 index	222
8.213pack_wds_GetPacketStatistics_t Struct Reference	222
8.213.1 Detailed Description	223
8.213.2 Field Documentation	223
8.213.2.1 pStatMask	223
8.214pack_wds_GetPacketStatus_t Struct Reference	223
8.214.1 Detailed Description	223
8.214.2 Field Documentation	223
8.214.2.1 statmask	223
8.215pack_wds_GetSessionDuration_t Struct Reference	223
8.216pack_wds_RMSetTransferStatistics_t Struct Reference	223
8.216.1 Detailed Description	223

8.216.2 Field Documentation	223
8.216.2.1 RmTrasferStaticsReq	224
8.217pack_wds_SetDefaultProfile_t Struct Reference	224
8.217.1 Detailed Description	224
8.217.2 Field Documentation	224
8.217.2.1 authentication	224
8.217.2.2 ipAddress	224
8.217.2.3 pApnname	224
8.217.2.4 pdpType	224
8.217.2.5 pName	224
8.217.2.6 pPassword	224
8.217.2.7 primaryDNS	224
8.217.2.8 profileType	224
8.217.2.9 pUsername	224
8.217.2.10secondaryDNS	224
8.218pack_wds_SetDefaultProfileNum_t Struct Reference	225
8.218.1 Field Documentation	225
8.218.1.1 family	225
8.218.1.2 index	225
8.218.1.3 type	225
8.219pack_wds_SetMobileIPProfile_t Struct Reference	225
8.219.1 Detailed Description	225
8.219.2 Field Documentation	225
8.219.2.1 index	226
8.219.2.2 pAAASPI	226
8.219.2.3 pAddress	226
8.219.2.4 pEnabled	226
8.219.2.5 pHASPI	226
8.219.2.6 pMNAHA	226
8.219.2.7 pMNHA	226
8.219.2.8 pNAI	226
8.219.2.9 pPrimaryHA	226
8.219.2.10pRevTunneling	226
8.219.2.11pSecondaryHA	226
8.219.2.12spc	226
8.220pack_wds_SLQSCreateProfile_t Struct Reference	226
8.220.1 Detailed Description	226
8.220.2 Field Documentation	227
8.220.2.1 pCurProfile	227
8.220.2.2 pProfileId	227

8.220.2.3 pProfileType	227
8.221pack_wds_SLQSDeleteProfile_t Struct Reference	227
8.221.1 Detailed Description	227
8.221.2 Field Documentation	227
8.221.2.1 profileIndex	227
8.221.2.2 profileType	227
8.222pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference	227
8.223pack_wds_SLQSGetDataBearerTechnology_t Struct Reference	227
8.224pack_wds_SLQSGetDUNCallInfo_t Struct Reference	227
8.224.1 Detailed Description	227
8.224.2 Field Documentation	228
8.224.2.1 Mask	228
8.224.2.2 pReportChannelRate	228
8.224.2.3 pReportConnStatus	228
8.224.2.4 pReportDataBearerTech	228
8.224.2.5 pReportDormStatus	228
8.224.2.6 pTransferStatInd	228
8.225pack_wds_SLQSGetProfileSettings_t Struct Reference	228
8.225.1 Detailed Description	228
8.225.2 Field Documentation	229
8.225.2.1 ProfileId	229
8.225.2.2 ProfileType	229
8.226pack_wds_SLQSGetRuntimeSettings_t Struct Reference	229
8.226.1 Detailed Description	229
8.226.2 Field Documentation	229
8.226.2.1 pReqSettings	229
8.227pack_wds_SLQSModifyProfile_t Struct Reference	229
8.227.1 Detailed Description	230
8.227.2 Field Documentation	230
8.227.2.1 curProfile	230
8.227.2.2 pProfileId	230
8.227.2.3 pProfileType	230
8.228pack_wds_SLQSSet3GPPConfigItem_t Struct Reference	230
8.228.1 Detailed Description	230
8.228.2 Field Documentation	231
8.228.2.1 LTEAttachProfileListLen	231
8.228.2.2 p3gppRelease	232
8.228.2.3 pDefaultPDNEnabled	232
8.228.2.4 pLTEAttachProfile	232
8.228.2.5 pLTEAttachProfileList	232

8.228.2.6 pProfileList	232
8.229pack_wds_SLQSSetIPFamilyPreference_t Struct Reference	232
8.229.1 Detailed Description	232
8.229.2 Field Documentation	232
8.229.2.1 IPFamilyPreference	232
8.230pack_wds_SLQSSetWdsEventCallback_t Struct Reference	232
8.230.1 Detailed Description	232
8.230.2 Field Documentation	233
8.230.2.1 currentDataBearer	233
8.230.2.2 dataBearer	233
8.230.2.3 dataSystemStatus	233
8.230.2.4 dormancyStatus	233
8.230.2.5 interval	233
8.230.2.6 mobileIP	233
8.230.2.7 transferStats	233
8.231pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference	233
8.231.1 Detailed Description	233
8.231.2 Field Documentation	233
8.231.2.1 pProfileId	233
8.232pack_wds_SLQSSetLoopback_t Struct Reference	233
8.232.1 Detailed Description	233
8.232.2 Field Documentation	234
8.232.2.1 loopbackMode	234
8.232.2.2 loopbackMultiplier	234
8.233pack_wds_SLQSStartDataSession_t Struct Reference	234
8.233.1 Detailed Description	234
8.233.2 Field Documentation	234
8.233.2.1 pAuth	235
8.233.2.2 pPass	235
8.233.2.3 pprofileid3gpp	235
8.233.2.4 pprofileid3gpp2	235
8.233.2.5 pTech	235
8.233.2.6 pUser	235
8.234pack_wds_SLQSStopDataSession_t Struct Reference	235
8.234.1 Detailed Description	235
8.234.2 Field Documentation	235
8.234.2.1 psid	235
8.235pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	235
8.235.1 Detailed Description	235
8.235.2 Field Documentation	235

8.235.2.1 contextId	235
8.235.2.2 contextType	235
8.236PackCreateProfileOut Struct Reference	235
8.236.1 Field Documentation	236
8.236.1.1 ExtErrorCode	236
8.236.1.2 ProfileIndex	236
8.236.1.3 ProfileType	236
8.237packgetDyingGaspCfg Struct Reference	236
8.237.1 Detailed Description	236
8.237.2 Field Documentation	236
8.237.2.1 pDestSMSContent	236
8.237.2.2 pDestSMSNum	236
8.238packgetDyingGaspStatistics Struct Reference	236
8.238.1 Detailed Description	236
8.238.2 Field Documentation	237
8.238.2.1 pSMSAttemptedFlag	237
8.238.2.2 pTimeStamp	237
8.239qmiSmsMessageList Struct Reference	237
8.239.1 Detailed Description	237
8.239.2 Field Documentation	237
8.239.2.1 messageIndex	237
8.239.2.2 messageTag	237
8.240qmiWSDDataBearerTechnology Struct Reference	237
8.240.1 Detailed Description	237
8.240.2 Field Documentation	238
8.240.2.1 currentNetwork	238
8.240.2.2 ratMask	238
8.240.2.3 soMask	238
8.241RFBandInfoElements Struct Reference	238
8.241.1 Detailed Description	238
8.241.2 Field Documentation	238
8.241.2.1 activeBandClass	238
8.241.2.2 activeChannel	238
8.241.2.3 radiolInterface	238
8.242rmTrasferStaticsReq Struct Reference	238
8.242.1 Detailed Description	238
8.242.2 Field Documentation	238
8.242.2.1 bResetStatistics	238
8.242.2.2 ulMask	238
8.243slot_t Struct Reference	239

8.243.1 Detailed Description	239
8.243.2 Field Documentation	239
8.243.2.1 bICCID	239
8.243.2.2 bICCIDLength	239
8.243.2.3 bLogicalSlot	239
8.243.2.4 uPhyCardStatus	239
8.243.2.5 uPhySlotStatus	240
8.244slotInf Struct Reference	240
8.244.1 Detailed Description	240
8.244.2 Field Documentation	241
8.244.2.1 AppStatus	241
8.244.2.2 cardState	241
8.244.2.3 errorState	241
8.244.2.4 numApp	241
8.244.2.5 upinRetries	241
8.244.2.6 upinState	241
8.244.2.7 upukRetries	241
8.245slots_t Struct Reference	241
8.245.1 Field Documentation	241
8.245.1.1 uimSlotStatus	241
8.246sMSCAddress Struct Reference	241
8.246.1 Detailed Description	242
8.246.2 Field Documentation	242
8.246.2.1 data	242
8.246.2.2 length	242
8.247sMSCAddressTlv Struct Reference	242
8.247.1 Detailed Description	242
8.247.2 Field Documentation	242
8.247.2.1 SMSCInfo	242
8.247.2.2 TlvPresent	242
8.248sMSEtwsMessage Struct Reference	242
8.248.1 Detailed Description	243
8.248.2 Field Documentation	243
8.248.2.1 data	243
8.248.2.2 length	243
8.248.2.3 notificationType	243
8.249sMSEtwsMessageTlv Struct Reference	243
8.249.1 Detailed Description	243
8.249.2 Field Documentation	243
8.249.2.1 EtwsMessageInfo	243

8.249.2.2 TlvPresent	243
8.250sMSEtwsPlmn Struct Reference	243
8.250.1 Detailed Description	244
8.250.2 Field Documentation	244
8.250.2.1 mobileCountryCode	244
8.250.2.2 mobileNetworkCode	244
8.251sMSMessageMode Struct Reference	244
8.251.1 Detailed Description	244
8.251.2 Field Documentation	244
8.251.2.1 messageMode	244
8.252sMSMTMessage Struct Reference	244
8.252.1 Detailed Description	244
8.252.2 Field Documentation	245
8.252.2.1 messageIndex	245
8.252.2.2 storageType	245
8.253sMSOnIMS Struct Reference	245
8.253.1 Detailed Description	245
8.253.2 Field Documentation	245
8.253.2.1 smsOnIMS	245
8.254sMSOnIMSTlv Struct Reference	245
8.254.1 Detailed Description	245
8.254.2 Field Documentation	245
8.254.2.1 IMSInfo	245
8.254.2.2 TlvPresent	245
8.255sMSTransferRouteMTMessage Struct Reference	245
8.255.1 Detailed Description	246
8.255.2 Field Documentation	246
8.255.2.1 ackIndicator	246
8.255.2.2 data	246
8.255.2.3 format	246
8.255.2.4 length	246
8.255.2.5 transactionID	246
8.256tdscdmaSigInfoExt Struct Reference	246
8.256.1 Detailed Description	246
8.256.2 Field Documentation	247
8.256.2.1 ecio	247
8.256.2.2 rscp	247
8.256.2.3 rssi	247
8.256.2.4 sinr	247
8.257transferRouteMessageTlv Struct Reference	247

8.257.1 Detailed Description	247
8.257.2 Field Documentation	247
8.257.2.1 TlvPresent	247
8.257.2.2 TransferRouteMTMessageInfo	247
8.258transferStatInd Struct Reference	247
8.258.1 Detailed Description	247
8.258.2 Field Documentation	248
8.258.2.1 StatsMask	248
8.258.2.2 StatsPeriod	248
8.259uim_appStatus Struct Reference	248
8.259.1 Detailed Description	248
8.259.2 Field Documentation	250
8.259.2.1 aidLength	250
8.259.2.2 aidVal	250
8.259.2.3 appState	250
8.259.2.4 appType	250
8.259.2.5 persoFeature	250
8.259.2.6 persoRetries	250
8.259.2.7 persoState	250
8.259.2.8 persoUnblockRetries	251
8.259.2.9 pin1Retries	251
8.259.2.10pin1State	251
8.259.2.11pin2Retries	251
8.259.2.12pin2State	251
8.259.2.13puk1Retries	251
8.259.2.14puk2Retries	251
8.259.2.15univPin	251
8.260uim_cardResult Struct Reference	251
8.260.1 Detailed Description	251
8.260.2 Field Documentation	251
8.260.2.1 sw1	251
8.260.2.2 sw2	251
8.261uim_cardStatus Struct Reference	251
8.261.1 Detailed Description	252
8.261.2 Field Documentation	252
8.261.2.1 index1xPri	252
8.261.2.2 index1xSec	252
8.261.2.3 indexGwPri	252
8.261.2.4 indexGwSec	252
8.261.2.5 numSlot	252

8.261.2.6 SlotInfo	252
8.262uim_changeUIMPIN Struct Reference	253
8.262.1 Detailed Description	253
8.262.2 Field Documentation	253
8.262.2.1 oldPINLen	253
8.262.2.2 oldPINVal	253
8.262.2.3 pinID	253
8.262.2.4 pinLen	253
8.262.2.5 pinVal	253
8.263uim_encryptedPIN1 Struct Reference	253
8.263.1 Detailed Description	254
8.263.2 Field Documentation	254
8.263.2.1 pin1Len	254
8.263.2.2 pin1Val	254
8.264uim_fileInfo Struct Reference	254
8.264.1 Detailed Description	254
8.264.2 Field Documentation	255
8.264.2.1 fileID	255
8.264.2.2 path	255
8.264.2.3 pathLen	255
8.265uim_hotSwapStatus Struct Reference	255
8.265.1 Detailed Description	255
8.265.2 Field Documentation	255
8.265.2.1 hotSwap	255
8.265.2.2 hotSwapLength	255
8.266uim_readResult Struct Reference	255
8.266.1 Detailed Description	255
8.266.2 Field Documentation	256
8.266.2.1 content	256
8.266.2.2 contentLen	256
8.267uim_readTransparentInfo Struct Reference	256
8.267.1 Detailed Description	256
8.267.2 Field Documentation	256
8.267.2.1 length	256
8.267.2.2 offset	256
8.268uim_remainingRetries Struct Reference	256
8.268.1 Detailed Description	257
8.268.2 Field Documentation	257
8.268.2.1 unblockLeft	257
8.268.2.2 verifyLeft	257

8.269uim_sessionInformation Struct Reference	257
8.269.1 Detailed Description	257
8.269.2 Field Documentation	258
8.269.2.1 aid	258
8.269.2.2 aidLength	258
8.269.2.3 sessionType	258
8.270uim_setPINProtection Struct Reference	258
8.270.1 Detailed Description	258
8.270.2 Field Documentation	259
8.270.2.1 pinID	259
8.270.2.2 pinLength	259
8.270.2.3 pinOperation	259
8.270.2.4 pinValue	259
8.271uim_slotInfo Struct Reference	259
8.271.1 Detailed Description	259
8.271.2 Field Documentation	260
8.271.2.1 AppStatus	260
8.271.2.2 cardState	260
8.271.2.3 errorState	260
8.271.2.4 numApp	260
8.271.2.5 upinRetries	260
8.271.2.6 upinState	260
8.271.2.7 upukRetries	260
8.272uim_UIMSessionInformation Struct Reference	260
8.272.1 Detailed Description	260
8.272.2 Field Documentation	261
8.272.2.1 aid	261
8.272.2.2 aidLength	261
8.272.2.3 sessionType	261
8.273uim_unblockUIMPIN Struct Reference	261
8.273.1 Detailed Description	261
8.273.2 Field Documentation	262
8.273.2.1 newPINLen	262
8.273.2.2 newPINVal	262
8.273.2.3 pinID	262
8.273.2.4 pukLen	262
8.273.2.5 pukVal	262
8.274uim_verifyUIMPIN Struct Reference	262
8.274.1 Detailed Description	262
8.274.2 Field Documentation	263

8.274.2.1 pinID	263
8.274.2.2 pinLen	263
8.274.2.3 pinVal	263
8.275unpack_dms_GetActivationState_t Struct Reference	263
8.275.1 Detailed Description	263
8.275.2 Field Documentation	263
8.275.2.1 state	263
8.276unpack_dms_GetBandCapability_t Struct Reference	263
8.276.1 Field Documentation	263
8.276.1.1 BandCapability	264
8.276.1.2 Tlvresult	264
8.277unpack_dms_GetCrashAction_t Struct Reference	264
8.277.1 Field Documentation	264
8.277.1.1 DevCrashState	264
8.277.1.2 Tlvresult	264
8.278unpack_dms_GetCustFeature_t Struct Reference	264
8.278.1 Field Documentation	264
8.278.1.1 DHCPRelayEnabled	264
8.278.1.2 DisableIMSI	264
8.278.1.3 GpsEnable	264
8.278.1.4 GPSLPM	264
8.278.1.5 GPSSel	264
8.278.1.6 IPFamSupport	264
8.278.1.7 IsVoiceEnabled	264
8.278.1.8 RMAutoConnect	264
8.278.1.9 SMSSupport	264
8.278.1.10Tlvresult	265
8.279unpack_dms_GetCustFeaturesV2_t Struct Reference	265
8.279.1 Detailed Description	265
8.279.2 Field Documentation	265
8.279.2.1 GetCustomFeatureV2	265
8.279.2.2 Tlvresult	265
8.280unpack_dms_GetDeviceCap_t Struct Reference	265
8.280.1 Field Documentation	265
8.280.1.1 DataServiceCapability	265
8.280.1.2 MaxRXChannelRate	265
8.280.1.3 MaxTXChannelRate	265
8.280.1.4 Radiofaces	265
8.280.1.5 RadiofacesSize	265
8.280.1.6 SimCapability	265

8.280.1.7 Tlvresult	265
8.281unpack_dms_GetDeviceCapabilities_t Struct Reference	266
8.281.1 Detailed Description	266
8.281.2 Field Documentation	266
8.281.2.1 dataServiceCaCapability	266
8.281.2.2 maxRxChannelRate	266
8.281.2.3 maxTxChannelRate	266
8.281.2.4 Radiolfaces	266
8.281.2.5 radiolfacesSize	266
8.281.2.6 simCapability	266
8.282unpack_dms_GetDeviceHardwareRev_t Struct Reference	266
8.282.1 Field Documentation	266
8.282.1.1 String	266
8.282.1.2 stringSize	266
8.282.1.3 Tlvresult	267
8.283unpack_dms_GetDeviceMfr_t Struct Reference	267
8.283.1 Field Documentation	267
8.283.1.1 String	267
8.283.1.2 stringSize	267
8.283.1.3 Tlvresult	267
8.284unpack_dms_GetDeviceSerialNumbers_t Struct Reference	267
8.284.1 Field Documentation	267
8.284.1.1 esnSize	267
8.284.1.2 ESNString	267
8.284.1.3 imeiSize	267
8.284.1.4 IMEIString	267
8.284.1.5 imeiSvnSize	267
8.284.1.6 lmeiSvnString	267
8.284.1.7 meidSize	267
8.284.1.8 MEIDString	267
8.284.1.9 Tlvresult	267
8.285unpack_dms_GetFirmwareInfo_t Struct Reference	268
8.285.1 Detailed Description	268
8.285.2 Field Documentation	268
8.285.2.1 appversion_str	268
8.285.2.2 bootversion_str	268
8.285.2.3 carrier_str	268
8.285.2.4 cur_carr_name	268
8.285.2.5 cur_carr_rev	268
8.285.2.6 modelid_str	268

8.285.2.7 packageid_str	268
8.285.2.8 prversion_str	268
8.285.2.9 sku_str	268
8.285.2.10Tlvresult	269
8.286unpack_dms_GetFirmwareRevision_t Struct Reference	269
8.286.1 Field Documentation	269
8.286.1.1 amssSize	269
8.286.1.2 AMSSString	269
8.286.1.3 PRIString	269
8.286.1.4 Tlvresult	269
8.287unpack_dms_GetFirmwareRevisions_t Struct Reference	269
8.287.1 Detailed Description	269
8.287.2 Field Documentation	269
8.287.2.1 amssSize	269
8.287.2.2 AMSSString	269
8.287.2.3 bootSize	269
8.287.2.4 BootString	269
8.287.2.5 priSize	269
8.287.2.6 PRIString	270
8.287.2.7 Tlvresult	270
8.288unpack_dms_GetFSN_t Struct Reference	270
8.288.1 Field Documentation	270
8.288.1.1 String	270
8.288.1.2 Tlvresult	270
8.289unpack_dms_GetHardwareRevision_t Struct Reference	270
8.289.1 Detailed Description	270
8.289.2 Field Documentation	270
8.289.2.1 hwVer	270
8.290unpack_dms_GetIMSI_t Struct Reference	270
8.290.1 Field Documentation	270
8.290.1.1 imsi	270
8.290.1.2 Tlvresult	270
8.291unpack_dms_GetManufacturer_t Struct Reference	270
8.291.1 Detailed Description	271
8.291.2 Field Documentation	271
8.291.2.1 manufacturer	271
8.291.2.2 Tlvresult	271
8.292unpack_dms_GetModelID_t Struct Reference	271
8.292.1 Detailed Description	271
8.292.2 Field Documentation	271

8.292.2.1 modelid	271
8.292.2.2 Tlvresult	271
8.293unpack_dms_GetNetworkTime_t Struct Reference	271
8.293.1 Detailed Description	271
8.293.2 Field Documentation	272
8.293.2.1 source	272
8.293.2.2 timestamp	272
8.293.2.3 Tlvresult	272
8.294unpack_dms_GetOfflineReason_t Struct Reference	272
8.294.1 Detailed Description	272
8.294.2 Field Documentation	273
8.294.2.1 pbPlatform	273
8.294.2.2 pReasonMask	273
8.294.2.3 Tlvresult	273
8.295unpack_dms_GetPower_t Struct Reference	273
8.295.1 Detailed Description	273
8.295.2 Field Documentation	273
8.295.2.1 HardwareControlledMode	273
8.295.2.2 OfflineReason	273
8.295.2.3 OperationMode	273
8.295.2.4 Tlvresult	273
8.296unpack_dms_GetPRLVersion_t Struct Reference	273
8.296.1 Field Documentation	274
8.296.1.1 Tlvresult	274
8.296.1.2 u16PRLVersion	274
8.296.1.3 u8PRLPreference	274
8.297unpack_dms_GetSerialNumbers_t Struct Reference	274
8.297.1 Detailed Description	274
8.297.2 Field Documentation	274
8.297.2.1 esn	274
8.297.2.2 imei_no	274
8.297.2.3 imeisv_svn	274
8.297.2.4 meid	274
8.298unpack_dms_GetUSBComp_t Struct Reference	274
8.298.1 Field Documentation	274
8.298.1.1 NumSupUSBComps	274
8.298.1.2 SupUSBComps	274
8.298.1.3 Tlvresult	275
8.298.1.4 USBComp	275
8.299unpack_dms_GetVoiceNumber_t Struct Reference	275

8.299.1 Field Documentation	275
8.299.1.1 MIN	275
8.299.1.2 minSize	275
8.299.1.3 Tlvresult	275
8.299.1.4 VoiceNumber	275
8.299.1.5 voiceNumberSize	275
8.300unpack_dms_SetCrashAction_t Struct Reference	275
8.300.1 Detailed Description	275
8.300.2 Field Documentation	275
8.300.2.1 notused	275
8.301unpack_dms_SetCustFeature_t Struct Reference	275
8.301.1 Field Documentation	275
8.301.1.1 Tlvresult	275
8.302unpack_dms_SetCustFeaturesV2_t Struct Reference	276
8.302.1 Detailed Description	276
8.302.2 Field Documentation	276
8.302.2.1 Tlvresult	276
8.303unpack_dms_SetEventReport_ind_t Struct Reference	276
8.303.1 Detailed Description	276
8.303.2 Field Documentation	276
8.303.2.1 ActivationStatusTlv	276
8.303.2.2 OperatingModeTlv	276
8.303.2.3 Tlvresult	276
8.304unpack_dms_SetEventReport_t Struct Reference	277
8.304.1 Field Documentation	277
8.304.1.1 Tlvresult	277
8.305unpack_dms_SetFirmwarePreference_t Struct Reference	277
8.305.1 Field Documentation	277
8.305.1.1 Tlvresult	277
8.306unpack_dms_SetPower_t Struct Reference	277
8.306.1 Field Documentation	277
8.306.1.1 Tlvresult	277
8.307unpack_dms_SetUSBComp_t Struct Reference	277
8.307.1 Field Documentation	277
8.307.1.1 Tlvresult	277
8.308unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference	277
8.308.1 Detailed Description	278
8.308.2 Field Documentation	278
8.308.2.1 source	278
8.308.2.2 Tlvresult	278

8.308.2.3 type	278
8.309unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference	278
8.309.1 Detailed Description	278
8.309.2 Field Documentation	279
8.309.2.1 source	279
8.309.2.2 Tlvresult	279
8.309.2.3 type	279
8.310unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	279
8.310.1 Detailed Description	279
8.310.2 Field Documentation	280
8.310.2.1 Tlvresult	280
8.311unpack_dms_SLQSGetBandCapability_t Struct Reference	280
8.311.1 Detailed Description	280
8.311.2 Field Documentation	283
8.311.2.1 bandCapability	283
8.311.2.2 is_LteBandCapability_Available	283
8.311.2.3 is_TdsBandCapability_Available	283
8.311.2.4 LteBandCapability	283
8.311.2.5 TdsBandCapability	283
8.312unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference	283
8.312.1 Detailed Description	283
8.312.2 Field Documentation	283
8.312.2.1 Tlvresult	283
8.313unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference	283
8.313.1 Detailed Description	284
8.313.2 Field Documentation	284
8.313.2.1 pGetDyingGaspCfg	284
8.313.2.2 Tlvresult	284
8.314unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference	284
8.314.1 Detailed Description	284
8.314.2 Field Documentation	284
8.314.2.1 pGetDyingGaspStatistics	284
8.314.2.2 Tlvresult	284
8.315unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference	284
8.315.1 Detailed Description	284
8.315.2 Field Documentation	285
8.315.2.1 carrier	285
8.315.2.2 fwvers	285
8.315.2.3 numEntries	285
8.315.2.4 pCurrImgInfo	285

8.315.2.5 pkgver	285
8.315.2.6 priver	285
8.316unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference	285
8.316.1 Detailed Description	285
8.316.2 Field Documentation	286
8.316.2.1 imgType	286
8.316.2.2 logString	286
8.316.2.3 refData	286
8.316.2.4 refString	286
8.316.2.5 ResCode	286
8.316.2.6 Tlvresult	287
8.317unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	287
8.317.1 Detailed Description	287
8.317.2 Field Documentation	287
8.317.2.1 Tlvresult	287
8.318unpack_dms_UIMGetICCID_t Struct Reference	287
8.318.1 Detailed Description	287
8.318.2 Field Documentation	287
8.318.2.1 String	287
8.318.2.2 stringSize	287
8.318.2.3 Tlvresult	288
8.319unpack_fms_GetImagesPreference_t Struct Reference	288
8.319.1 Detailed Description	288
8.319.2 Field Documentation	288
8.319.2.1 ImageListSize	288
8.319.2.2 plmImageList	288
8.319.2.3 Tlvresult	288
8.320unpack_fms_GetStoredImages_t Struct Reference	288
8.320.1 Detailed Description	288
8.320.2 Field Documentation	289
8.320.2.1 imageList	289
8.320.2.2 imagelistSize	289
8.320.2.3 Tlvresult	289
8.321unpack_fms_SetImagesPreference_t Struct Reference	289
8.321.1 Detailed Description	289
8.321.2 Field Documentation	289
8.321.2.1 ImageTypes	289
8.321.2.2 ImageTypesSize	289
8.321.2.3 Tlvresult	289
8.322unpack_loc_BestAvailPos_Ind_t Struct Reference	289

8.322.1 Detailed Description	290
8.322.2 Field Documentation	294
8.322.2.1 pAltitudeWrtEllipsoid	294
8.322.2.2 pAltitudeWrtMeanSeaLevel	294
8.322.2.3 pGpsTime	294
8.322.2.4 pHeading	294
8.322.2.5 pHeadingUnc	294
8.322.2.6 pHorCirConf	294
8.322.2.7 pHorEllpConf	294
8.322.2.8 pHorReliability	294
8.322.2.9 pHorUncCircular	294
8.322.2.10 pHorUncEllipseOrientAzimuth	294
8.322.2.11 pHorUncEllipseSemiMajor	294
8.322.2.12 pHorUncEllipseSemiMinor	295
8.322.2.13 pLatitude	295
8.322.2.14 pLongitude	295
8.322.2.15 pMagneticDeviation	295
8.322.2.16 pPrecisionDilution	295
8.322.2.17 pSensorDataUsage	295
8.322.2.18 pSpeedHorizontal	295
8.322.2.19 pSpeedUnc	295
8.322.2.20 pSpeedVertical	295
8.322.2.21 pSpeedVerticalUnc	295
8.322.2.22 pSvUsedforFix	295
8.322.2.23 pTechnologyMask	295
8.322.2.24 pTimeSrc	295
8.322.2.25 pTimestampUtc	295
8.322.2.26 pTimeUnc	295
8.322.2.27 pVertConfidence	295
8.322.2.28 pVertReliability	295
8.322.2.29 pVertUnc	295
8.322.2.30 pXid	295
8.322.2.31 status	295
8.322.2.32 Tlvresult	295
8.323 unpack_loc_Delete_Assist_Data_t Struct Reference	295
8.323.1 Detailed Description	295
8.323.2 Field Documentation	296
8.323.2.1 Tlvresult	296
8.324 unpack_loc_EngineState_Ind_t Struct Reference	296
8.324.1 Detailed Description	296

8.324.2 Field Documentation	296
8.324.2.1 engineState	296
8.324.2.2 Tlvresult	296
8.325unpack_loc_EventRegister_t Struct Reference	296
8.325.1 Detailed Description	296
8.325.2 Field Documentation	297
8.325.2.1 Tlvresult	297
8.326unpack_loc_PositionRpt_Ind_t Struct Reference	297
8.326.1 Detailed Description	297
8.326.2 Field Documentation	301
8.326.2.1 pAltitudeAssumed	301
8.326.2.2 pAltitudeWrtEllipsoid	301
8.326.2.3 pAltitudeWrtMeanSeaLevel	301
8.326.2.4 pFixId	301
8.326.2.5 pGpsTime	301
8.326.2.6 pHeading	302
8.326.2.7 pHeadingUnc	302
8.326.2.8 pHorConfidence	302
8.326.2.9 pHorReliability	302
8.326.2.10pHorUncCircular	302
8.326.2.11pHorUncEllipseOrientAzimuth	302
8.326.2.12pHorUncEllipseSemiMajor	302
8.326.2.13pHorUncEllipseSemiMinor	302
8.326.2.14pLatitude	302
8.326.2.15pLeapSeconds	302
8.326.2.16pLongitude	302
8.326.2.17pMagneticDeviation	302
8.326.2.18pPrecisionDilution	302
8.326.2.19pSensorDataUsage	302
8.326.2.20pSpeedHorizontal	302
8.326.2.21pSpeedUnc	302
8.326.2.22pSpeedVertical	302
8.326.2.23pSvUsedforFix	302
8.326.2.24pTechnologyMask	302
8.326.2.25pTimeSrc	302
8.326.2.26pTimestampUtc	302
8.326.2.27pTimeUnc	302
8.326.2.28pVertConfidence	302
8.326.2.29pVertReliability	302
8.326.2.30pVertUnc	302

8.326.2.31sessionId	302
8.326.2.32sessionStatus	302
8.326.2.33Tlvresult	302
8.327unpack_loc_SetExtPowerConfig_Ind_t Struct Reference	303
8.327.1 Detailed Description	303
8.327.2 Field Documentation	303
8.327.2.1 status	303
8.327.2.2 Tlvresult	303
8.328unpack_loc_SetExtPowerState_t Struct Reference	303
8.328.1 Detailed Description	303
8.328.2 Field Documentation	304
8.328.2.1 Tlvresult	304
8.329unpack_loc_SetOperationMode_t Struct Reference	304
8.329.1 Detailed Description	304
8.329.2 Field Documentation	304
8.329.2.1 Tlvresult	304
8.330unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	304
8.330.1 Detailed Description	304
8.330.2 Field Documentation	304
8.330.2.1 Tlvresult	304
8.331unpack_loc_Start_t Struct Reference	304
8.331.1 Detailed Description	305
8.331.2 Field Documentation	305
8.331.2.1 Tlvresult	305
8.332unpack_loc_Stop_t Struct Reference	305
8.332.1 Detailed Description	305
8.332.2 Field Documentation	305
8.332.2.1 Tlvresult	305
8.333unpack_nas_GetCDMANetworkParameters_t Struct Reference	305
8.333.1 Detailed Description	305
8.333.2 Field Documentation	306
8.333.2.1 Application	306
8.333.2.2 Broadcast	306
8.333.2.3 CustomSCP	306
8.333.2.4 ForceRev0	306
8.333.2.5 Protocol	306
8.333.2.6 RegForeignNID	306
8.333.2.7 RegForeignSID	306
8.333.2.8 RegHomeSID	306
8.333.2.9 Roaming	306

8.333.2.10SCI	306
8.333.2.11SCM	306
8.334unpack_nas_GetHomeNetwork_t Struct Reference	306
8.334.1 Detailed Description	306
8.334.2 Field Documentation	307
8.334.2.1 mcc	307
8.334.2.2 mnc	307
8.334.2.3 name	307
8.334.2.4 nid	307
8.334.2.5 sid	307
8.335unpack_nas_GetNetworkPreference_t Struct Reference	307
8.335.1 Detailed Description	307
8.335.2 Field Documentation	308
8.335.2.1 ActiveTechPref	308
8.335.2.2 Duration	308
8.335.2.3 PersistentTechPref	308
8.335.2.4 Tlvresult	308
8.336unpack_nas_GetRFInfo_t Struct Reference	308
8.336.1 Detailed Description	308
8.336.2 Field Documentation	308
8.336.2.1 instancesSize	308
8.336.2.2 RFBandInfoElements	308
8.337unpack_nas_GetServingNetwork_t Struct Reference	308
8.337.1 Detailed Description	309
8.337.2 Field Documentation	309
8.337.2.1 CSDomain	309
8.337.2.2 DataCaps	309
8.337.2.3 DataCapsLen	309
8.337.2.4 MCC	309
8.337.2.5 MNC	309
8.337.2.6 Name	309
8.337.2.7 nameSize	309
8.337.2.8 PSDomain	309
8.337.2.9 Radiofaces	309
8.337.2.10RadiofacesSize	309
8.337.2.11RAN	309
8.337.2.12RegistrationState	309
8.337.2.13Roaming	309
8.338unpack_nas_GetServingNetworkCapabilities_t Struct Reference	309
8.338.1 Detailed Description	310

8.338.2 Field Documentation	310
8.338.2.1 DataCaps	310
8.338.2.2 DataCapsLen	310
8.339unpack_nas_GetSignalStrengths_t Struct Reference	310
8.339.1 Detailed Description	310
8.339.2 Field Documentation	310
8.339.2.1 len	310
8.339.2.2 radio	310
8.339.2.3 rssi	310
8.340unpack_nas_PerformNetworkScan_t Struct Reference	310
8.340.1 Detailed Description	310
8.340.2 Field Documentation	311
8.340.2.1 p3GppNetworkInfoInstances	311
8.340.2.2 p3GppNetworkInstanceSize	311
8.340.2.3 pPCSInstance	311
8.340.2.4 pPCSInstanceSize	311
8.340.2.5 pRATInstance	311
8.340.2.6 pRATInstanceSize	311
8.340.2.7 pScanResult	311
8.341unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	311
8.341.1 Detailed Description	311
8.341.2 Field Documentation	311
8.341.2.1 dataCaps	311
8.341.2.2 dataCapsSize	311
8.342unpack_nas_SetEventReportInd_t Struct Reference	311
8.342.1 Detailed Description	311
8.342.2 Field Documentation	312
8.342.2.1 RFTIv	312
8.342.2.2 RRTIv	312
8.342.2.3 SLQSSSTIv	312
8.342.2.4 SSTIv	312
8.343unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference	312
8.343.1 Detailed Description	312
8.343.2 Field Documentation	312
8.343.2.1 sPhyCaAggPcellInfo	312
8.343.2.2 sPhyCaAggScellIDBw	313
8.343.2.3 sPhyCaAggScellIndex	313
8.343.2.4 sPhyCaAggScellIndType	313
8.343.2.5 sPhyCaAggScellInfo	313
8.344unpack_nas_SetNetworkPreference_t Struct Reference	313

8.344.1 Detailed Description	313
8.344.2 Field Documentation	313
8.344.2.1 Tlvresult	313
8.345unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference	314
8.345.1 Detailed Description	314
8.345.2 Field Documentation	314
8.345.2.1 roaming	314
8.346unpack_nas_SetServingSystemCallback_ind_t Struct Reference	314
8.346.1 Detailed Description	314
8.346.2 Field Documentation	314
8.346.2.1 SSInfo	314
8.346.2.2 Tlvresult	314
8.347unpack_nas_SLqsGetLTECphyCAInfo_t Struct Reference	314
8.347.1 Detailed Description	315
8.347.2 Field Documentation	315
8.347.2.1 LTECphyCAInfo	315
8.347.2.2 Tlvresult	315
8.348unpack_nas_SLQSGetNetworkTime_t Struct Reference	315
8.348.1 Detailed Description	315
8.348.2 Field Documentation	315
8.348.2.1 p3GPP2TimeInfo	315
8.348.2.2 p3GPPTimeInfo	315
8.349unpack_nas_SLQSGetPLMNName_t Struct Reference	315
8.349.1 Field Documentation	316
8.349.1.1 longName	316
8.349.1.2 longNameCI	316
8.349.1.3 longNameEn	316
8.349.1.4 longNameLen	316
8.349.1.5 longNameSB	316
8.349.1.6 shortName	316
8.349.1.7 shortNameCI	316
8.349.1.8 shortNameEn	316
8.349.1.9 shortNameLen	316
8.349.1.10shortNameSB	316
8.349.1.11spn	316
8.349.1.12spnEncoding	316
8.349.1.13spnLength	316
8.350unpack_nas_SLQSGetServingSystem_t Struct Reference	316
8.350.1 Detailed Description	317
8.350.2 Field Documentation	317

8.350.2.1 BasestationID	318
8.350.2.2 BasestationLatitude	318
8.350.2.3 BasestationLongitude	318
8.350.2.4 CallBarStatus	318
8.350.2.5 CDMA_P_Rev	318
8.350.2.6 CDMASystemInfoExt	318
8.350.2.7 CellID	318
8.350.2.8 ConcSvcInfo	318
8.350.2.9 CurrentPLMN	318
8.350.2.10DataSrvCapabilities	318
8.350.2.11DefaultRoamInd	318
8.350.2.12DetailedSvcInfo	318
8.350.2.13DTMInd	318
8.350.2.14Gpp2TimeZone	318
8.350.2.15GppNetworkDSTAdjustment	318
8.350.2.16GppTimeZone	318
8.350.2.17HdrPersonality	318
8.350.2.18Lac	318
8.350.2.19NetworkID	318
8.350.2.20PRLInd	318
8.350.2.21RoamIndicatorVal	318
8.350.2.22RoamingIndicatorList	318
8.350.2.23ServingSystem	318
8.350.2.24SystemID	318
8.350.2.25TrackAreaCode	318
8.351unpack_nas_SLQSGetSignalStrength_t Struct Reference	318
8.351.1 Detailed Description	319
8.351.2 Field Documentation	319
8.351.2.1 ecioList	319
8.351.2.2 ecioListLen	319
8.351.2.3 errorRateList	319
8.351.2.4 errorRateListLen	319
8.351.2.5 lo	319
8.351.2.6 ltersrp	319
8.351.2.7 ltesnr	319
8.351.2.8 rsrqInfo	319
8.351.2.9 rxSignalStrengthList	319
8.351.2.10rxSignalStrengthListLen	319
8.351.2.11signalStrengthReqMask	319
8.351.2.12sinr	319

8.352unpack_nas_SLQSGetSysInfo_t Struct Reference	320
8.352.1 Detailed Description	320
8.352.2 Field Documentation	322
8.352.2.1 pAddCDMASysInfo	322
8.352.2.2 pAddGSMSysInfo	322
8.352.2.3 pAddHDRSysInfo	322
8.352.2.4 pAddLTESysInfo	322
8.352.2.5 pAddWCDMASysInfo	322
8.352.2.6 pCDMASrvStatusInfo	322
8.352.2.7 pCDMASysInfo	322
8.352.2.8 pGSMCallBarringSysInfo	322
8.352.2.9 pGSMCipherDomainSysInfo	322
8.352.2.10pGSMSrvStatusInfo	322
8.352.2.11pGSMSysInfo	322
8.352.2.12pHDRSrvStatusInfo	322
8.352.2.13pHDRSysInfo	322
8.352.2.14pLTESrvStatusInfo	322
8.352.2.15pLTESysInfo	322
8.352.2.16pLTEVoiceSupportSysInfo	322
8.352.2.17pWCDMACallBarringSysInfo	322
8.352.2.18pWCDMACipherDomainSysInfo	322
8.352.2.19pWCDMASrvStatusInfo	322
8.352.2.20pWCDMASysInfo	322
8.353unpack_nas_SLQSGetSysSelectionPref_t Struct Reference	322
8.353.1 Detailed Description	323
8.353.2 Field Documentation	326
8.353.2.1 pBandPref	326
8.353.2.2 pEmerMode	326
8.353.2.3 pGWAcqOrderPref	326
8.353.2.4 pLTEBandPref	326
8.353.2.5 pModePref	326
8.353.2.6 pNetSelPref	326
8.353.2.7 pPRLPref	326
8.353.2.8 pRoamPref	326
8.353.2.9 pSrvDomainPref	326
8.354unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference	326
8.354.1 Detailed Description	327
8.354.2 Field Documentation	327
8.354.2.1 pCDMAInfo	327
8.354.2.2 pGERANInfo	327

8.354.2.3 pLTEInfoInterfreq	327
8.354.2.4 pLTEInfoIntrafreq	327
8.354.2.5 pLTEInfoNeighboringGSM	327
8.354.2.6 pLTEInfoNeighboringWCDMA	327
8.354.2.7 pUMTSCellID	328
8.354.2.8 pUMTSInfo	328
8.354.2.9 pWCDMAInfoLTENeighborCell	328
8.355unpack_nas_SLQSNasGetSigInfo_t Struct Reference	328
8.355.1 Detailed Description	328
8.355.2 Field Documentation	328
8.355.2.1 CDMASigInfo	328
8.355.2.2 GSMSSInfo	328
8.355.2.3 HDRSSInfo	328
8.355.2.4 LTESInfo	328
8.355.2.5 WCDMAInfo	328
8.356unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference	328
8.356.1 Detailed Description	328
8.356.2 Field Documentation	329
8.356.2.1 pDayltSavAdj	329
8.356.2.2 pRadioInterface	329
8.356.2.3 pTimeZone	329
8.356.2.4 universalTime	329
8.357unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference	329
8.357.1 Detailed Description	329
8.357.2 Field Documentation	330
8.357.2.1 pCDMASigInfo	330
8.357.2.2 pGSMSigInfo	330
8.357.2.3 pHDRSigInfo	330
8.357.2.4 pLTESigInfo	330
8.357.2.5 pRscp	330
8.357.2.6 pTDSCDMASigInfoExt	330
8.357.2.7 pWCDMASigInfo	330
8.358unpack_nas_SLQSNasSwiModemStatus_t Struct Reference	330
8.358.1 Detailed Description	330
8.358.2 Field Documentation	330
8.358.2.1 commonInfo	330
8.358.2.2 pLTEInfo	330
8.359unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference	331
8.359.1 Detailed Description	331
8.359.2 Field Documentation	331

8.359.2.1 Info	331
8.359.2.2 Tlvresult	331
8.360unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference	331
8.360.1 Detailed Description	331
8.360.2 Field Documentation	331
8.360.2.1 Info	331
8.360.2.2 Tlvresult	331
8.361unpack_nas_SLQSSwiGetLteCQI_t Struct Reference	331
8.361.1 Detailed Description	332
8.361.2 Field Documentation	332
8.361.2.1 CQIValueCW0	332
8.361.2.2 CQIValueCW1	332
8.361.2.3 ValidityCW0	332
8.361.2.4 ValidityCW1	332
8.362unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference	332
8.362.1 Detailed Description	332
8.362.2 Field Documentation	333
8.362.2.1 pSccRxInfo	333
8.363unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference	333
8.363.1 Detailed Description	333
8.363.2 Field Documentation	335
8.363.2.1 pAddCDMASysInfo	335
8.363.2.2 pAddGSMSysInfo	335
8.363.2.3 pAddHDRSysInfo	335
8.363.2.4 pAddLTESysInfo	335
8.363.2.5 pAddWCDMASysInfo	335
8.363.2.6 pCDMASrvStatusInfo	335
8.363.2.7 pCDMASysInfo	335
8.363.2.8 pGSMCallBarringSysInfo	335
8.363.2.9 pGSMCipherDomainSysInfo	335
8.363.2.10pGSMSrvStatusInfo	335
8.363.2.11pGSMSysInfo	335
8.363.2.12pHDRSrvStatusInfo	335
8.363.2.13pHDRSysInfo	335
8.363.2.14pLTESrvStatusInfo	335
8.363.2.15pLTESysInfo	335
8.363.2.16pLTEVoiceSupportSysInfo	335
8.363.2.17pSysInfoNoChange	335
8.363.2.18pWCDMACallBarringSysInfo	335
8.363.2.19pWCDMACipherDomainSysInfo	335

8.363.2.20pWCDMASrvStatusInfo	335
8.363.2.21pWCDMASysInfo	335
8.364unpack_omaDmConfigTlv_t Struct Reference	335
8.364.1 Detailed Description	336
8.364.2 Field Documentation	336
8.364.2.1 alertmsg	336
8.364.2.2 alertmsglength	336
8.364.2.3 state	336
8.364.2.4 userInputReq	336
8.364.2.5 userInputTimeout	336
8.365unpack_omaDmFotaTlv_t Struct Reference	336
8.365.1 Detailed Description	337
8.365.2 Field Documentation	338
8.365.2.1 description	338
8.365.2.2 descriptionlength	338
8.365.2.3 fwdloadsize	338
8.365.2.4 fwloadComplete	338
8.365.2.5 namelength	338
8.365.2.6 package_name	338
8.365.2.7 sessionType	338
8.365.2.8 severity	338
8.365.2.9 state	338
8.365.2.10updateCompleteStatus	338
8.365.2.11userInputReq	338
8.365.2.12userInputTimeout	338
8.365.2.13version	338
8.365.2.14versionlength	338
8.366unpack_omaDmNotificationsTlv_t Struct Reference	338
8.366.1 Field Documentation	339
8.366.1.1 notification	339
8.366.1.2 sessionStatus	339
8.367unpack_qmi_t Struct Reference	339
8.367.1 Detailed Description	339
8.367.2 Field Documentation	339
8.367.2.1 msgid	339
8.367.2.2 type	339
8.367.2.3 xid	339
8.368unpack_qos_dataRate_t Struct Reference	339
8.368.1 Detailed Description	339
8.368.2 Field Documentation	339

8.368.2.1 dataRateMax	339
8.368.2.2 guaranteedRate	340
8.369unpack_qos_IPv4Addr_t Struct Reference	340
8.369.1 Detailed Description	340
8.369.2 Field Documentation	340
8.369.2.1 addr	340
8.369.2.2 subnetMask	340
8.370unpack_qos_IPv6Addr_t Struct Reference	340
8.370.1 Detailed Description	340
8.370.2 Field Documentation	340
8.370.2.1 addr	340
8.370.2.2 prefixLen	340
8.371unpack_qos_IPv6TrafCls_t Struct Reference	341
8.371.1 Detailed Description	341
8.371.2 Field Documentation	341
8.371.2.1 mask	341
8.371.2.2 val	341
8.372unpack_qos_pktErrRate_t Struct Reference	341
8.372.1 Detailed Description	341
8.372.2 Field Documentation	341
8.372.2.1 exponent	341
8.372.2.2 multiplier	341
8.373unpack_qos_Port_t Struct Reference	341
8.373.1 Detailed Description	342
8.373.2 Field Documentation	342
8.373.2.1 port	342
8.373.2.2 range	342
8.374unpack_qos_QosFlowInfo_t Struct Reference	342
8.374.1 Detailed Description	342
8.374.2 Field Documentation	343
8.374.2.1 BearerID	343
8.374.2.2 is_RxQFlowGranted_Available	343
8.374.2.3 is_TxQFlowGranted_Available	343
8.374.2.4 NumRxFilters	343
8.374.2.5 NumTxFilters	343
8.374.2.6 QFlowState	343
8.374.2.7 RxQFilter	343
8.374.2.8 RxQFlowGranted	343
8.374.2.9 TxQFilter	343
8.374.2.10TxQFlowGranted	343

8.375unpack_qos_QosFlowInfoState_t Struct Reference	343
8.375.1 Detailed Description	344
8.375.2 Field Documentation	344
8.375.2.1 id	344
8.375.2.2 isNewFlow	344
8.375.2.3 state	344
8.376unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference	344
8.376.1 Detailed Description	344
8.376.2 Field Documentation	344
8.376.2.1 NWQoSStatus	345
8.377unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference	345
8.377.1 Detailed Description	345
8.377.2 Field Documentation	345
8.377.2.1 ambr_dl	345
8.377.2.2 ambr_dl_ext	346
8.377.2.3 ambr_dl_ext2	346
8.377.2.4 ambr_ul	346
8.377.2.5 ambr_ul_ext	346
8.377.2.6 ambr_ul_ext2	346
8.377.2.7 apnId	346
8.378unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference	346
8.378.1 Detailed Description	346
8.378.2 Field Documentation	347
8.378.2.1 apnId	347
8.378.2.2 numQosFlow	347
8.378.2.3 qosFlow	347
8.378.2.4 total_rx_bytes	347
8.378.2.5 total_rx_pkt	347
8.378.2.6 total_tx_bytes	347
8.378.2.7 total_tx_bytes_drp	347
8.378.2.8 total_tx_pkt	347
8.378.2.9 total_tx_pkt_drp	347
8.379unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference	347
8.379.1 Detailed Description	347
8.379.2 Field Documentation	347
8.379.2.1 NumFlows	348
8.379.2.2 QosFlowInfo	348
8.380unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference	348
8.380.1 Detailed Description	348
8.380.2 Field Documentation	348

8.380.2.1 status	348
8.381unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference	348
8.381.1 Detailed Description	348
8.381.2 Field Documentation	348
8.381.2.1 event	348
8.382unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference	348
8.382.1 Detailed Description	349
8.382.2 Field Documentation	350
8.382.2.1 event	350
8.382.2.2 id	350
8.382.2.3 reason	350
8.382.2.4 status	350
8.383unpack_qos_swiQosFilter_t Struct Reference	350
8.383.1 Detailed Description	351
8.383.2 Field Documentation	352
8.383.2.1 EspSpi	352
8.383.2.2 Id	352
8.383.2.3 index	352
8.383.2.4 IPv4DstAddr	352
8.383.2.5 IPv4SrcAddr	352
8.383.2.6 IPv4Tos	352
8.383.2.7 IPv6DstAddr	352
8.383.2.8 IPv6Label	352
8.383.2.9 IPv6SrcAddr	352
8.383.2.10IPv6TrafCls	352
8.383.2.11is_EspSpi_Available	352
8.383.2.12s_Id_Available	352
8.383.2.13s_IPv4DstAddr_Available	352
8.383.2.14s_IPv4SrcAddr_Available	352
8.383.2.15s_IPv4Tos_Available	352
8.383.2.16s_IPv6DstAddr_Available	352
8.383.2.17s_IPv6Label_Available	352
8.383.2.18s_IPv6SrcAddr_Available	352
8.383.2.19s_IPv6TrafCls_Available	353
8.383.2.20s_NxtHdrProto_Available	353
8.383.2.21is_Precedence_Available	353
8.383.2.22s_TCPDstPort_Available	353
8.383.2.23s_TCPSrcPort_Available	353
8.383.2.24s_TranDstPort_Available	353
8.383.2.25s_TranSrcPort_Available	353

8.383.2.26s_UDPDstPort_Available	353
8.383.2.27s_UDPSrcPort_Available	353
8.383.2.28NxtHdrProto	353
8.383.2.29Precedence	353
8.383.2.30TCPDstPort	353
8.383.2.31TCPSrcPort	353
8.383.2.32TranDstPort	353
8.383.2.33TranSrcPort	353
8.383.2.34UDPDstPort	353
8.383.2.35UDPSrcPort	353
8.383.2.36version	353
8.384unpack_qos_swiQosFlow_t Struct Reference	353
8.384.1 Detailed Description	354
8.384.2 Field Documentation	356
8.384.2.1 DataRate	356
8.384.2.2 index	356
8.384.2.3 is_DataRate_Available	356
8.384.2.4 is_Jitter_Available	356
8.384.2.5 is_Latency_Available	356
8.384.2.6 is_LteQci_Available	356
8.384.2.7 is_MaxAllowedPktSz_Available	356
8.384.2.8 is_MinPolicedPktSz_Available	356
8.384.2.9 is_PktErrRate_Available	356
8.384.2.10s_ProfileId3GPP2_Available	356
8.384.2.11is_TokenBucket_Available	356
8.384.2.12s_TrafficClass_Available	356
8.384.2.13s_val_3GPP2Pri_Available	356
8.384.2.14s_val_3GPPImCn_Available	356
8.384.2.15s_val_3GPPResResidualBER_Available	357
8.384.2.16s_val_3GPPSigInd_Available	357
8.384.2.17s_val_3GPPTraHdlPri_Available	357
8.384.2.18Jitter	357
8.384.2.19Latency	357
8.384.2.20LteQci	357
8.384.2.21MaxAllowedPktSz	357
8.384.2.22MinPolicedPktSz	357
8.384.2.23PktErrRate	357
8.384.2.24ProfileId3GPP2	357
8.384.2.25TokenBucket	357
8.384.2.26TrafficClass	357

8.384.2.27val_3GPP2Pri	357
8.384.2.28val_3GPPImCn	357
8.384.2.29val_3GPPResResidualBER	357
8.384.2.30val_3GPPSigInd	357
8.384.2.31val_3GPPTraHdlPri	357
8.385unpack_qos_tokenBucket_t Struct Reference	357
8.385.1 Detailed Description	357
8.385.2 Field Documentation	358
8.385.2.1 bucketSz	358
8.385.2.2 peakRate	358
8.385.2.3 tokenRate	358
8.386unpack_qos_Tos_t Struct Reference	358
8.386.1 Detailed Description	358
8.386.2 Field Documentation	358
8.386.2.1 mask	358
8.386.2.2 val	358
8.387unpack_QosFlowStat_t Struct Reference	358
8.387.1 Detailed Description	358
8.387.2 Field Documentation	359
8.387.2.1 bearerId	359
8.387.2.2 tx_bytes	359
8.387.2.3 tx_bytes_drp	359
8.387.2.4 tx_pkt	359
8.387.2.5 tx_pkt_drp	359
8.388unpack_sms_SendSMS_t Struct Reference	359
8.388.1 Detailed Description	359
8.388.2 Field Documentation	359
8.388.2.1 messageFailureCode	359
8.388.2.2 messageId	359
8.389unpack_sms_SetNewSMSCallback_ind_t Struct Reference	360
8.389.1 Detailed Description	360
8.389.2 Field Documentation	360
8.389.2.1 ETWSPLMNTlv	360
8.389.2.2 ETWSTlv	360
8.389.2.3 IMSTlv	360
8.389.2.4 MMTlv	360
8.389.2.5 NewMMTlv	361
8.389.2.6 SMSCTlv	361
8.389.2.7 TRMessageTlv	361
8.390unpack_sms_SetNewSMSCallback_t Struct Reference	361

8.391unpack_sms_SLQSDeleteSMS_t Struct Reference	361
8.392unpack_sms_SLQSGetSMS_t Struct Reference	361
8.392.1 Detailed Description	361
8.392.2 Field Documentation	361
8.392.2.1 message	361
8.392.2.2 messageFormat	362
8.392.2.3 messageSize	362
8.392.2.4 messageTag	362
8.393unpack_sms_SLQSGetSMSList_t Struct Reference	362
8.393.1 Detailed Description	362
8.393.2 Field Documentation	362
8.393.2.1 messageList	362
8.393.2.2 messageListSize	362
8.394unpack_sms_SLQSModifySMSStatus_t Struct Reference	362
8.395unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference	362
8.395.1 Detailed Description	362
8.395.2 Field Documentation	363
8.395.2.1 messageMode	363
8.395.2.2 storageType	363
8.396unpack_swiloc_SwiLocGetAutoStart_t Struct Reference	363
8.396.1 Detailed Description	363
8.396.2 Field Documentation	364
8.396.2.1 fix_rate	364
8.396.2.2 fix_rate_reported	364
8.396.2.3 fix_type	364
8.396.2.4 fix_type_reported	364
8.396.2.5 function	364
8.396.2.6 function_reported	364
8.396.2.7 max_dist	364
8.396.2.8 max_dist_reported	364
8.396.2.9 max_time	364
8.396.2.10max_time_reported	364
8.397unpack_swioma_SLQSOMADMAAlertCallback_ind_t Struct Reference	364
8.397.1 Detailed Description	365
8.397.2 Field Documentation	365
8.397.2.1 eventType	365
8.397.2.2 SessionInfoConfig	365
8.397.2.3 SessionInfoFota	365
8.397.2.4 SessionInfoNotification	365
8.398unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference	365

8.398.1 Detailed Description	366
8.398.2 Field Documentation	367
8.398.2.1 Date	367
8.398.2.2 DateLength	367
8.398.2.3 PkgDescLength	367
8.398.2.4 PkgDescription	367
8.398.2.5 PkgName	367
8.398.2.6 PkgNameLength	367
8.398.2.7 RetryCount	367
8.398.2.8 SessionState	367
8.398.2.9 SessionType	367
8.398.2.10Severity	368
8.398.2.11Source	368
8.398.2.12SourceLength	368
8.398.2.13Status	368
8.398.2.14Time	368
8.398.2.15TimeLength	368
8.398.2.16UpdateCompleteStatus	368
8.399unpack_swioama_SLQSOMADMGetSettings_t Struct Reference	368
8.399.1 Detailed Description	368
8.399.2 Field Documentation	369
8.399.2.1 Autosdm	369
8.399.2.2 FOTAdownload	369
8.399.2.3 FOTAUpdate	369
8.399.2.4 FwAutoCheck	369
8.399.2.5 OMADMEEnabled	369
8.400unpack_swioama_SLQSOMADMStartSession_t Struct Reference	369
8.400.1 Detailed Description	369
8.400.2 Field Documentation	370
8.400.2.1 FwAvailability	370
8.401unpack_uim_ChangePin_t Struct Reference	370
8.401.1 Detailed Description	370
8.401.2 Field Documentation	370
8.401.2.1 pEncryptedPIN1	370
8.401.2.2 pIndicationToken	370
8.401.2.3 pRemainingRetries	370
8.401.2.4 Tlvresult	371
8.402unpack_uim_GetCardStatus_t Struct Reference	371
8.402.1 Detailed Description	371
8.402.2 Field Documentation	371

8.402.2.1 pCardStatus	371
8.402.2.2 pHotSwapStatus	371
8.402.2.3 Tlvresult	371
8.403unpack_uim_ReadTransparent_t Struct Reference	371
8.403.1 Detailed Description	371
8.403.2 Field Documentation	372
8.403.2.1 pCardResult	372
8.403.2.2 pEncryptedData	372
8.403.2.3 pIndicationToken	372
8.403.2.4 pReadResult	372
8.403.2.5 Tlvresult	372
8.404unpack_uim_SetPinProtection_t Struct Reference	372
8.404.1 Detailed Description	372
8.404.2 Field Documentation	373
8.404.2.1 pEncryptedPIN1	373
8.404.2.2 pIndicationToken	373
8.404.2.3 pRemainingRetries	373
8.404.2.4 Tlvresult	373
8.405unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference	373
8.405.1 Detailed Description	373
8.405.2 Field Documentation	373
8.405.2.1 bNumberOfPhySlots	373
8.405.2.2 slotsstatusChange	373
8.406unpack_uim_SLQSUIMEventRegister_t Struct Reference	373
8.406.1 Detailed Description	374
8.406.2 Field Documentation	374
8.406.2.1 eventMask	374
8.407unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference	374
8.407.1 Detailed Description	374
8.407.2 Field Documentation	374
8.407.2.1 pNumberOfPhySlot	374
8.407.2.2 pUimSlotsStatus	374
8.408unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference	374
8.408.1 Detailed Description	374
8.408.2 Field Documentation	375
8.408.2.1 pCardStatus	375
8.409unpack_uim_UnblockPin_t Struct Reference	375
8.409.1 Detailed Description	375
8.409.2 Field Documentation	375
8.409.2.1 pEncryptedPIN1	375

8.409.2.2 pIndicationToken	375
8.409.2.3 pRemainingRetries	375
8.409.2.4 Tlvresult	375
8.410unpack_uim_VerifyPin_t Struct Reference	375
8.410.1 Detailed Description	375
8.410.2 Field Documentation	376
8.410.2.1 pEncryptedPIN1	376
8.410.2.2 pIndicationToken	376
8.410.2.3 pRemainingRetries	376
8.410.2.4 Tlvresult	376
8.411unpack_wds_GetByteTotals_t Struct Reference	376
8.411.1 Detailed Description	376
8.411.2 Field Documentation	376
8.411.2.1 pRXTotalBytes	376
8.411.2.2 pTXTotalBytes	376
8.412unpack_wds_GetConnectionRate_t Struct Reference	377
8.412.1 Detailed Description	377
8.412.2 Field Documentation	377
8.412.2.1 currentChannelRXRate	377
8.412.2.2 currentChannelTXRate	377
8.412.2.3 maxChannelRXRate	377
8.412.2.4 maxChannelTXRate	377
8.413unpack_wds_GetDefaultProfile_t Struct Reference	377
8.413.1 Detailed Description	377
8.413.2 Field Documentation	378
8.413.2.1 apnname	378
8.413.2.2 apnsize	378
8.413.2.3 auth	378
8.413.2.4 ipaddr	378
8.413.2.5 ipaddrv6	378
8.413.2.6 name	378
8.413.2.7 namesize	378
8.413.2.8 pdptype	378
8.413.2.9 pridns	378
8.413.2.10pridnsv6	378
8.413.2.11secdns	378
8.413.2.12secdnsv6	378
8.413.2.13username	378
8.413.2.14usersize	378
8.414unpack_wds_GetDefaultProfileNum_t Struct Reference	378

8.414.1 Detailed Description	378
8.414.2 Field Documentation	379
8.414.2.1 index	379
8.415unpack_wds_GetDormancyState_t Struct Reference	379
8.415.1 Detailed Description	379
8.415.2 Field Documentation	379
8.415.2.1 dormancyState	379
8.416unpack_wds_GetLastMobileIPError_t Struct Reference	379
8.416.1 Detailed Description	379
8.416.2 Field Documentation	379
8.416.2.1 error	379
8.417unpack_wds_GetMobileIP_t Struct Reference	379
8.417.1 Detailed Description	379
8.417.2 Field Documentation	380
8.417.2.1 mipMode	380
8.418unpack_wds_GetMobileIPProfile_t Struct Reference	380
8.418.1 Detailed Description	380
8.418.2 Field Documentation	380
8.418.2.1 AAASPI	380
8.418.2.2 AAASState	380
8.418.2.3 address	380
8.418.2.4 enabled	380
8.418.2.5 HASPI	380
8.418.2.6 HASState	380
8.418.2.7 NAI	380
8.418.2.8 naiSize	380
8.418.2.9 primaryHA	381
8.418.2.10revTunneling	381
8.418.2.11secondaryHA	381
8.419unpack_wds_GetPacketStatistics_t Struct Reference	381
8.419.1 Detailed Description	381
8.419.2 Field Documentation	382
8.419.2.1 pRXDroppedCount	382
8.419.2.2 pRXOkBytesCount	382
8.419.2.3 pRXOKBytesLastCall	382
8.419.2.4 pRXPacketErrors	382
8.419.2.5 pRXPacketOverflows	382
8.419.2.6 pRXPacketSuccesses	382
8.419.2.7 pTXDroppedCount	382
8.419.2.8 pTXOkBytesCount	382

8.419.2.9 pTXOKBytesLastCall	382
8.419.2.10pTXPacketErrors	382
8.419.2.11pTXPacketOverflows	382
8.419.2.12pTXPacketSuccesses	382
8.420unpack_wds_GetPacketStatus_t Struct Reference	382
8.420.1 Detailed Description	383
8.420.2 Field Documentation	383
8.420.2.1 rXDroppedCount	383
8.420.2.2 rXOkBytesCount	383
8.420.2.3 rXOKBytesLastCall	383
8.420.2.4 rXPacketErrors	383
8.420.2.5 rXPacketOverflows	383
8.420.2.6 rXPacketSuccesses	383
8.420.2.7 tXDroppedCount	383
8.420.2.8 tXOkBytesCount	383
8.420.2.9 tXOKBytesLastCall	383
8.420.2.10tXPacketErrors	383
8.420.2.11tXPacketOverflows	383
8.420.2.12tXPacketSuccesses	383
8.421unpack_wds_GetSessionDuration_t Struct Reference	383
8.421.1 Detailed Description	384
8.421.2 Field Documentation	384
8.421.2.1 callDuration	384
8.422unpack_wds_GetSessionState_t Struct Reference	384
8.422.1 Detailed Description	384
8.422.2 Field Documentation	384
8.422.2.1 connectionStatus	384
8.423unpack_wds_RMSetTransferStatistics_t Struct Reference	384
8.424unpack_wds_SetMobileIPProfile_t Struct Reference	384
8.425unpack_wds_SLQSCreateProfile_t Struct Reference	384
8.425.1 Detailed Description	384
8.425.2 Field Documentation	384
8.425.2.1 pCreateProfileOut	384
8.425.2.2 pProfileID	385
8.425.2.3 Tlvresult	385
8.426unpack_wds_SLQSDeleteProfile_t Struct Reference	385
8.426.1 Detailed Description	385
8.426.2 Field Documentation	385
8.426.2.1 extendedErrorCode	385
8.427unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference	385

8.427.1 Detailed Description	385
8.427.2 Field Documentation	386
8.427.2.1 _3gppRelease	386
8.427.2.2 defaultPDNEnabled	386
8.427.2.3 LTEAttachProfile	386
8.427.2.4 LTEAttachProfileList	386
8.427.2.5 LTEAttachProfileListLen	386
8.427.2.6 profileList	386
8.428unpack_wds_SLQSGGetCurrDataSystemStat_t Struct Reference	386
8.428.1 Detailed Description	386
8.428.2 Field Documentation	387
8.428.2.1 currNetworkInfo	387
8.428.2.2 networkInfoLen	387
8.428.2.3 prefNetwork	387
8.429unpack_wds_SLQSGGetCurrentChannelRate_t Struct Reference	387
8.429.1 Detailed Description	387
8.429.2 Field Documentation	388
8.429.2.1 current_channel_rx_rate	388
8.429.2.2 current_channel_tx_rate	388
8.429.2.3 max_channel_rx_rate	388
8.429.2.4 max_channel_tx_rate	388
8.430unpack_wds_SLQSGGetDataBearerTechnology_t Struct Reference	388
8.430.1 Detailed Description	388
8.430.2 Field Documentation	388
8.430.2.1 curDataBearerTechnology	388
8.430.2.2 dataBearerMask	388
8.430.2.3 lastCallDataBearerTechnology	388
8.431unpack_wds_SLQSGGetDUNCallInfo_t Struct Reference	388
8.431.1 Detailed Description	389
8.431.2 Field Documentation	389
8.431.2.1 callEndReason	389
8.431.2.2 channelRate	389
8.431.2.3 connectionStatus	389
8.431.2.4 dataBearerTech	389
8.431.2.5 dormancyStatus	389
8.431.2.6 lastCallDataBearerTech	389
8.431.2.7 lastCallRXOKBytesCnt	389
8.431.2.8 lastCallTXOKBytesCnt	389
8.431.2.9 mdmCallDurationActive	389
8.431.2.10rxOKBytesCount	389

8.431.2.1 ttxOKBytesCount	389
8.432unpack_wds_SLQSGetProfileSettings_t Struct Reference	389
8.432.1 Field Documentation	389
8.432.1.1 pProfileSettings	390
8.432.1.2 ProfileType	390
8.432.1.3 Tlvresult	390
8.433unpack_wds_SLQSGetRuntimeSettings_t Struct Reference	390
8.433.1 Detailed Description	390
8.433.2 Field Documentation	391
8.433.2.1 APNName	391
8.433.2.2 Authentication	391
8.433.2.3 DomainList	391
8.433.2.4 GPRSGrantedQoS	391
8.433.2.5 GWAddressV4	391
8.433.2.6 IMCNflag	391
8.433.2.7 IPFamilyPreference	391
8.433.2.8 IPv4	391
8.433.2.9 IPV6AddrInfo	391
8.433.2.10 IPV6GWAddrInfo	391
8.433.2.11 Mtu	391
8.433.2.12 PCSCFAddrPCO	391
8.433.2.13 PCSCFFQDNAddrList	391
8.433.2.14 PDPTtype	391
8.433.2.15 PrimaryDNSV4	391
8.433.2.16 PrimaryDNSV6	391
8.433.2.17 ProfileID	391
8.433.2.18 ProfileName	391
8.433.2.19 SecondaryDNSV4	392
8.433.2.20 SecondaryDNSV6	392
8.433.2.21 ServerAddrList	392
8.433.2.22 SubnetMaskV4	392
8.433.2.23 Technology	392
8.433.2.24 UMTSGrantedQoS	392
8.433.2.25 Username	392
8.434unpack_wds_SLQSModifyProfile_t Struct Reference	392
8.434.1 Detailed Description	392
8.434.2 Field Documentation	392
8.434.2.1 pExtErrorCode	392
8.435unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference	392
8.435.1 Detailed Description	392

8.435.2 Field Documentation	392
8.435.2.1 Tlvresult	392
8.436unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference	392
8.436.1 Detailed Description	393
8.436.2 Field Documentation	393
8.436.2.1 bearerID	393
8.436.2.2 conn_status	393
8.436.2.3 ipFamily	393
8.436.2.4 reconfigReqd	393
8.436.2.5 sessionEndReason	393
8.436.2.6 techName	393
8.436.2.7 verboseSessnEndReason	393
8.436.2.8 verboseSessnEndReasonType	393
8.437unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference	393
8.437.1 Detailed Description	394
8.437.2 Field Documentation	394
8.437.2.1 currDBTechAvail	394
8.437.2.2 currNWInfo	394
8.437.2.3 dataSysStatAvail	394
8.437.2.4 dBTechAvail	394
8.437.2.5 dBTechnology	394
8.437.2.6 dormancyStatAvail	394
8.437.2.7 dormancyStatus	394
8.437.2.8 mipstatAvail	394
8.437.2.9 mipStatus	395
8.437.2.10netInfoLen	395
8.437.2.11prefNetwork	395
8.437.2.12atMask	395
8.437.2.13rx_bytes	395
8.437.2.14rx_pkts	395
8.437.2.15soMask	395
8.437.2.16x_bytes	395
8.437.2.17tx_pkts	395
8.437.2.18ferStatAvail	395
8.438unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	395
8.438.1 Detailed Description	395
8.438.2 Field Documentation	395
8.438.2.1 pHwConfig	395
8.438.2.2 pRequestOptionList	395
8.439unpack_wds_SLQSSGetLoopback_t Struct Reference	395

8.439.1 Detailed Description	395
8.439.2 Field Documentation	396
8.439.2.1 ByteLoopbackMode	396
8.439.2.2 ByteLoopbackMultiplier	396
8.440unpack_wds_SLQSSStartDataSession_t Struct Reference	396
8.440.1 Detailed Description	396
8.440.2 Field Documentation	396
8.440.2.1 pFailureReason	396
8.440.2.2 psid	396
8.440.2.3 pVerboseFailReasonType	396
8.440.2.4 pVerboseFailureReason	397
8.441unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	397
8.441.1 Detailed Description	397
8.441.2 Field Documentation	397
8.441.2.1 apnName	397
8.441.2.2 bearerId	398
8.441.2.3 contextId	398
8.441.2.4 ipv4Address	398
8.441.2.5 ipv4GWAddress	398
8.441.2.6 ipv6Address	398
8.441.2.7 ipv6GWAddress	398
8.441.2.8 prDNSIPv4Address	398
8.441.2.9 prDNSIPv6Address	398
8.441.2.10prPCSCFIPv4Address	398
8.441.2.11prPCSCFIPv6Address	398
8.441.2.12seDNSIPv4Address	398
8.441.2.13seDNSIPv6Address	398
8.441.2.14sePCSCFIPv4Address	398
8.441.2.15sePCSCFIPv6Address	398
8.442UnPackGetProfileSettingOut Struct Reference	398
8.442.1 Field Documentation	398
8.442.1.1 curProfile	398
8.442.1.2 pExtErrCode	398
8.443unpackWdsProfileParam Union Reference	398
8.443.1 Field Documentation	398
8.443.1.1 SlqsProfile3GPP	398
8.443.1.2 SlqsProfile3GPP2	398
8.444wds_currNetworkInfo Struct Reference	399
8.444.1 Detailed Description	399
8.444.2 Field Documentation	400

8.444.2.1 NetworkType	400
8.444.2.2 RATMask	400
8.444.2.3 SOMask	400
8.445wds_Domain Struct Reference	400
8.445.1 Detailed Description	400
8.445.2 Field Documentation	401
8.445.2.1 domainLen	401
8.445.2.2 domainName	401
8.446wds_DomainNameList Struct Reference	401
8.446.1 Detailed Description	401
8.446.2 Field Documentation	401
8.446.2.1 domain	401
8.446.2.2 numInstances	401
8.447wds_GPRSQoS Struct Reference	401
8.447.1 Detailed Description	401
8.447.2 Field Documentation	402
8.447.2.1 delayClass	402
8.447.2.2 meanThroughputClass	402
8.447.2.3 peakThroughputClass	402
8.447.2.4 precedenceClass	402
8.447.2.5 reliabilityClass	402
8.448wds_IPV6AddressInfo Struct Reference	402
8.448.1 Detailed Description	402
8.448.2 Field Documentation	402
8.448.2.1 IPAddressV6	402
8.448.2.2 IPV6PrefixLen	402
8.449wds_IPV6GWAddressInfo Struct Reference	402
8.449.1 Detailed Description	403
8.449.2 Field Documentation	403
8.449.2.1 gwAddressV6	403
8.449.2.2 gwV6PrefixLen	403
8.450wds_PCSCFFQDNAddress Struct Reference	403
8.450.1 Detailed Description	403
8.450.2 Field Documentation	403
8.450.2.1 fqdnAddr	403
8.450.2.2 fqdnLen	403
8.451wds_PCSCFFQDNAddressList Struct Reference	403
8.451.1 Detailed Description	404
8.451.2 Field Documentation	404
8.451.2.1 numInstances	404

8.451.2.2 pscfQDNAddress	404
8.452wds_PCSCFIPv4ServerAddressList Struct Reference	404
8.452.1 Detailed Description	404
8.452.2 Field Documentation	404
8.452.2.1 numInstances	404
8.452.2.2 pscsfIPv4Addr	404
8.453wds_ProfileIdentifier Struct Reference	404
8.453.1 Detailed Description	405
8.453.2 Field Documentation	405
8.453.2.1 profileIndex	405
8.453.2.2 profileType	405
8.454wds_profileInfo Union Reference	405
8.454.1 Detailed Description	405
8.454.2 Field Documentation	405
8.454.2.1 SIqsProfile3GPP	405
8.454.2.2 SIqsProfile3GPP2	405
8.455wds_UMTSMInQoS Struct Reference	405
8.455.1 Detailed Description	406
8.455.2 Field Documentation	407
8.455.2.1 deliveryErrSDU	407
8.455.2.2 grntDownlinkBitrate	407
8.455.2.3 grntUplinkBitrate	407
8.455.2.4 maxDownlinkBitrate	407
8.455.2.5 maxSDUSize	407
8.455.2.6 maxUplinkBitrate	407
8.455.2.7 qosDeliveryOrder	407
8.455.2.8 resBerRatio	408
8.455.2.9 sduErrorRatio	408
8.455.2.10trafficClass	408
8.455.2.11trafficPriority	408
8.455.2.12transferDelay	408
8.456wdsDhcpv4HwConfig Struct Reference	408
8.456.1 Detailed Description	408
8.456.2 Field Documentation	408
8.456.2.1 chaddr	408
8.456.2.2 chaddrLen	408
8.456.2.3 hwType	408
8.457wdsDhcpv4Option Struct Reference	408
8.457.1 Detailed Description	408
8.457.2 Field Documentation	409

8.457.2.1 optCode	409
8.457.2.2 optVal	409
8.457.2.3 optValLen	409
8.458wdsDhcpv4OptionList Struct Reference	409
8.458.1 Detailed Description	409
8.458.2 Field Documentation	409
8.458.2.1 numOpt	409
8.458.2.2 pOptList	409
8.459wdsDhcpv4ProfileId Struct Reference	409
8.459.1 Detailed Description	409
8.459.2 Field Documentation	410
8.459.2.1 profileId	410
8.459.2.2 profileType	410
9 File Documentation	411
9.1 apdoxypages.c File Reference	411
9.1.1 Detailed Description	411
9.2 common.h File Reference	411
9.2.1 Macro Definition Documentation	413
9.2.1.1 DEAUT_LOC_TIMEOUT_IN_SEC	413
9.2.1.2 MINREQBKLEN	413
9.2.1.3 MSGID_AND_LEN	413
9.2.1.4 MSGID_DONT_CARE	413
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM	413
9.2.1.6 SDU_HDR_LEN	413
9.2.1.7 UNUSEDPARAM	413
9.2.2 Typedef Documentation	413
9.2.2.1 logger	413
9.2.3 Enumeration Type Documentation	413
9.2.3.1 eLOG_LEVEL	413
9.2.3.2 eQMI_SVC	413
9.2.3.3 eTimeout	414
9.2.3.4 msgtype	414
9.2.4 Function Documentation	414
9.2.4.1 fill_pack_ctx	414
9.2.4.2 fill_sdu_hdr	414
9.2.4.3 get_version	414
9.2.4.4 helper_get_resp_ctx	414
9.2.4.5 helper_get_xid	415
9.2.4.6 helper_set_log_func	415

9.2.4.7	helper_set_log_lvl	415
9.2.4.8	libpack_GetVersion	415
9.2.4.9	libpack_log	415
9.2.4.10	unpack_result_code_only	415
9.2.5	Variable Documentation	415
9.2.5.1	glog	415
9.2.5.2	gloglvl	415
9.3	dms.h File Reference	415
9.3.1	Macro Definition Documentation	420
9.3.1.1	DMS_IMGDETAILS_LEN	420
9.3.1.2	DMS_MAX_CUST_ID_LEN	420
9.3.1.3	DMS_MAX_CUST_VALUE_LEN	420
9.3.1.4	DMS_MAX_FWUPDATE_LOG_STR_SZ	420
9.3.1.5	DMS_MAX_FWUPDATE_REF_STR_SZ	420
9.3.1.6	DMS_PM_FACTORY	420
9.3.1.7	DMS_PM_LOW	420
9.3.1.8	DMS_PM_OFFLINE	420
9.3.1.9	DMS_PM_ONLINE	420
9.3.1.10	DMS_PM_PERSISTENT_LOW	420
9.3.1.11	DMS_PM_RESET	420
9.3.1.12	DMS_PM_SHUT_DOWN	420
9.3.1.13	DMS_SET_REPORT_DISABLE	420
9.3.1.14	DMS_SET_REPORT_ENABLE	420
9.3.1.15	DMS_SLQSFWINFO_APPVERSION_SZ	420
9.3.1.16	DMS_SLQSFWINFO_BOOTVERSION_SZ	420
9.3.1.17	DMS_SLQSFWINFO_CARRIER_SZ	420
9.3.1.18	DMS_SLQSFWINFO_CUR_CARR_NAME	420
9.3.1.19	DMS_SLQSFWINFO_CUR_CARR_REV	420
9.3.1.20	DMS_SLQSFWINFO_MODELID_SZ	420
9.3.1.21	DMS_SLQSFWINFO_PACKAGEID_SZ	420
9.3.1.22	DMS_SLQSFWINFO_PRIVERSION_SZ	420
9.3.1.23	DMS_SLQSFWINFO_SKU_SZ	420
9.3.1.24	DMS_SWI_SET_IND_DISABLE	420
9.3.1.25	DMS_SWI_SET_IND_ENABLE	420
9.3.1.26	DMS_UINT8_MAX_STRING_SZ	420
9.3.1.27	MAX_BUILD_ID_LEN	420
9.3.1.28	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	421
9.3.1.29	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	421
9.3.1.30	UNIQUE_ID_LEN	421
9.3.2	Function Documentation	421

9.3.2.1	pack_dms_GetActivationState	421
9.3.2.2	pack_dms_GetBandCapability	421
9.3.2.3	pack_dms_GetCrashAction	421
9.3.2.4	pack_dms_GetCustFeature	422
9.3.2.5	pack_dms_GetCustFeaturesV2	422
9.3.2.6	pack_dms_GetDeviceCap	422
9.3.2.7	pack_dms_GetDeviceCapabilities	422
9.3.2.8	pack_dms_GetDeviceHardwareRev	423
9.3.2.9	pack_dms_GetDeviceMfr	423
9.3.2.10	pack_dms_GetDeviceSerialNumbers	424
9.3.2.11	pack_dms_GetFirmwareInfo	424
9.3.2.12	pack_dms_GetFirmwareRevision	424
9.3.2.13	pack_dms_GetFirmwareRevisions	425
9.3.2.14	pack_dms_GetFSN	425
9.3.2.15	pack_dms_GetHardwareRevision	425
9.3.2.16	pack_dms_GetIMSI	426
9.3.2.17	pack_dms_GetManufacturer	426
9.3.2.18	pack_dms_GetModelID	426
9.3.2.19	pack_dms_GetNetworkTime	427
9.3.2.20	pack_dms_GetOfflineReason	427
9.3.2.21	pack_dms_GetPower	427
9.3.2.22	pack_dms_GetPRLVersion	428
9.3.2.23	pack_dms_GetSerialNumbers	428
9.3.2.24	pack_dms_GetUSBComp	429
9.3.2.25	pack_dms_GetVoiceNumber	429
9.3.2.26	pack_dms_SetCrashAction	429
9.3.2.27	pack_dms_SetCustFeature	430
9.3.2.28	pack_dms_SetCustFeaturesV2	430
9.3.2.29	pack_dms_SetEventReport	431
9.3.2.30	pack_dms_SetFirmwarePreference	431
9.3.2.31	pack_dms_SetPower	431
9.3.2.32	pack_dms_SetUSBComp	432
9.3.2.33	pack_dms_SLQSDmsSwiGetResetInfo	432
9.3.2.34	pack_dms_SLQSDmsSwiIndicationRegister	433
9.3.2.35	pack_dms_SLQSGetBandCapability	433
9.3.2.36	pack_dms_SLQSSwiClearDyingGaspStatistics	433
9.3.2.37	pack_dms_SLQSSwiGetDyingGaspCfg	434
9.3.2.38	pack_dms_SLQSSwiGetDyingGaspStatistics	434
9.3.2.39	pack_dms_SLQSSwiGetFirmwareCurr	434
9.3.2.40	pack_dms_SLQSSwiGetFwUpdateStatus	435

9.3.2.41	pack_dms_SLQSSwiSetDyingGaspCfg	435
9.3.2.42	pack_dms_UIMGetICCID	435
9.3.2.43	unpack_dms_GetActivationState	436
9.3.2.44	unpack_dms_GetBandCapability	436
9.3.2.45	unpack_dms_GetCrashAction	437
9.3.2.46	unpack_dms_GetCustFeature	437
9.3.2.47	unpack_dms_GetCustFeaturesV2	437
9.3.2.48	unpack_dms_GetDeviceCap	437
9.3.2.49	unpack_dms_GetDeviceCapabilities	438
9.3.2.50	unpack_dms_GetDeviceHardwareRev	438
9.3.2.51	unpack_dms_GetDeviceMfr	438
9.3.2.52	unpack_dms_GetDeviceSerialNumbers	439
9.3.2.53	unpack_dms_GetFirmwareInfo	439
9.3.2.54	unpack_dms_GetFirmwareRevision	439
9.3.2.55	unpack_dms_GetFirmwareRevisions	440
9.3.2.56	unpack_dms_GetFSN	440
9.3.2.57	unpack_dms_GetHardwareRevision	440
9.3.2.58	unpack_dms_GetIMSI	441
9.3.2.59	unpack_dms_GetManufacturer	441
9.3.2.60	unpack_dms_GetModelID	441
9.3.2.61	unpack_dms_GetNetworkTime	442
9.3.2.62	unpack_dms_GetOfflineReason	442
9.3.2.63	unpack_dms_GetPower	442
9.3.2.64	unpack_dms_GetPRLVersion	443
9.3.2.65	unpack_dms_GetSerialNumbers	443
9.3.2.66	unpack_dms_GetUSBComp	443
9.3.2.67	unpack_dms_GetVoiceNumber	444
9.3.2.68	unpack_dms_SetCrashAction	444
9.3.2.69	unpack_dms_SetCustFeature	444
9.3.2.70	unpack_dms_SetCustFeaturesV2	445
9.3.2.71	unpack_dms_SetEventReport	445
9.3.2.72	unpack_dms_SetEventReport_ind	446
9.3.2.73	unpack_dms_SetFirmwarePreference	446
9.3.2.74	unpack_dms_SetPower	446
9.3.2.75	unpack_dms_SetUSBComp	447
9.3.2.76	unpack_dms_SLQSDmsSwiGetResetInfo	447
9.3.2.77	unpack_dms_SLQSDmsSwiGetResetInfo_Ind	447
9.3.2.78	unpack_dms_SLQSDmsSwiIndicationRegister	448
9.3.2.79	unpack_dms_SLQSGetBandCapability	448
9.3.2.80	unpack_dms_SLQSSwiClearDyingGaspStatistics	448

9.3.2.81	unpack_dms_SLQSSwiGetDyingGaspCfg	449
9.3.2.82	unpack_dms_SLQSSwiGetDyingGaspStatistics	449
9.3.2.83	unpack_dms_SLQSSwiGetFirmwareCurr	450
9.3.2.84	unpack_dms_SLQSSwiGetFwUpdateStatus	450
9.3.2.85	unpack_dms_SLQSSwiSetDyingGaspCfg	450
9.3.2.86	unpack_dms_UIMGetICCID	451
9.4	fms.h File Reference	451
9.4.1	Macro Definition Documentation	452
9.4.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	452
9.4.1.2	FMS_GOBI_LISTENTRIES_MAX	452
9.4.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	452
9.4.1.4	FMS_GOBI_MBN_IMG_ID_STR_LEN	452
9.4.1.5	FMS_IMAGE_ID_BUILD_ID_LEN	452
9.4.1.6	FMS_IMAGE_ID_IMG_ID_LEN	452
9.4.1.7	FMS_IMAGE_ID_MAX_ENTRIES	452
9.4.1.8	FMS_IMAGE_ID_PRI_IMGTYPE	452
9.4.1.9	FMS_MAX_IMAGE_ID_ELEMENT	452
9.4.1.10	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	452
9.4.2	Function Documentation	452
9.4.2.1	GetValidFwPriCombinations	452
9.4.2.2	pack_fms_GetImagesPreference	453
9.4.2.3	pack_fms_GetStoredImages	453
9.4.2.4	pack_fms_SetImagesPreference	453
9.4.2.5	unpack_fms_GetImagesPreference	454
9.4.2.6	unpack_fms_GetStoredImages	454
9.4.2.7	unpack_fms_SetImagesPreference	454
9.5	libsdh.h File Reference	454
9.5.1	Detailed Description	455
9.5.2	Macro Definition Documentation	456
9.5.2.1	FIRMWARE_INFO_STRING_SIZE	456
9.5.2.2	IMG_MASK_CLEAR	456
9.5.2.3	IMG_MASK_GENERIC	456
9.5.2.4	IMG_MASK_MDM	456
9.5.2.5	IMG_MASK_PRI	456
9.5.2.6	LIBSDP_CARRIER_PACKAGE_SKU	456
9.5.2.7	LIBSDP_SKU_STRING_LENGTH	456
9.5.3	Typedef Documentation	456
9.5.3.1	libSDP_FirmwareInfo	456
9.5.3.2	libsdplogger	456
9.5.4	Enumeration Type Documentation	457

9.5.4.1	libSDP_Fw_Type	457
9.5.4.2	libSDP_fwdwl_error_codes	457
9.5.4.3	libSDP_Models	457
9.5.5	Function Documentation	458
9.5.5.1	libSDP_BuildImagesPreferenceRequest	458
9.5.5.2	libSDP_CalculateImageMask	458
9.5.5.3	libSDP_CheckValidFirmwareInfo	458
9.5.5.4	libSDP_DownloadFW	459
9.5.5.5	libSDP_ExtractFirmwareParametersByPath	459
9.5.5.6	libSDP_getFileType	460
9.5.5.7	libSDP_GetModelFamily	460
9.5.5.8	libSDP_GetVersion	460
9.5.5.9	libsdp_set_log_func	460
9.6	loc.h File Reference	461
9.6.1	Macro Definition Documentation	463
9.6.1.1	LOC_UINT8_MAX_STRING_SZ	463
9.6.1.2	LOCEVENTMASKBATCHFULLNOTIFICATION	463
9.6.1.3	LOCEVENTMASKENGINESTATE	463
9.6.1.4	LOCEVENTMASKFIXSESSIONSTATE	463
9.6.1.5	LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION	463
9.6.1.6	LOCEVENTMASKGEOFENCEBREACHNOTIFICATION	463
9.6.1.7	LOCEVENTMASKGEOFENCEGENALERT	463
9.6.1.8	LOCEVENTMASKGNSSMEASUREMENTREPORT	463
9.6.1.9	LOCEVENTMASKGNSSSVINFO	463
9.6.1.10	LOCEVENTMASKINJECTPOSITIONREQ	464
9.6.1.11	LOCEVENTMASKINJECTPREDICTEDORBITSREQ	464
9.6.1.12	LOCEVENTMASKINJECTTIMERREQ	464
9.6.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	464
9.6.1.14	LOCEVENTMASKINVALIDVALUE	464
9.6.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	464
9.6.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	464
9.6.1.17	LOCEVENTMASKMOTIONDATACONTROL	464
9.6.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	464
9.6.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	464
9.6.1.20	LOCEVENTMASKNMEA	464
9.6.1.21	LOCEVENTMASKPEDOMETERCONTROL	465
9.6.1.22	LOCEVENTMASKPOSITIONREPORT	465
9.6.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	465
9.6.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	465
9.6.1.25	LOCEVENTMASKTIMESYNCREQ	465

9.6.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	465
9.6.1.27	LOCEVENTMASKWIFIREQ	465
9.6.2	Enumeration Type Documentation	465
9.6.2.1	anonymous enum	465
9.6.3	Function Documentation	465
9.6.3.1	pack_loc_DeleteAssistData	465
9.6.3.2	pack_loc_EventRegister	466
9.6.3.3	pack_loc_SetExtPowerState	466
9.6.3.4	pack_loc_SetOperationMode	467
9.6.3.5	pack_loc_SLQSLOCGetBestAvailPos	467
9.6.3.6	pack_loc_Start	467
9.6.3.7	pack_loc_Stop	468
9.6.3.8	unpack_loc_BestAvailPos_Ind	468
9.6.3.9	unpack_loc_DeleteAssistData	468
9.6.3.10	unpack_loc_EngineState_Ind	469
9.6.3.11	unpack_loc_EventRegister	469
9.6.3.12	unpack_loc_PositionRpt_Ind	469
9.6.3.13	unpack_loc_SetExtPowerConfig_Ind	470
9.6.3.14	unpack_loc_SetExtPowerState	470
9.6.3.15	unpack_loc_SetOperationMode	470
9.6.3.16	unpack_loc_SLQSLOCGetBestAvailPos	471
9.6.3.17	unpack_loc_Start	471
9.6.3.18	unpack_loc_Stop	471
9.7	nas.h File Reference	472
9.7.1	Macro Definition Documentation	477
9.7.1.1	NAS_MAX_DESCRIPTION_LENGTH	477
9.7.1.2	NAS_MAX_NUM_NETWORKS	477
9.7.1.3	NAS_MAX_SCC_RX_INFO_INSTANCES	477
9.7.1.4	NAS_OTA_MESSAGE_MAX_BUF_SIZE	477
9.7.1.5	NAS_PLMN_LENGTH	477
9.7.1.6	NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST	478
9.7.2	Enumeration Type Documentation	478
9.7.2.1	LIBPACK_NAS_LTE_CPHY_CA_BW_NRB	478
9.7.2.2	LIBPACK_NAS_LTE_CPHY_SCELL_STATE	478
9.7.2.3	NAS_LTE_CPHY_CA_BW_NRB_LITE	478
9.7.2.4	NAS_LTE_CPHY_SCELL_STATE_LITE	478
9.7.3	Function Documentation	478
9.7.3.1	pack_nas_GetACCOLC	478
9.7.3.2	pack_nas_GetANAAAAAuthenticationStatus	479
9.7.3.3	pack_nas_GetCDMANetworkParameters	479

9.7.3.4	pack_nas_GetHomeNetwork	479
9.7.3.5	pack_nas_GetNetworkPreference	480
9.7.3.6	pack_nas_GetRFInfo	480
9.7.3.7	pack_nas_GetServingNetwork	480
9.7.3.8	pack_nas_GetServingNetworkCapabilities	480
9.7.3.9	pack_nas_GetSignalStrengths	481
9.7.3.10	pack_nas_PerformNetworkScan	481
9.7.3.11	pack_nas_SetACCOLC	481
9.7.3.12	pack_nas_SetLURRejectCallback	482
9.7.3.13	pack_nas_SetNetworkPreference	482
9.7.3.14	pack_nas_SetRFInfoCallback	482
9.7.3.15	pack_nas_SlqsGetLTECphyCAInfo	482
9.7.3.16	pack_nas_SLQSGetNetworkTime	483
9.7.3.17	pack_nas_SLQSGetPLMNName	483
9.7.3.18	pack_nas_SLQSGetServingSystem	483
9.7.3.19	pack_nas_SLQSGetSignalStrength	484
9.7.3.20	pack_nas_SLQSGetSysInfo	484
9.7.3.21	pack_nas_SLQSGetSysSelectionPref	484
9.7.3.22	pack_nas_SLQSIInitiateNetworkRegistration	484
9.7.3.23	pack_nas_SLQSNasConfigSigInfo2	485
9.7.3.24	pack_nas_SLQSNasGetCellLocationInfo	485
9.7.3.25	pack_nas_SLQSNasGetSigInfo	485
9.7.3.26	pack_nas_SLQSNasIndicationRegisterExt	486
9.7.3.27	pack_nas_SLQSNasSwiModemStatus	486
9.7.3.28	pack_nas_SLQSNasSwiOTAMessageCallback	486
9.7.3.29	pack_nas_SLQSSetBandPreference	487
9.7.3.30	pack_nas_SLQSSetSignalStrengthsCallback	487
9.7.3.31	pack_nas_SLQSSetSysSelectionPref	487
9.7.3.32	pack_nas_SLQSSwiGetLteCQI	488
9.7.3.33	pack_nas_SLQSSwiGetLteSccRxInfo	488
9.7.3.34	unpack_nas_GetACCOLC	488
9.7.3.35	unpack_nas_GetANAAAAuthenticationStatus	489
9.7.3.36	unpack_nas_GetCDMANetworkParameters	489
9.7.3.37	unpack_nas_GetHomeNetwork	489
9.7.3.38	unpack_nas_GetNetworkPreference	490
9.7.3.39	unpack_nas_GetRFInfo	490
9.7.3.40	unpack_nas_GetServingNetwork	490
9.7.3.41	unpack_nas_GetServingNetworkCapabilities	490
9.7.3.42	unpack_nas_GetSignalStrengths	491
9.7.3.43	unpack_nas_PerformNetworkScan	491

9.7.3.44	unpack_nas_SetACCOLC	491
9.7.3.45	unpack_nas_SetDataCapabilitiesCallback_ind	492
9.7.3.46	unpack_nas_SetEventReportInd	492
9.7.3.47	unpack_nas_SetLURejectCallback	492
9.7.3.48	unpack_nas_SetNasLTECphyCalIndCallback_ind	492
9.7.3.49	unpack_nas_SetNetworkPreference	493
9.7.3.50	unpack_nas_SetRFInfoCallback	493
9.7.3.51	unpack_nas_SetRoamingIndicatorCallback_ind	493
9.7.3.52	unpack_nas_SetServingSystemCallback_ind	493
9.7.3.53	unpack_nas_SLqsGetLTECphyCAInfo	493
9.7.3.54	unpack_nas_SLQSGetNetworkTime	493
9.7.3.55	unpack_nas_SLQSGetPLMNName	494
9.7.3.56	unpack_nas_SLQSGetServingSystem	494
9.7.3.57	unpack_nas_SLQSGetSignalStrength	494
9.7.3.58	unpack_nas_SLQSGetSysInfo	495
9.7.3.59	unpack_nas_SLQSGetSysSelectionPref	495
9.7.3.60	unpack_nas_SLQSInitiateNetworkRegistration	495
9.7.3.61	unpack_nas_SLQSNasConfigSigInfo2	496
9.7.3.62	unpack_nas_SLQSNasGetCellLocationInfo	496
9.7.3.63	unpack_nas_SLQSNasGetSigInfo	496
9.7.3.64	unpack_nas_SLQSNasIndicationRegisterExt	496
9.7.3.65	unpack_nas_SLQSNasNetworkTimeCallBack_ind	497
9.7.3.66	unpack_nas_SLQSNasSigInfoCallback_ind	497
9.7.3.67	unpack_nas_SLQSNasSwiModemStatus	497
9.7.3.68	unpack_nas_SLQSNasSwiOTAMessageCallback	498
9.7.3.69	unpack_nas_SLQSNasSwiOTAMessageCallback_ind	498
9.7.3.70	unpack_nas_SLQSNasSysInfoCallback_ind	498
9.7.3.71	unpack_nas_SLQSSetBandPreference	499
9.7.3.72	unpack_nas_SLQSSetSignalStrengthsCallback	499
9.7.3.73	unpack_nas_SLQSSetSysSelectionPref	499
9.7.3.74	unpack_nas_SLQSSetSysSelectionPrefCallBack_ind	499
9.7.3.75	unpack_nas_SLQSSwiGetLteCQI	500
9.7.3.76	unpack_nas_SLQSSwiGetLteSccRxInfo	500
9.8	qaGobiApiTableBandClasses.h File Reference	500
9.8.1	Detailed Description	500
9.8.2	Band Classes (Value - Description)	501
9.8.2.1	LTE Bands	502
9.9	qaGobiApiTableCallControlReturnReasons.h File Reference	503
9.9.1	Detailed Description	503
9.9.2	Call Control Result Reasons (Value - Name - Description)	504

9.10	qaGobiApiTableCallEndReasons.h File Reference	504
9.10.1	Detailed Description	504
9.10.2	Call end reason codes (Code - Reason)	505
9.10.2.1	Technology-agnostic call end reasons	505
9.10.2.2	EVDO CDMA 1xEV-DO	505
9.10.2.3	WCDMA/GSM call end reasons	506
9.10.2.4	EVDO CDMA 1xEV-DO	508
9.10.2.5	call end reason type	508
9.10.2.6	Mobile IP call end reasons (Type=1)	508
9.10.2.7	Internal call end reasons (Type=2)	510
9.10.2.8	Call Manager defined call end reasons (Type=3)	511
9.10.2.9	3GPP specification defined call end reasons (Type=6)	516
9.10.2.10	PPP call end reasons (Type=7)	518
9.10.2.11	EHRPD call end reasons (Type=8)	519
9.10.2.12	IPV6 call end reasons (Type=9)	520
9.11	qaGobiApiTableCarrierCodes.h File Reference	520
9.11.1	Detailed Description	520
9.11.2	Carrier Codes (Number - Carrier)	520
9.12	qaGobiApiTableCodingScheme.h File Reference	522
9.12.1	Detailed Description	522
9.12.2	Call Control Result Reasons (Value - Name - Description)	522
9.12.2.1	Use of bits 3..0	522
9.12.3	Coding Group Bits 7..4(0001)	522
9.12.3.1	use of bits 3..0	522
9.12.4	Coding Group Bits 7..4(0010)	523
9.12.4.1	use of bits 3..0	523
9.12.5	Coding Group Bits 7..4(0011)	523
9.12.5.1	use of bits 3..0	523
9.12.6	Coding Group Bits 7..4(01xx)	523
9.12.6.1	use of bits 3..0	523
9.12.7	Coding Group Bits 7..4(1001)	524
9.12.7.1	Reserved coding groups	524
9.12.8	Coding Group Bits 7..4(1010..1101)	524
9.12.8.1	Reserved coding groups	524
9.12.9	Coding Group Bits 7..4(1110)	524
9.12.9.1	Defined by the WAP Forum	524
9.12.10	Coding Group Bits 7..4 (1111)	524
9.12.10.1	Data coding / message handling	524
9.12.11	Macro Definition Documentation	524
9.12.11.1	__GOBI_API_CODING_SCHEME_H__	524

9.13	qaGobiApiTableGpsCapabilityCodes.h File Reference	524
9.13.1	Detailed Description	525
9.13.2	GPS capability (Value - Capability)	525
9.14	qaGobiApiTablePowerModes.h File Reference	525
9.14.1	Detailed Description	525
9.14.2	Power Modes (Value - Description)	525
9.15	qaGobiApiTableRadioInterfaces.h File Reference	526
9.15.1	Detailed Description	526
9.15.2	Radio interface	526
9.15.2.1	Technology (Value - Radio Interface Technology)	526
9.16	qaGobiApiTableRegionCodes.h File Reference	526
9.16.1	Detailed Description	526
9.16.2	Region Codes (Code - Region)	526
9.17	qaGobiApiTableServiceOptions.h File Reference	527
9.17.1	Detailed Description	527
9.17.2	Service Option codes (Code - Reason)	527
9.17.2.1	Description	527
9.18	qaGobiApiTableSupServiceInfoClasses.h File Reference	529
9.18.1	Detailed Description	529
9.18.2	Supplementary Service Information Classes (Value - Service Class)	529
9.19	qaGobiApiTableSwiAudio.h File Reference	529
9.19.1	Detailed Description	529
9.19.2	ACDB Device (Device ID - description)	530
9.19.3	Physical Interface (Device ID - description - Interface parameters)	530
9.20	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	530
9.20.1	Detailed Description	530
9.20.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	530
9.21	qaGobiApiTableVoiceCallEndReasons.h File Reference	531
9.21.1	Detailed Description	531
9.21.2	Voice Call and supplementary services end reason codes (Code - Reason)	532
9.21.2.1	General	532
9.21.2.2	service Errors	533
9.21.2.3	control cause values	534
9.21.2.4	reject causes	536
9.21.2.5	reject causes	536
9.21.2.6	reject causes	537
9.21.2.7	stratum reject causes	537
9.21.2.8	reject causes	537
9.21.2.9	IP end reasons	537
9.22	qmerrno.h File Reference	538

9.22.1	Enumeration Type Documentation	540
9.22.1.1	eQCWWANError	540
9.22.1.2	qm_wds_ds_profile_extended_err_codes	545
9.23	qos.h File Reference	545
9.23.1	Macro Definition Documentation	547
9.23.1.1	LIBPACK_MAX_QOS_FILTERS	547
9.23.1.2	LIBPACK_MAX_QOS_FLOW_PER_APN_STATS	547
9.23.1.3	LIBPACK_MAX_QOS_FLOWS	547
9.23.2	Function Documentation	547
9.23.2.1	pack_qos_SLQSQosGetNetworkStatus	547
9.23.2.2	pack_qos_SLQSQosSwiReadApnExtraParams	547
9.23.2.3	pack_qos_SLQSQosSwiReadDataStats	548
9.23.2.4	pack_qos_SLQSSetQosEventCallback	549
9.23.2.5	unpack_qos_SLQSQosGetNetworkStatus	549
9.23.2.6	unpack_qos_SLQSQosSwiReadApnExtraParams	550
9.23.2.7	unpack_qos_SLQSQosSwiReadDataStats	550
9.23.2.8	unpack_qos_SLQSSetQosEventCallback	551
9.23.2.9	unpack_qos_SLQSSetQosEventCallback_ind	551
9.23.2.10	unpack_qos_SLQSSetQosNWStatusCallback_ind	552
9.23.2.11	unpack_qos_SLQSSetQosPriEventCallback_ind	552
9.23.2.12	unpack_qos_SLQSSetQosStatusCallback_ind	553
9.24	sms.h File Reference	553
9.24.1	Macro Definition Documentation	555
9.24.1.1	MAX_CDMA_ENC_MO_TXT_MSG_SIZE	555
9.24.1.2	MAX_MS_TRANSFER_ROUTE_MSG	555
9.24.1.3	MAX_MSC_ADDRESS_SIZE	555
9.24.1.4	MAX_MSE_TWS_MSG	555
9.24.1.5	MAX_SMS_LIST_SIZE	555
9.24.1.6	MAX_SMS_MESSAGE_SIZE	555
9.24.2	Typedef Documentation	555
9.24.2.1	sMSCAddressInfo	555
9.24.2.2	sMSEtwsMessageInfo	556
9.24.2.3	sMSEtwsPlmnInfo	556
9.24.2.4	sMSMessageModelInfo	556
9.24.2.5	sMSMTMessageInfo	556
9.24.2.6	sMSOnIMSInfo	556
9.24.2.7	sMSTransferRouteMTMessageInfo	556
9.24.3	Enumeration Type Documentation	557
9.24.3.1	eqmiCbkSetStatus	557
9.24.4	Function Documentation	557

9.24.4.1	pack_sms_SendSMS	557
9.24.4.2	pack_sms_SetNewSMSCallback	557
9.24.4.3	pack_sms_SLQSDDeleteSMS	558
9.24.4.4	pack_sms_SLQSGetSMS	558
9.24.4.5	pack_sms_SLQSGetSMSList	559
9.24.4.6	pack_sms_SLQSModifySMSStatus	559
9.24.4.7	unpack_sms_SendSMS	559
9.24.4.8	unpack_sms_SetNewSMSCallback	560
9.24.4.9	unpack_sms_SetNewSMSCallback_ind	560
9.24.4.10	unpack_sms_SLQSDDeleteSMS	560
9.24.4.11	unpack_sms_SLQSGetSMS	561
9.24.4.12	unpack_sms_SLQSGetSMSList	561
9.24.4.13	unpack_sms_SLQSModifySMSStatus	561
9.24.4.14	unpack_sms_SLQSWmsMemoryFullCallBack_ind	562
9.25	SwiDataTypes.h File Reference	562
9.25.1	Detailed Description	563
9.25.2	Macro Definition Documentation	563
9.25.2.1	QMI_NO_LTE_FW_SUPPORT	563
9.25.2.2	QMI_TLV_PLACEHOLDER	563
9.25.2.3	SWI_API	563
9.25.2.4	UNUSEDPARAM	563
9.25.3	Typedef Documentation	563
9.25.3.1	BOOL	563
9.25.3.2	BYTE	563
9.25.3.3	CHAR	563
9.25.3.4	FLOAT	563
9.25.3.5	INT32	563
9.25.3.6	INT8	563
9.25.3.7	LPCSTR	563
9.25.3.8	SHORT	563
9.25.3.9	ULONG	563
9.25.3.10	ULONGLONG	563
9.25.3.11	USHORT	563
9.25.3.12	WORD	563
9.26	swiloc.h File Reference	563
9.26.1	Function Documentation	564
9.26.1.1	pack_swiloc_SwiLocGetAutoStart	564
9.26.1.2	pack_swiloc_SwiLocSetAutoStart	564
9.26.1.3	unpack_swiloc_SwiLocGetAutoStart	564
9.26.1.4	unpack_swiloc_SwiLocSetAutoStart	565

9.27	swioma.h File Reference	565
9.27.1	Macro Definition Documentation	566
9.27.1.1	LIBPACK_MAX_SWIOMA_STR_LEN	566
9.27.2	Function Documentation	566
9.27.2.1	pack_swioma_SLQSOMADMAAlertCallback	566
9.27.2.2	pack_swioma_SLQSOMADMCancelSession	567
9.27.2.3	pack_swioma_SLQSOMADMGetSessionInfo	567
9.27.2.4	pack_swioma_SLQSOMADMGetSettings	568
9.27.2.5	pack_swioma_SLQSOMADMSendSelection	569
9.27.2.6	pack_swioma_SLQSOMADMSetSettings	569
9.27.2.7	pack_swioma_SLQSOMADMStartSession	570
9.27.2.8	unpack_swioma_SLQSOMADMAAlertCallback	570
9.27.2.9	unpack_swioma_SLQSOMADMAAlertCallback_ind	571
9.27.2.10	unpack_swioma_SLQSOMADMCancelSession	571
9.27.2.11	unpack_swioma_SLQSOMADMGetSessionInfo	572
9.27.2.12	unpack_swioma_SLQSOMADMGetSettings	572
9.27.2.13	unpack_swioma_SLQSOMADMSendSelection	573
9.27.2.14	unpack_swioma_SLQSOMADMSetSettings	573
9.27.2.15	unpack_swioma_SLQSOMADMStartSession	574
9.28	SWIWWANCMAPI.h File Reference	574
9.29	uim.h File Reference	574
9.29.1	Macro Definition Documentation	576
9.29.1.1	MAX_DESCRIPTION_LENGTH	576
9.29.1.2	MAX_ICCID_LENGTH	576
9.29.1.3	MAX_NO_OF_APPLICATIONS	576
9.29.1.4	MAX_NO_OF_SLOTS	576
9.29.1.5	MAX_SLOTS_STATUS	576
9.29.1.6	UIM_MAX_DESCRIPTION_LENGTH	576
9.29.1.7	UIM_MAX_NO_OF_APPLICATIONS	576
9.29.1.8	UIM_MAX_NO_OF_SLOTS	576
9.29.1.9	UIM_UINT8_MAX_STRING_SZ	576
9.29.2	Function Documentation	576
9.29.2.1	pack_uim_ChangePin	576
9.29.2.2	pack_uim_GetCardStatus	577
9.29.2.3	pack_uim_ReadTransparent	577
9.29.2.4	pack_uim_SetPinProtection	577
9.29.2.5	pack_uim_SLQSUIEventRegister	578
9.29.2.6	pack_uim_SLQSUIGetSlotsStatus	578
9.29.2.7	pack_uim_SLQSUIPowerDown	578
9.29.2.8	pack_uim_SLQSUIPowerUp	579

9.29.2.9	pack_uim_SLQSUIMSwitchSlot	579
9.29.2.10	pack_uim_UnblockPin	580
9.29.2.11	pack_uim_VerifyPin	580
9.29.2.12	unpack_uim_ChangePin	580
9.29.2.13	unpack_uim_GetCardStatus	581
9.29.2.14	unpack_uim_ReadTransparent	581
9.29.2.15	unpack_uim_SetPinProtection	581
9.29.2.16	unpack_uim_SetUimSlotStatusChangeCallback_ind	582
9.29.2.17	unpack_uim_SLQSUIEventRegister	582
9.29.2.18	unpack_uim_SLQSUIMGetSlotsStatus	582
9.29.2.19	unpack_uim_SLQSUIPowerDown	583
9.29.2.20	unpack_uim_SLQSUIPowerUp	583
9.29.2.21	unpack_uim_SLQSUIMSetStatusChangeCallBack_ind	583
9.29.2.22	unpack_uim_SLQSUIMSwitchSlot	584
9.29.2.23	unpack_uim_UnblockPin	584
9.29.2.24	unpack_uim_VerifyPin	585
9.30	wds.h File Reference	585
9.30.1	Macro Definition Documentation	589
9.30.1.1	BYT_STAT_STAT_MASK	589
9.30.1.2	IPV6_ADDRESS_ARRAY_SIZE	589
9.30.1.3	MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	589
9.30.1.4	PACK_WDS_IPV4	589
9.30.1.5	PACK_WDS_IPV6	590
9.30.2	Typedef Documentation	590
9.30.2.1	UnpackQmiProfileInfo	590
9.30.3	Function Documentation	590
9.30.3.1	pack_wds_GetByteTotals	590
9.30.3.2	pack_wds_GetConnectionRate	590
9.30.3.3	pack_wds_GetDefaultProfile	590
9.30.3.4	pack_wds_GetDefaultProfileNum	591
9.30.3.5	pack_wds_GetDormancyState	591
9.30.3.6	pack_wds_GetLastMobileIPError	592
9.30.3.7	pack_wds_GetMobileIP	592
9.30.3.8	pack_wds_GetMobileIPProfile	593
9.30.3.9	pack_wds_GetPacketStatistics	593
9.30.3.10	pack_wds_GetPacketStatus	593
9.30.3.11	pack_wds_GetSessionDuration	594
9.30.3.12	pack_wds_GetSessionState	594
9.30.3.13	pack_wds_RMSetTransferStatistics	595
9.30.3.14	pack_wds_SetDefaultProfile	595

9.30.3.15 pack_wds_SetDefaultProfileNum	595
9.30.3.16 pack_wds_SetMobileIPProfile	596
9.30.3.17 pack_wds_SLQSCreateProfile	596
9.30.3.18 pack_wds_SLQSDeleteProfile	597
9.30.3.19 pack_wds_SLQSGet3GPPConfigItem	597
9.30.3.20 pack_wds_SLQSGetCurrDataSystemStat	597
9.30.3.21 pack_wds_SLQSGetCurrentChannelRate	598
9.30.3.22 pack_wds_SLQSGetDataBearerTechnology	598
9.30.3.23 pack_wds_SLQSGetDUNCallInfo	599
9.30.3.24 pack_wds_SLQSGetProfileSettings	599
9.30.3.25 pack_wds_SLQSGetRuntimeSettings	599
9.30.3.26 pack_wds_SLQSModifyProfile	600
9.30.3.27 pack_wds_SLQSSet3GPPConfigItem	600
9.30.3.28 pack_wds_SLQSSetIPFamilyPreference	601
9.30.3.29 pack_wds_SLQSSetWdsEventCallback	601
9.30.3.30 pack_wds_SLQSSetDHCPv4ClientConfig	601
9.30.3.31 pack_wds_SLQSSetLoopback	602
9.30.3.32 pack_wds_SLQSSetLoopback	602
9.30.3.33 pack_wds_SLQSStartDataSession	603
9.30.3.34 pack_wds_SLQSStopDataSession	603
9.30.3.35 pack_wds_SLQSWdsSwiPDPRuntimeSettings	604
9.30.3.36 unpack_wds_GetByteTotals	604
9.30.3.37 unpack_wds_GetConnectionRate	604
9.30.3.38 unpack_wds_GetDefaultProfile	605
9.30.3.39 unpack_wds_GetDefaultProfileNum	605
9.30.3.40 unpack_wds_GetDormancyState	605
9.30.3.41 unpack_wds_GetLastMobileIPError	606
9.30.3.42 unpack_wds_GetMobileIP	606
9.30.3.43 unpack_wds_GetMobileIPProfile	606
9.30.3.44 unpack_wds_GetPacketStatistics	607
9.30.3.45 unpack_wds_GetPacketStatus	607
9.30.3.46 unpack_wds_GetSessionDuration	607
9.30.3.47 unpack_wds_GetSessionState	608
9.30.3.48 unpack_wds_RMSetTransferStatistics	608
9.30.3.49 unpack_wds_SetDefaultProfile	608
9.30.3.50 unpack_wds_SetDefaultProfileNum	609
9.30.3.51 unpack_wds_SetMobileIPProfile	609
9.30.3.52 unpack_wds_SLQSCreateProfile	609
9.30.3.53 unpack_wds_SLQSDeleteProfile	610
9.30.3.54 unpack_wds_SLQSGet3GPPConfigItem	610

9.30.3.55 unpack_wds_SLQSGetCurrDataSystemStat	610
9.30.3.56 unpack_wds_SLQSGetCurrentChannelRate	611
9.30.3.57 unpack_wds_SLQSGetDataBearerTechnology	611
9.30.3.58 unpack_wds_SLQSGetDUNCallInfo	611
9.30.3.59 unpack_wds_SLQSGetProfileSettings	612
9.30.3.60 unpack_wds_SLQSGetRuntimeSettings	612
9.30.3.61 unpack_wds_SLQSModifyProfile	612
9.30.3.62 unpack_wds_SLQSSet3GPPConfigItem	613
9.30.3.63 unpack_wds_SLQSSetIPFamilyPreference	613
9.30.3.64 unpack_wds_SLQSSetPacketSrvStatusCallback	613
9.30.3.65 unpack_wds_SLQSSetWdsEventCallback	614
9.30.3.66 unpack_wds_SLQSSetWdsEventCallback_ind	614
9.30.3.67 unpack_wds_SLQSSetDHCPv4ClientConfig	614
9.30.3.68 unpack_wds_SLQSSetLoopback	615
9.30.3.69 unpack_wds_SLQSSetLoopback	615
9.30.3.70 unpack_wds_SLQSStartDataSession	615
9.30.3.71 unpack_wds_SLQSStopDataSession	616
9.30.3.72 unpack_wds_SLQSWdsSwiPDPRuntimeSettings	616

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
Streaming Download Protocol (sdp)	20

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:

_libSDP_FirmwareInfo_	23
appStats	24
CarrierImage_t	27
cdmaSSInfo	28
connectionStatus	28
currNetworkInfo	29
dms_ActivationStatusTlv	29
dms_OperatingModeTlv	30
DMScustSettingInfo	31
DMScustSettingList	31
DMSgetCustomFeatureV2	32
DMSgetCustomInput	33
dunchannelRate	33
eTWSPLMNInfoTlv	34
FMSImageElement	34
FMSImageIdElement	35
FMSImageIDEntries	36
FMSImageList	37
FMSPrefImageList	37
hdrSSInfo	38
image_info_t	38
ipv6AddressInfo	39
LibPackGPRSRequestedQoS	39
LibpackProfile3GPP	40
LibpackProfile3GPP2	46
LibPackprofile_3GPP	51
LibPackprofile_3GPP2	57
LibPackQosClassID	62
LibPackTFTIDParams	63
LibPackUMTSQoS	65
LibPackUMTSReqQoSSigInd	67
loc_BdsSV	68
loc_BdsSVInfo	68
loc_CellDb	69
loc_ClkInfo	70
loc_GnssData	71
loc_gpsTime	72
loc_LocApplicationInfo	73

loc_precisionDilution	74
loc_sensorDataUsage	75
loc_SV	76
loc_SVInfo	76
loc_svUsedforFix	77
lteSSInfo	78
messageModeTlv	78
nas_acqOrderPref	79
nas_AddCDMASysInfo	79
nas_AddSysInfo	80
nas_CallBarringSysInfo	80
nas_callBarStatus	81
nas_CDMAECIOThresh	82
nas_CDMAInfo	83
nas_CDMA RSSI Thresh	84
nas_CDMA SysInfo	84
nas_CDMA SysInfoExt	88
nas_cellParams	88
nas_CommInfo	89
nas_CSGID	91
nas_currentPLMN	92
nas_dataSrvCapabilities	93
nas_detailSvcInfo	93
nas_ecioListElement	95
nas_errorRateListElement	96
nas_GERANInfo	96
nas_geranInstInfo	98
nas_gsmCellInfo	99
nas_GSMRSSI Thresh	100
nas_GSMSrvStatusInfo	101
nas_GSMSysInfo	102
nas_HDRECIOTresh	105
nas_HDRIOThresh	105
nas_HDRRSSI Thresh	106
nas_HDR SINR Threshold	106
nas_HDR SysInfo	107
nas_infoInterFreq	109
nas_lteGsmCellInfo	110
nas_LTEInfo	111
nas_LTEInfoInterfreq	114
nas_LTEInfoIntrafreq	114
nas_LTEInfoNeighboringGSM	116
nas_LTEInfoNeighboringWCDMA	117
nas_lteRsrpInformation	118
nas_LTERS RP Thresh	118
nas_LTERS RQ Thresh	119
nas_LTERSS I Thresh	119
nas_LTESigRptConfig	120
nas_lteSnrInformation	120
nas_LTESNR Threshold	121
nas_LTESysInfo	121
nas_lteWcdmaCellInfo	124
nas_MNRInfo	125
nas_netSelectionPref	126
nas_nmrCellInfo	126
nas_PhyCaAggPcellInfo	128
nas_PhyCaAggScellDIBw	128
nas_PhyCaAggScellIndex	129

nas_PhyCaAggScellIndType	129
nas_PhyCaAggScellInfo	130
nas_qaQmi3Gpp2TimeZone	133
nas_QmiNas3GppNetworkInfo	134
nas_QmiNas3GppNetworkRAT	134
nas_QmisNasPcsDigit	135
nas_RejectReasonTlv	136
nas_RFInfoTlv	136
nas_roamIndList	137
nas_rsrqInformation	138
nas_RxSigInfo	138
nas_rxSignalStrengthListElement	139
nas_SccRxInfo	140
nas_servSystem	141
nas_SignalStrengthTlv	143
nas_SLQSSignalStrengthsIndReq	143
nas_SLQSSignalStrengthsInformation	144
nas_SLQSSignalStrengthsTlv	145
nas_SrvStatusInfo	145
nas_sysInfoCommon	146
nas_TDSCDMAECIOThresh	149
nas_TDSCDMARSCPTthresh	149
nas_TDSCDMARSSIThresh	150
nas_TDSCDMASINRThresh	150
nas_timeInfo	151
nas_UMTSInfo	152
nas_UMTSinstInfo	154
nas_umtsLTENbrCell	155
nas_UniversalTime	156
nas_wcdmaCellInfo	157
nas_WCDMAECIOThresh	158
nas_WCDMAInfoLTENeighborCell	159
nas_WCDMARSSIThresh	159
nas_WCDMASysInfo	160
NASBandPreferenceTlv	163
NASEmergencyModeTlv	164
NasGetLTECphyCaInfo	164
NASGWAcqOrderPrefTlv	164
NASLTEBandPreferenceTlv	165
NASLteNasReleaseInfoTlv	165
NASModePreferenceTlv	165
NASNetSelPreferenceTlv	165
NASOTAMessageTlv	166
NASPhyCaAggPcellInfo	166
NASPhyCaAggScellIDIBw	167
NASPhyCaAggScellIndex	167
NASPhyCaAggScellIndType	168
NASPhyCaAggScellInfo	169
NASPRLPreferenceTlv	170
NASQmiCbkNasSwiOTAMessageInd	170
NASQmiCbkNasSystemSelPrefInd	170
NASRoamPreferenceTlv	171
NASServDomainPrefTlv	171
NASServingSystemInfo	171
NASTimeInfoTlv	173
newMTMessageTlv	173
pack_dms_GetCustFeaturesV2_t	174
pack_dms_SetCrashAction_t	174

pack_dms_SetCustFeature_t	175
pack_dms_SetCustFeaturesV2_t	175
pack_dms_SetEventReport_t	176
pack_dms_SetPower_t	176
pack_dms_SetUSBComp_t	177
pack_dms_SLQSDmsSwiIndicationRegister_t	177
pack_dms_SLQSSwiSetDyingGaspCfg_t	177
pack_dms_UIMGetICCID_t	178
pack_fms_GetImagesPreference_t	178
pack_fms_GetStoredImages_t	179
pack_fms_SetImagesPreference_t	179
pack_loc_Delete_Assist_Data_t	180
pack_loc_EventRegister_t	181
pack_loc_SetExtPowerState_t	183
pack_loc_SetOperationMode_t	183
pack_loc_SLQSLOCGetBestAvailPos_t	184
pack_loc_Start_t	184
pack_loc_Stop_t	186
pack_nas_SetACCOLC_t	186
pack_nas_SetNetworkPreference_t	187
pack_nas_SLQSGetPLMNName_t	188
pack_nas_SLQSIInitiateNetworkRegistration_t	188
pack_nas_SLQSNasConfigSigInfo2_t	189
pack_nas_SLQSNasIndicationRegisterExt_t	194
pack_nas_SLQSNasSwiOTAMessageCallback_t	197
pack_nas_SLQSSetSignalStrengthsCallback_t	198
pack_nas_SLQSSetSysSelectionPref_t	198
pack_qmi_t	203
pack_qos_SLQSQosSwiReadApnExtraParams_t	203
pack_qos_SLQSQosSwiReadDataStats_t	204
pack_qos_SLQSSetQosEventCallback_t	204
pack_sms_SendSMS_t	205
pack_sms_SetNewSMSCallback_t	205
pack_sms_SLQSDeleteSMS_t	206
pack_sms_SLQSGetSMS_t	207
pack_sms_SLQSGetSMSList_t	207
pack_sms_SLQSModifySMSStatus_t	208
pack_swiloc_SwiLocSetAutoStart_t	209
pack_swioama_SLQSOMADMCancelSession_t	210
pack_swioama_SLQSOMADMGetSessionInfo_t	211
pack_swioama_SLQSOMADMSelectSelection_t	211
pack_swioama_SLQSOMADMSetSettings_t	212
pack_swioama_SLQSOMADMStartSession_t	213
pack_uim_ChangePin_t	214
pack_uim_ReadTransparent_t	215
pack_uim_SetPinProtection_t	216
pack_uim_SLQSUIEventRegister_t	217
pack_uim_SLQSUIPowerDown_t	217
pack_uim_SLQSUIPowerUp_t	217
pack_uim_SLQSUIMSwitchSlot_t	218
pack_uim_UnblockPin_t	219
pack_uim_VerifyPin_t	220
pack_wds_GetDefaultProfile_t	221
pack_wds_GetDefaultProfileNum_t	221
pack_wds_GetDormancyState_t	222
pack_wds_GetLastMobileIPError_t	222
pack_wds_GetMobileIP_t	222
pack_wds_GetMobileIPProfile_t	222

pack_wds_GetPacketStatistics_t	222
pack_wds_GetPacketStatus_t	223
pack_wds_GetSessionDuration_t	223
pack_wds_RMSetTransferStatistics_t	223
pack_wds_SetDefaultProfile_t	224
pack_wds_SetDefaultProfileNum_t	225
pack_wds_SetMobileIPProfile_t	225
pack_wds_SLQSCreateProfile_t	226
pack_wds_SLQSDeleteProfile_t	227
pack_wds_SLQSGetCurrDataSystemStat_t	227
pack_wds_SLQSGetDataBearerTechnology_t	227
pack_wds_SLQSGetDUNCallInfo_t	227
pack_wds_SLQSGetProfileSettings_t	228
pack_wds_SLQSGetRuntimeSettings_t	229
pack_wds_SLQSModifyProfile_t	229
pack_wds_SLQSSet3GPPConfigItem_t	230
pack_wds_SLQSSetIPFamilyPreference_t	232
pack_wds_SLQSSetWdsEventCallback_t	232
pack_wds_SLQSSetDHCPv4ClientConfig_t	233
pack_wds_SLQSSetLoopback_t	233
pack_wds_SLQSStartDataSession_t	234
pack_wds_SLQSStopDataSession_t	235
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	235
PackCreateProfileOut	235
packgetDyingGaspCfg	236
packgetDyingGaspStatistics	236
qmiSmsMessageList	237
qmiWDSDataBearerTechnology	237
RFBandInfoElements	238
rmTrasnferStaticsReq	238
slot_t	239
slotInf	240
slots_t	241
sMSCAddress	241
sMSCAddressTlv	242
sMSEtwsMessage	242
sMSEtwsMessageTlv	243
sMSEtwsPlmn	243
sMSMessageMode	244
sMSMTMessage	244
sMSOnIMS	245
sMSOnIMSTlv	245
sMSTransferRouteMTMessage	245
tdscdmaSigInfoExt	246
transferRouteMessageTlv	247
transferStatInd	247
uim_appStatus	248
uim_cardResult	251
uim_cardStatus	251
uim_changeUIMPIN	253
uim_encryptedPIN1	253
uim_fileInfo	254
uim_hotSwapStatus	255
uim_readResult	255
uim_readTransparentInfo	256
uim_remainingRetries	256
uim_sessionInformation	257
uim_setPINProtection	258

uim_slotInfo	259
uim_UIMSessionInformation	260
uim_unblockUIMPIN	261
uim_verifyUIMPIN	262
unpack_dms_GetActivationState_t	263
unpack_dms_GetBandCapability_t	263
unpack_dms_GetCrashAction_t	264
unpack_dms_GetCustFeature_t	264
unpack_dms_GetCustFeaturesV2_t	265
unpack_dms_GetDeviceCap_t	265
unpack_dms_GetDeviceCapabilities_t	266
unpack_dms_GetDeviceHardwareRev_t	266
unpack_dms_GetDeviceMfr_t	267
unpack_dms_GetDeviceSerialNumbers_t	267
unpack_dms_GetFirmwareInfo_t	268
unpack_dms_GetFirmwareRevision_t	269
unpack_dms_GetFirmwareRevisions_t	269
unpack_dms_GetFSN_t	270
unpack_dms_GetHardwareRevision_t	270
unpack_dms_GetIMSI_t	270
unpack_dms_GetManufacturer_t	270
unpack_dms_GetModelID_t	271
unpack_dms_GetNetworkTime_t	271
unpack_dms_GetOfflineReason_t	272
unpack_dms_GetPower_t	273
unpack_dms_GetPRLVersion_t	273
unpack_dms_GetSerialNumbers_t	274
unpack_dms_GetUSBComp_t	274
unpack_dms_GetVoiceNumber_t	275
unpack_dms_SetCrashAction_t	275
unpack_dms_SetCustFeature_t	275
unpack_dms_SetCustFeaturesV2_t	276
unpack_dms_SetEventReport_ind_t	276
unpack_dms_SetEventReport_t	277
unpack_dms_SetFirmwarePreference_t	277
unpack_dms_SetPower_t	277
unpack_dms_SetUSBComp_t	277
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t	277
unpack_dms_SLQSDmsSwiGetResetInfo_t	278
unpack_dms_SLQSDmsSwiIndicationRegister_t	279
unpack_dms_SLQSGetBandCapability_t	280
unpack_dms_SLQSSwiClearDyingGaspStatistics_t	283
unpack_dms_SLQSSwiGetDyingGaspCfg_t	283
unpack_dms_SLQSSwiGetDyingGaspStatistics_t	284
unpack_dms_SLQSSwiGetFirmwareCurr_t	284
unpack_dms_SLQSSwiGetFwUpdateStatus_t	285
unpack_dms_SLQSSwiSetDyingGaspCfg_t	287
unpack_dms_UIMGetICCID_t	287
unpack_fms_GetImagesPreference_t	288
unpack_fms_GetStoredImages_t	288
unpack_fms_SetImagesPreference_t	289
unpack_loc_BestAvailPos_Ind_t	289
unpack_loc_Delete_Assist_Data_t	295
unpack_loc_EngineState_Ind_t	296
unpack_loc_EventRegister_t	296
unpack_loc_PositionRpt_Ind_t	297
unpack_loc_SetExtPowerConfig_Ind_t	303
unpack_loc_SetExtPowerState_t	303

unpack_loc_SetOperationMode_t	304
unpack_loc_SLQSLOCGetBestAvailPos_t	304
unpack_loc_Start_t	304
unpack_loc_Stop_t	305
unpack_nas_GetCDMANetworkParameters_t	305
unpack_nas_GetHomeNetwork_t	306
unpack_nas_GetNetworkPreference_t	307
unpack_nas_GetRFInfo_t	308
unpack_nas_GetServingNetwork_t	308
unpack_nas_GetServingNetworkCapabilities_t	309
unpack_nas_GetSignalStrengths_t	310
unpack_nas_PerformNetworkScan_t	310
unpack_nas_SetDataCapabilitiesCallback_ind_t	311
unpack_nas_SetEventReportInd_t	311
unpack_nas_SetNasLTECphyCalIndCallback_ind_t	312
unpack_nas_SetNetworkPreference_t	313
unpack_nas_SetRoamingIndicatorCallback_ind_t	314
unpack_nas_SetServingSystemCallback_ind_t	314
unpack_nas_SlqsGetLTECphyCAInfo_t	314
unpack_nas_SLQSGetNetworkTime_t	315
unpack_nas_SLQSGetPLMNName_t	315
unpack_nas_SLQSGetServingSystem_t	316
unpack_nas_SLQSGetSignalStrength_t	318
unpack_nas_SLQSGetSysInfo_t	320
unpack_nas_SLQSGetSysSelectionPref_t	322
unpack_nas_SLQSNasGetCellLocationInfo_t	326
unpack_nas_SLQSNasGetSigInfo_t	328
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t	328
unpack_nas_SLQSNasSigInfoCallback_ind_t	329
unpack_nas_SLQSNasSwiModemStatus_t	330
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t	331
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t	331
unpack_nas_SLQSSwiGetLteCQI_t	331
unpack_nas_SLQSSwiGetLteSccRxInfo_t	332
unpack_nas_SLQSSysInfoCallback_ind_t	333
unpack_omaDmConfigTlv_t	335
unpack_omaDmFotaTlv_t	336
unpack_omaDmNotificationsTlv_t	338
unpack_qmi_t	339
unpack_qos_dataRate_t	339
unpack_qos_IPv4Addr_t	340
unpack_qos_IPv6Addr_t	340
unpack_qos_IPv6TrafCls_t	341
unpack_qos_pktErrRate_t	341
unpack_qos_Port_t	341
unpack_qos_QosFlowInfo_t	342
unpack_qos_QosFlowInfoState_t	343
unpack_qos_SLQSQosGetNetworkStatus_t	344
unpack_qos_SLQSQosSwiReadApnExtraParams_t	345
unpack_qos_SLQSQosSwiReadDataStats_t	346
unpack_qos_SLQSSetQosEventCallback_ind_t	347
unpack_qos_SLQSSetQosNWStatusCallback_ind_t	348
unpack_qos_SLQSSetQosPriEventCallback_ind_t	348
unpack_qos_SLQSSetQosStatusCallback_ind_t	348
unpack_qos_swiQosFilter_t	350
unpack_qos_swiQosFlow_t	353
unpack_qos_tokenBucket_t	357
unpack_qos_Tos_t	358

unpack_QosFlowStat_t	358
unpack_sms_SendSMS_t	359
unpack_sms_SetNewSMSCallback_ind_t	360
unpack_sms_SetNewSMSCallback_t	361
unpack_sms_SLQSDDeleteSMS_t	361
unpack_sms_SLQSGetSMS_t	361
unpack_sms_SLQSGetSMSList_t	362
unpack_sms_SLQSModifySMSStatus_t	362
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	362
unpack_swiloc_SwiLocGetAutoStart_t	363
unpack_swioama_SLQSOMADMAAlertCallback_ind_t	364
unpack_swioama_SLQSOMADMGetSessionInfo_t	365
unpack_swioama_SLQSOMADMGetSettings_t	368
unpack_swioama_SLQSOMADMStartSession_t	369
unpack_uim_ChangePin_t	370
unpack_uim_GetCardStatus_t	371
unpack_uim_ReadTransparent_t	371
unpack_uim_SetPinProtection_t	372
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	373
unpack_uim_SLQSUIMEventRegister_t	373
unpack_uim_SLQSUIMGetSlotsStatus_t	374
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t	374
unpack_uim_UnblockPin_t	375
unpack_uim_VerifyPin_t	375
unpack_wds_GetByteTotals_t	376
unpack_wds_GetConnectionRate_t	377
unpack_wds_GetDefaultProfile_t	377
unpack_wds_GetDefaultProfileNum_t	378
unpack_wds_GetDormancyState_t	379
unpack_wds_GetLastMobileIPError_t	379
unpack_wds_GetMobileIP_t	379
unpack_wds_GetMobileIPProfile_t	380
unpack_wds_GetPacketStatistics_t	381
unpack_wds_GetPacketStatus_t	382
unpack_wds_GetSessionDuration_t	383
unpack_wds_GetSessionState_t	384
unpack_wds_RMSetTransferStatistics_t	384
unpack_wds_SetMobileIPProfile_t	384
unpack_wds_SLQSCreateProfile_t	384
unpack_wds_SLQSDDeleteProfile_t	385
unpack_wds_SLQSGet3GPPConfigItem_t	385
unpack_wds_SLQSGetCurrDataSystemStat_t	386
unpack_wds_SLQSGetCurrentChannelRate_t	387
unpack_wds_SLQSGetDataBearerTechnology_t	388
unpack_wds_SLQSGetDUNCallInfo_t	388
unpack_wds_SLQSGetProfileSettings_t	389
unpack_wds_SLQSGetRuntimeSettings_t	390
unpack_wds_SLQSModifyProfile_t	392
unpack_wds_SLQSSetIPFamilyPreference_t	392
unpack_wds_SLQSSetPacketSrvStatusCallback_t	392
unpack_wds_SLQSSetWdsEventCallback_ind_t	393
unpack_wds_SLQSSGetDHCPv4ClientConfig_t	395
unpack_wds_SLQSSGetLoopback_t	395
unpack_wds_SLQSSStartDataSession_t	396
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	397
UnPackGetProfileSettingOut	398
unpackWdsProfileParam	398
wds_currNetworkInfo	399

wds_Domain	400
wds_DomainNameList	401
wds_GPRSQoS	401
wds_IPV6AddressInfo	402
wds_IPV6GWAddressInfo	402
wds_PCSCFFQDNAddress	403
wds_PCSCFFQDNAddressList	403
wds_PCSCFIPv4ServerAddressList	404
wds_ProfileIdentifier	404
wds_profileInfo	405
wds_UMTSMInQoS	405
wdsDhcpv4HwConfig	408
wdsDhcpv4Option	408
wdsDhcpv4OptionList	409
wdsDhcpv4ProfileId	409

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	411
common.h		411
dms.h		415
fms.h		451
libsdp.h		454
loc.h		461
nas.h		472
qaGobiApiTableBandClasses.h	Network Access Service API Band Classes table	500
qaGobiApiTableCallControlReturnReasons.h	Call Control Return Reasons table	503
qaGobiApiTableCallEndReasons.h	Wireless Data Service Call End Reasons	504
qaGobiApiTableCarrierCodes.h	Carrier Codes table	520
qaGobiApiTableCodingScheme.h	Data Coding Scheme	522
qaGobiApiTableGpsCapabilityCodes.h	Position Determination Service API GPS Capability Codes	524
qaGobiApiTablePowerModes.h	Device Management Service API Power Modes table	525
qaGobiApiTableRadioInterfaces.h	Network Access Service API Radio Interfaces table	526
qaGobiApiTableRegionCodes.h	Region Codes table	526
qaGobiApiTableServiceOptions.h	Voice Service Options	527
qaGobiApiTableSupServiceInfoClasses.h	Voice Supplementary Service Information Classes	529
qaGobiApiTableSwiAudio.h	Swi Audio related tables	529
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h	Update Complete Status table	530
qaGobiApiTableVoiceCallEndReasons.h	Voice Service Call and supplementary services end reasons	531
qmerrno.h		538

qos.h	545
sms.h	553
SwiDataTypes.h	
SWI data types	562
swiloc.h	563
swioma.h	565
SWIWWANCMAPI.h	574
uim.h	574
wds.h	585

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

6.2 Streaming Download Protocol (sdp)

Files

- file [libsdp.h](#)

6.2.1 Detailed Description

Chapter 7

Namespace Documentation

7.1 Tables Namespace Reference

7.1.1 Detailed Description

[Tables](#) referenced in the API function headers:

- Table 1 - Call End Reason Codes The reason a call (either in process or connected) was ended. [qaGobiApiTableCallEndReasons.h](#)
- Table 2 - Carrier codes List of carrier identification codes. [qaGobiApiTableCarrierCodes.h](#)
- Table 3 - Region codes List of region identification codes. [qaGobiApiTableRegionCodes.h](#)
- Table 4 - GPS capability codes List of GPS capability codes. [qaGobiApiTableGpsCapabilityCodes.h](#)
- Table 5 - Radio Interfaces List of radio interface technologies. [qaGobiApiTableRadioInterfaces.h](#)
- Table 6 - Band classes List of band classes. [qaGobiApiTableBandClasses.h](#)
- Table 7 - Power modes List of operating modes. [qaGobiApiTablePowerModes.h](#)
- Table 8 - Service Options List of Service Options. [qaGobiApiTableServiceOptions.h](#)
- Table 9 - Voice Call End Reason List of Voice Call End Reason. [qaGobiApiTableVoiceCallEndReasons.h](#)
- Table 10 - Data Coding Scheme List of Data Coding Scheme. [qaGobiApiTableCodingScheme.h](#)
- Table 11 - Call Control Return Reasons List of Voice Call Control Return Reasons. [qaGobiApiTableCallControlReturnReasons.h](#)
- Table 12 - Supplementary Service Information Classes List of Voice Supplementary Service Information Classes. [qaGobiApiTableSupServiceInfoClasses.h](#)
- Table 13 - Audio Calibration Data Base(ACDB) Device List of ACDB devices. [qaGobiApiTableSwiAudio.h](#)
- Table 14 - PIFACE List of physical interfaces. [qaGobiApiTableSwiAudio.h](#)

Chapter 8

Data Structure Documentation

8.1 _libSDP_FirmwareInfo_ Struct Reference

Data Fields

- char [szModelid_str](#) [20]
- char [szFwversion_str](#) [20]
- char [szSku_str](#) [20]
- char [szPackageid_str](#) [20]
- char [szCarrier_str](#) [20]
- char [szCarrierPriversion_str](#) [20]

8.1.1 Detailed Description

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

Parameters

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

8.1.2 Field Documentation

8.1.2.1 `char _libSDP_FirmwareInfo_::szCarrier_str[20]`

8.1.2.2 `char _libSDP_FirmwareInfo_::szCarrierPriversion_str[20]`

8.1.2.3 `char _libSDP_FirmwareInfo_::szFwversion_str[20]`

8.1.2.4 `char _libSDP_FirmwareInfo_::szModelid_str[20]`

8.1.2.5 `char _libSDP_FirmwareInfo_::szPackageid_str[20]`

8.1.2.6 `char _libSDP_FirmwareInfo_::szSku_str[20]`

8.2 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`

8.2.1 Detailed Description

This structure contains Application Status Information loaded on the card.

Parameters

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

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

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

8.2.2 Field Documentation

8.2.2.1 `uint8_t appStats::aidLength`

8.2.2.2 `uint8_t appStats::aidVal[255]`

8.2.2.3 `uint8_t appStats::appState`

8.2.2.4 `uint8_t appStats::appType`

8.2.2.5 `uint8_t appStats::persoFeature`

8.2.2.6 `uint8_t appStats::persoRetries`

8.2.2.7 `uint8_t appStats::persoState`

8.2.2.8 uint8_t appStats::persoUnblockRetries

8.2.2.9 uint8_t appStats::pin1Retries

8.2.2.10 uint8_t appStats::pin1State

8.2.2.11 uint8_t appStats::pin2Retries

8.2.2.12 uint8_t appStats::pin2State

8.2.2.13 uint8_t appStats::puk1Retries

8.2.2.14 uint8_t appStats::puk2Retries

8.2.2.15 uint8_t appStats::univPin

8.3 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]

8.3.1 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_PrImageld</i>	<ul style="list-style-type: none"> • PRI image ID
<i>m_PriBuildId</i>	<ul style="list-style-type: none"> • PRI build ID

8.3.2 Field Documentation

8.3.2.1 `uint8_t CarrierImage_t::m_FwBuildId[32]`

8.3.2.2 `uint8_t CarrierImage_t::m_FwImageld[16]`

8.3.2.3 `uint32_t CarrierImage_t::m_nCarrierId`

8.3.2.4 `uint32_t CarrierImage_t::m_nFolderId`

8.3.2.5 `uint32_t CarrierImage_t::m_nStorage`

8.3.2.6 `uint8_t CarrierImage_t::m_PriBuildId[32]`

8.3.2.7 `uint8_t CarrierImage_t::m_PrImageld[16]`

8.4 cdmaSSInfo Struct Reference

Data Fields

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

8.4.1 Detailed Description

Parameters

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

8.4.2 Field Documentation

8.4.2.1 `int16_t cdmaSSInfo::ecio`

8.4.2.2 `int8_t cdmaSSInfo::rssi`

8.5 connectionStatus Struct Reference

Data Fields

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

8.5.1 Detailed Description

Parameters

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

8.5.2 Field Documentation

8.5.2.1 uint64_t connectionStatus::MDMCallDuration

8.5.2.2 uint8_t connectionStatus::MDMConnStatus

8.6 currNetworkInfo Struct Reference

Data Fields

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

8.6.1 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.

8.6.2 Field Documentation

8.6.2.1 uint8_t currNetworkInfo::NetworkType

8.6.2.2 uint32_t currNetworkInfo::RATMask

8.6.2.3 uint32_t currNetworkInfo::SOMask

8.7 dms_ActivationStatusTlv Struct Reference

Data Fields

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

8.7.1 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
-------------------------	--

8.7.2 Field Documentation

8.7.2.1 uint32_t dms_ActivationStatusTlv::activationStatus

8.7.2.2 uint16_t dms_ActivationStatusTlv::TlvPresent

8.8 dms_OperatingModeTlv Struct Reference

Data Fields

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

8.8.1 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
----------------------	--

8.8.2 Field Documentation

8.8.2.1 uint32_t dms_OperatingModeTlv::operatingMode

8.8.2.2 uint16_t dms_OperatingModeTlv::TlvPresent

8.9 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](#)

8.9.1 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

8.9.2 Field Documentation

8.9.2.1 uint16_t DMScustSettingInfo::cust_attr

8.9.2.2 uint8_t DMScustSettingInfo::cust_id[64+1]

8.9.2.3 uint8_t DMScustSettingInfo::cust_value[8+1]

8.9.2.4 uint16_t DMScustSettingInfo::id_length

8.9.2.5 uint16_t DMScustSettingInfo::value_length

8.10 DMScustSettingList Struct Reference

Data Fields

- [uint8_t list_type](#)
- [uint16_t num_instances](#)
- [DMScustSettingInfo custSetting](#) [255+1]

8.10.1 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

8.10.2 Field Documentation

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

8.10.2.2 [uint8_t](#) [DMScustSettingList::list_type](#)

8.10.2.3 [uint16_t](#) [DMScustSettingList::num_instances](#)

8.11 DMSgetCustomFeatureV2 Struct Reference

Data Fields

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

8.11.1 Detailed Description

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

Parameters

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

8.11.2 Field Documentation

8.11.2.1 **DMScustSettingInfo*** [DMSgetCustomFeatureV2::pCustSettingInfo](#)

8.11.2.2 **DMScustSettingList*** [DMSgetCustomFeatureV2::pCustSettingList](#)

8.11.2.3 **DMSgetCustomInput*** [DMSgetCustomFeatureV2::pGetCustomInput](#)

8.12 DMSgetCustomInput Struct Reference

Data Fields

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

8.12.1 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

8.12.2 Field Documentation

8.12.2.1 [uint8_t DMSgetCustomInput::cust_id](#)[64+1]

8.12.2.2 [uint8_t DMSgetCustomInput::list_type](#)

8.13 dunchannelRate Struct Reference

Data Fields

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

8.13.1 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

8.13.2 Field Documentation

8.13.2.1 `uint32_t dunchannelRate::CurrChanRxRate`

8.13.2.2 `uint32_t dunchannelRate::CurrChanTxRate`

8.13.2.3 `uint32_t dunchannelRate::MaxChanRxRate`

8.13.2.4 `uint32_t dunchannelRate::MaxChanTxRate`

8.14 eTWSPLMNInfoTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSEtwSPlmnInfo ETWSPLMNInfo`

8.14.1 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

8.14.2 Field Documentation

8.14.2.1 `sMSEtwSPlmnInfo eTWSPLMNInfoTlv::ETWSPLMNInfo`

8.14.2.2 `uint8_t eTWSPLMNInfoTlv::TlvPresent`

8.15 FMSImageElement Struct Reference

Data Fields

- `uint8_t imageType`
- `uint8_t imageId` [16]
- `uint8_t buildIdLength`
- `uint8_t buildId` [100]

8.15.1 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

8.15.2 Field Documentation

8.15.2.1 `uint8_t FMSImageElement::buildId[100]`

8.15.2.2 `uint8_t FMSImageElement::buildIdLength`

8.15.2.3 `uint8_t FMSImageElement::imageId[16]`

8.15.2.4 `uint8_t FMSImageElement::imageType`

8.16 FMSImageIdElement Struct Reference

Data Fields

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

8.16.1 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)

8.16.2 Field Documentation

8.16.2.1 `uint8_t FMSImageIdElement::buildID[100]`

8.16.2.2 `uint8_t FMSImageIdElement::buildIDLength`

8.16.2.3 `uint8_t FMSImageIdElement::failureCount`

8.16.2.4 `uint8_t FMSImageIdElement::imageID[16]`

8.16.2.5 `uint8_t FMSImageIdElement::storageIndex`

8.17 FMSImageIDEntries Struct Reference

Data Fields

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

8.17.1 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

8.17.2 Field Documentation

8.17.2.1 `uint8_t FMSImageIDEntries::executingImage`

8.17.2.2 `FMSImageIdElement FMSImageIDEntries::imageIDElement[50]`

8.17.2.3 `uint8_t FMSPrefImageList::imageIDSize`8.17.2.4 `uint8_t FMSPrefImageList::imageType`8.17.2.5 `uint8_t FMSPrefImageList::maxImages`

8.18 FMSPrefImageList Struct Reference

Data Fields

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

8.18.1 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)

8.18.2 Field Documentation

8.18.2.1 `FMSPrefImageListEntry FMSPrefImageList::imageIDEntries[2]`8.18.2.2 `uint8_t FMSPrefImageList::listSize`

8.19 FMSPrefImageList Struct Reference

Data Fields

- `uint8_t listSize`
- `FMSPrefImageListEntry listEntries [2]`

8.19.1 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>FMSPrefImageListEntry</code>

8.19.2 Field Documentation

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

8.19.2.2 **uint8_t** FMSPrefImageList::listSize

8.20 hdrSSInfo Struct Reference

Data Fields

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

8.20.1 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.

8.20.2 Field Documentation

8.20.2.1 **int16_t** hdrSSInfo::ecio

8.20.2.2 **int32_t** hdrSSInfo::io

8.20.2.3 **int8_t** hdrSSInfo::rssi

8.20.2.4 **uint8_t** hdrSSInfo::sinr

8.21 image_info_t Struct Reference

Data Fields

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

8.21.1 Field Documentation

8.21.1.1 **uint8_t** image_info_t::buildID[255]

8.21.1.2 **uint8_t** image_info_t::buildIDLen

8.21.1.3 **uint8_t** image_info_t::imageType

8.21.1.4 **uint8_t** image_info_t::uniqueID[16]

8.22 ipv6AddressInfo Struct Reference

Data Fields

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

8.22.1 Detailed Description

Parameters

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

8.22.2 Field Documentation

8.22.2.1 uint16_t ipv6AddressInfo::IPAddressV6[8]

8.22.2.2 uint8_t ipv6AddressInfo::IPV6PrefixLen

8.23 LibPackGPRSRequestedQoS Struct Reference

Data Fields

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

8.23.1 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

8.23.2 Field Documentation

- 8.23.2.1 `uint32_t LibPackGPRSRequestedQoS::delayClass`
- 8.23.2.2 `uint32_t LibPackGPRSRequestedQoS::meanThroughputClass`
- 8.23.2.3 `uint32_t LibPackGPRSRequestedQoS::peakThroughputClass`
- 8.23.2.4 `uint32_t LibPackGPRSRequestedQoS::precedenceClass`
- 8.23.2.5 `uint32_t LibPackGPRSRequestedQoS::reliabilityClass`

8.24 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 * pGPRSMInQoS`
- `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`

8.24.1 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().

8.24.2 Field Documentation

8.24.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

8.24.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

8.24.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

8.24.2.4 uint8_t* LibpackProfile3GPP::pAPNName

8.24.2.5 uint16_t* LibpackProfile3GPP::pAPNnameSize

8.24.2.6 uint8_t* LibpackProfile3GPP::pAuthenticationPref

8.24.2.7 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSMinimumQoS

8.24.2.8 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSRequestedQoS

- 8.24.2.9 `uint8_t*` LibpackProfile3GPP::plmCnFlag
- 8.24.2.10 `uint32_t*` LibpackProfile3GPP::pIPv4AddrPref
- 8.24.2.11 `uint16_t*` LibpackProfile3GPP::pIPv6AddrPref
- 8.24.2.12 `uint8_t*` LibpackProfile3GPP::pPassword
- 8.24.2.13 `uint16_t*` LibpackProfile3GPP::pPasswordSize
- 8.24.2.14 `uint8_t*` LibpackProfile3GPP::pPcscfAddrUsingDhcp
- 8.24.2.15 `uint8_t*` LibpackProfile3GPP::pPcscfAddrUsingPCO
- 8.24.2.16 `uint32_t*` LibpackProfile3GPP::pPDNInactivTimeout
- 8.24.2.17 `uint8_t*` LibpackProfile3GPP::pPdpAccessConFlag
- 8.24.2.18 `uint8_t*` LibpackProfile3GPP::pPdpContext
- 8.24.2.19 `uint8_t*` LibpackProfile3GPP::pPdpDataCompType
- 8.24.2.20 `uint8_t*` LibpackProfile3GPP::pPdpHdrCompType
- 8.24.2.21 `uint8_t*` LibpackProfile3GPP::pPDPTtype
- 8.24.2.22 `uint32_t*` LibpackProfile3GPP::pPriDNSIPv4AddPref
- 8.24.2.23 `uint16_t*` LibpackProfile3GPP::pPriDNSIPv6addpref
- 8.24.2.24 `uint8_t*` LibpackProfile3GPP::pPrimaryID
- 8.24.2.25 `uint8_t*` LibpackProfile3GPP::pProfilename
- 8.24.2.26 `uint16_t*` LibpackProfile3GPP::pProfilenameSize
- 8.24.2.27 `LibPackQosClassID*` LibpackProfile3GPP::pQosClassID
- 8.24.2.28 `uint32_t*` LibpackProfile3GPP::pSecDNSIPv4AddPref
- 8.24.2.29 `uint16_t*` LibpackProfile3GPP::pSecDNSIPv6addpref
- 8.24.2.30 `uint8_t*` LibpackProfile3GPP::pSecondaryFlag
- 8.24.2.31 `LibPackTFTIDParams*` LibpackProfile3GPP::pTFTID1Params
- 8.24.2.32 `LibPackTFTIDParams*` LibpackProfile3GPP::pTFTID2Params
- 8.24.2.33 `LibPackUMTSQoS*` LibpackProfile3GPP::pUMTSMinQoS
- 8.24.2.34 `LibPackUMTSReqQoSSigInd*` LibpackProfile3GPP::pUMTSMinQoSSigInd
- 8.24.2.35 `LibPackUMTSQoS*` LibpackProfile3GPP::pUMTSReqQoS
- 8.24.2.36 `LibPackUMTSReqQoSSigInd*` LibpackProfile3GPP::pUMTSReqQoSsigInd

8.24.2.37 uint8_t* LibpackProfile3GPP::pUsername

8.24.2.38 uint16_t* LibpackProfile3GPP::pUsernameSize

8.25 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 * [pIsPcsfAddressNedded](#)
- uint32_t * [pPrimaryV4DnsAddress](#)
- uint32_t * [pSecondaryV4DnsAddress](#)
- uint16_t * [pPriV6DnsAddress](#)
- uint16_t * [pSecV6DnsAddress](#)
- uint8_t * [pRATType](#)
- uint8_t * [pAPNEnabled3GPP2](#)
- uint32_t * [pPDNIactivTimeout3GPP2](#)
- uint8_t * [pAPNClass3GPP2](#)

8.25.1 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>pNegoDnsSrvrPref</i>	<ul style="list-style-type: none"> • Negotiate DNS Server Preference <ul style="list-style-type: none"> – 1 - (TRUE)implies request DNS addresses from the PDSN – 0 - (FALSE)implies do not request DNS addresses from the PDSN – Default value is 1 (TRUE)
-------------------------	--

<i>pPppSessClose- TimerDO</i>	<ul style="list-style-type: none"> • PPP Session Close Timer for DO <ul style="list-style-type: none"> – Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down
<i>pPppSessClose- Timer1x</i>	<ul style="list-style-type: none"> • PPP Session Close Timer for 1X <ul style="list-style-type: none"> – Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down
<i>pAllowLinger</i>	<ul style="list-style-type: none"> • Allow/disallow lingering of interface <ul style="list-style-type: none"> – 1 -(TRUE) implies allow lingering – 0 -(FALSE) implies do not allow lingering
<i>pLcpAckTimeout</i>	<ul style="list-style-type: none"> • LCP ACK Timeout <ul style="list-style-type: none"> – Value of LCP ACK Timeout in milliseconds
<i>plpcpAck- Timeout</i>	<ul style="list-style-type: none"> • IPCP ACK Timeout <ul style="list-style-type: none"> – Value of IPCP ACK Timeout in milliseconds
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> • AUTH Timeout <ul style="list-style-type: none"> – Value of Authentication Timeout in milliseconds
<i>pLcpCreqRetry- Count</i>	<ul style="list-style-type: none"> • LCP Configuration Request Retry Count
<i>plpcpCreqRetry- Count</i>	<ul style="list-style-type: none"> • IPCP Configuration Request Retry Count
<i>pAuthRetry- Count</i>	<ul style="list-style-type: none"> • Authentication Retry Count value
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> • Authentication Protocol <ul style="list-style-type: none"> – 1 - PAP – 2 - CHAP – 3 - PAP or CHAP
<i>pUserId</i>	<ul style="list-style-type: none"> • User ID to be used during data network authentication • maximum length allowed is 127 uint8_ts; • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
<i>pUserIdSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 uint8_ts.

<i>pAuthPassword</i>	<ul style="list-style-type: none"> • Password to be used during data network authentication; • maximum length allowed is 127 uint8_ts • QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.
<i>pAuthPassword-Size;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 uint8_ts.
<i>pDataRate</i>	<ul style="list-style-type: none"> • Data Rate Requested <ul style="list-style-type: none"> – 0 - Low (Low speed Service Options (SO15) only) – 1 - Medium (SO33 + low R-SCH) – 2 - High (SO33 + high R-SCH) – Default is 2
<i>pAppType</i>	<ul style="list-style-type: none"> • Application Type: <ul style="list-style-type: none"> – 0x00000001 - Default Application Type – 0x00000020 - LBS Application Type – 0x00000040 - Tethered Application Type – This parameter is not used while creating/modifying a profile
<i>pDataMode</i>	<ul style="list-style-type: none"> • Data Mode to use: <ul style="list-style-type: none"> – 0 - CDMA or HDR (Hybrid 1X/1xEV-DO) – 1 - CDMA Only (1X only) – 2 - HDR Only (1xEV-DO only) – Default is 0
<i>pAppPriority</i>	<ul style="list-style-type: none"> • Application Priority <ul style="list-style-type: none"> – Numerical 1 uint8_t value defining the application priority; higher value implies higher priority – This parameter is not used while creating/modifying a profile
<i>pApnString</i>	<ul style="list-style-type: none"> • String representing the Access Point Name • maximum length allowed is 100 uint8_ts • QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.
<i>pApnStringSize;</i>	<ul style="list-style-type: none"> • This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 uint8_ts.

<i>pPdnType</i>	<ul style="list-style-type: none"> • Packed Data Network Type Requested: <ul style="list-style-type: none"> – 0 - IPv4 PDN Type – 1 - IPv6 PDN Type – 2 - IPv4 or IPv6 PDN Type – 3 - Unspecified PDN Type (implying no preference)
<i>pIsPcscf-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().

8.25.2 Field Documentation

8.25.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

8.25.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

8.25.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

8.25.2.4 uint8_t* LibpackProfile3GPP2::pApnString

8.25.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

8.25.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

8.25.2.7 uint32_t* LibpackProfile3GPP2::pAppType

8.25.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

8.25.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

8.25.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

8.25.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

8.25.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

8.25.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

8.25.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

8.25.2.15 uint16_t* LibpackProfile3GPP2::plpcpAckTimeout

8.25.2.16 uint8_t* LibpackProfile3GPP2::plpcpCreqRetryCount

8.25.2.17 uint8_t* LibpackProfile3GPP2::plsPcscfAddressNedded

8.25.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

- 8.25.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount
- 8.25.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref
- 8.25.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2
- 8.25.2.22 uint8_t* LibpackProfile3GPP2::pPdnType
- 8.25.2.23 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimer1x
- 8.25.2.24 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimerDO
- 8.25.2.25 uint32_t* LibpackProfile3GPP2::pPrimaryV4DnsAddress
- 8.25.2.26 uint16_t* LibpackProfile3GPP2::pPriV6DnsAddress
- 8.25.2.27 uint8_t* LibpackProfile3GPP2::pRATType
- 8.25.2.28 uint32_t* LibpackProfile3GPP2::pSecondaryV4DnsAddress
- 8.25.2.29 uint16_t* LibpackProfile3GPP2::pSecV6DnsAddress
- 8.25.2.30 uint8_t* LibpackProfile3GPP2::pUserId
- 8.25.2.31 uint16_t* LibpackProfile3GPP2::pUserIdSize

8.26 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](#)
- [LibPackUMTSReqQoS](#) * [pUMTSReqQoSSigInd](#)
- [LibPackUMTSReqQoS](#) * [pUMTSMinQoS](#) * [pUMTSMQoS](#) * [pUMTSMQoS](#)
- uint16_t * [pPriDNSIPv6addpref](#)
- uint16_t * [pSecDNSIPv6addpref](#)
- uint8_t * [pAddrAllocPref](#)
- [LibPackQoSClassID](#) * [pQoSClassID](#)
- uint8_t * [pAPNDisabledFlag](#)
- uint32_t * [pPDNInactivTimeout](#)
- uint8_t * [pAPNClass](#)

8.26.1 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().

8.26.2 Field Documentation

8.26.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.26.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.26.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.26.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

- 8.26.2.5 `uint16_t*` `LibPackprofile_3GPP::pAPNnameSize`
- 8.26.2.6 `uint8_t*` `LibPackprofile_3GPP::pAuthenticationPref`
- 8.26.2.7 `LibPackGPRSRequestedQoS*` `LibPackprofile_3GPP::pGPRSMinimumQoS`
- 8.26.2.8 `LibPackGPRSRequestedQoS*` `LibPackprofile_3GPP::pGPRSRequestedQoS`
- 8.26.2.9 `uint8_t*` `LibPackprofile_3GPP::pImCnFlag`
- 8.26.2.10 `uint32_t*` `LibPackprofile_3GPP::pIPv4AddrPref`
- 8.26.2.11 `uint16_t*` `LibPackprofile_3GPP::pIPv6AddPref`
- 8.26.2.12 `uint8_t*` `LibPackprofile_3GPP::pPassword`
- 8.26.2.13 `uint16_t*` `LibPackprofile_3GPP::pPasswordSize`
- 8.26.2.14 `uint8_t*` `LibPackprofile_3GPP::pPcsfAddrUsingDhcp`
- 8.26.2.15 `uint8_t*` `LibPackprofile_3GPP::pPcsfAddrUsingPCO`
- 8.26.2.16 `uint32_t*` `LibPackprofile_3GPP::pPDNInactivTimeout`
- 8.26.2.17 `uint8_t*` `LibPackprofile_3GPP::pPdpAccessConFlag`
- 8.26.2.18 `uint8_t*` `LibPackprofile_3GPP::pPdpContext`
- 8.26.2.19 `uint8_t*` `LibPackprofile_3GPP::pPdpDataCompType`
- 8.26.2.20 `uint8_t*` `LibPackprofile_3GPP::pPdpHdrCompType`
- 8.26.2.21 `uint8_t*` `LibPackprofile_3GPP::pPDPTtype`
- 8.26.2.22 `uint32_t*` `LibPackprofile_3GPP::pPriDNSIPv4AddPref`
- 8.26.2.23 `uint16_t*` `LibPackprofile_3GPP::pPriDNSIPv6addpref`
- 8.26.2.24 `uint8_t*` `LibPackprofile_3GPP::pPrimaryID`
- 8.26.2.25 `uint8_t*` `LibPackprofile_3GPP::pProfileName`
- 8.26.2.26 `uint16_t*` `LibPackprofile_3GPP::pProfileNameSize`
- 8.26.2.27 `LibPackQosClassID*` `LibPackprofile_3GPP::pQosClassID`
- 8.26.2.28 `uint32_t*` `LibPackprofile_3GPP::pSecDNSIPv4AddPref`
- 8.26.2.29 `uint16_t*` `LibPackprofile_3GPP::pSecDNSIPv6addpref`
- 8.26.2.30 `uint8_t*` `LibPackprofile_3GPP::pSecondaryFlag`
- 8.26.2.31 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID1Params`
- 8.26.2.32 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID2Params`

8.26.2.33 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSMinQoS

8.26.2.34 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSMinQoSsigInd

8.26.2.35 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSReqQoS

8.26.2.36 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSReqQoSSigInd

8.26.2.37 uint8_t* LibPackprofile_3GPP::pUsername

8.26.2.38 uint16_t* LibPackprofile_3GPP::pUsernameSize

8.27 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 * pPDNInactiveTimeout3GPP2
- uint8_t * pAPNClass3GPP2

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

8.27.2 Field Documentation

8.27.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

8.27.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

8.27.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

8.27.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

8.27.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

8.27.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

8.27.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

8.27.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

8.27.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

8.27.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

8.27.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

8.27.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout

8.27.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode

8.27.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate

8.27.2.15 uint16_t* LibPackprofile_3GPP2::plpcpAckTimeout

8.27.2.16 uint8_t* LibPackprofile_3GPP2::plpcpCreqRetryCount

8.27.2.17 uint8_t* LibPackprofile_3GPP2::plsPcscfAddressNedded

8.27.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout

- 8.27.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount
- 8.27.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref
- 8.27.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2
- 8.27.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType
- 8.27.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x
- 8.27.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO
- 8.27.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress
- 8.27.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress
- 8.27.2.27 uint8_t* LibPackprofile_3GPP2::pRATType
- 8.27.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress
- 8.27.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress
- 8.27.2.30 uint8_t* LibPackprofile_3GPP2::pUserId
- 8.27.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

8.28 LibPackQosClassID Struct Reference

Data Fields

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

8.28.1 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

8.28.2 Field Documentation

8.28.2.1 `uint8_t LibPackQosClassID::gDIBitRate`

8.28.2.2 `uint32_t LibPackQosClassID::gUIBitRate`

8.28.2.3 `uint32_t LibPackQosClassID::maxDIBitRate`

8.28.2.4 `uint32_t LibPackQosClassID::maxUIBitRate`

8.28.2.5 `uint8_t LibPackQosClassID::QCI`

8.29 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`

8.29.1 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

8.29.2 Field Documentation

8.29.2.1 uint16_t LibPackTFTIDParams::destPortRangeEnd

8.29.2.2 uint32_t LibPackTFTIDParams::destPortRangeStart

8.29.2.3 uint8_t LibPackTFTIDParams::eValid

8.29.2.4 uint8_t LibPackTFTIDParams::filterId

8.29.2.5 uint32_t LibPackTFTIDParams::flowLabel

8.29.2.6 uint32_t LibPackTFTIDParams::IPSECSPi

8.29.2.7 `uint8_t` LibPackTFTIDParams::ipVersion

8.29.2.8 `uint8_t` LibPackTFTIDParams::nextHeader

8.29.2.9 `uint16_t*` LibPackTFTIDParams::pSourceIP

8.29.2.10 `uint8_t` LibPackTFTIDParams::sourceIPMask

8.29.2.11 `uint16_t` LibPackTFTIDParams::srcPortRangeEnd

8.29.2.12 `uint16_t` LibPackTFTIDParams::srcPortRangeStart

8.29.2.13 `uint16_t` LibPackTFTIDParams::tosMask

8.30 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](#)

8.30.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

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

Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> • 0x00 - Subscribed • 0x01 - Conversational • 0x02 - Streaming • 0x03 - Interactive • 0x04 - Background
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> • Maximum uplink bit rate in bits/sec
<i>maxDownlink- Bitrate</i>	<ul style="list-style-type: none"> • Maximum downlink bit rate in bits/sec

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

<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.

8.30.2 Field Documentation

8.30.2.1 `uint8_t LibPackUMTSQoS::deliveryErrSDU`

8.30.2.2 `uint32_t LibPackUMTSQoS::grntDownlinkBitrate`

8.30.2.3 `uint32_t LibPackUMTSQoS::grntUplinkBitrate`

8.30.2.4 `uint32_t LibPackUMTSQoS::maxDownlinkBitrate`

8.30.2.5 `uint32_t LibPackUMTSQoS::maxSDUSize`

8.30.2.6 `uint32_t LibPackUMTSQoS::maxUplinkBitrate`

8.30.2.7 `uint8_t LibPackUMTSQoS::qosDeliveryOrder`

8.30.2.8 `uint8_t LibPackUMTSQoS::resBerRatio`

8.30.2.9 `uint8_t LibPackUMTSQoS::sduErrorRatio`

8.30.2.10 `uint8_t LibPackUMTSQoS::trafficClass`

8.30.2.11 `uint32_t LibPackUMTSQoS::trafficPriority`

8.30.2.12 `uint32_t LibPackUMTSQoS::transferDelay`

8.31 LibPackUMTSReqQoSsigInd Struct Reference

Data Fields

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

8.31.1 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>	- Signaling Indication flag <ul style="list-style-type: none"> • TRUE - Signaling indication ON • FALSE - Signaling indication OFF
---------------	--

8.31.2 Field Documentation

8.31.2.1 `uint8_t LibPackUMTSReqQoS::SigInd`

8.31.2.2 `LibPackUMTSReqQoS LibPackUMTSReqQoS::UMTSReqQoS`

8.32 loc_BdsSV Struct Reference

Data Fields

- `uint16_t id`
- `uint8_t mask`

8.32.1 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

8.32.2 Field Documentation

8.32.2.1 `uint16_t loc_BdsSV::id`

8.32.2.2 `uint8_t loc_BdsSV::mask`

8.33 loc_BdsSVInfo Struct Reference

Data Fields

- `uint8_t len`
- `loc_BdsSV * pSV`

8.33.1 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

8.33.2 Field Documentation

8.33.2.1 `uint8_t loc_BdsSVInfo::len`

8.33.2.2 `loc_BdsSV* loc_BdsSVInfo::pSV`

8.34 loc_CellDb Struct Reference

Data Fields

- `uint32_t mask`

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

8.34.2 Field Documentation

8.34.2.1 `uint32_t loc_CellDb::mask`

8.35 loc_ClkInfo Struct Reference

Data Fields

- uint32_t [mask](#)

8.35.1 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
-------------	--

8.35.2 Field Documentation

8.35.2.1 `uint32_t loc_ClkInfo::mask`

8.36 loc_GnssData Struct Reference

Data Fields

- `uint64_t` [mask](#)

8.36.1 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
-------------	---

8.36.2 Field Documentation

8.36.2.1 uint64_t loc_GnssData::mask

8.37 loc_gpsTime Struct Reference

Data Fields

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

8.37.1 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

8.37.2 Field Documentation

8.37.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

8.37.2.2 uint16_t loc_gpsTime::gpsWeek

8.38 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](#)

8.38.1 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

8.38.2 Field Documentation

8.38.2.1 `uint8_t loc_LocApplicationInfo::appNameLength`

8.38.2.2 `uint8_t loc_LocApplicationInfo::appProviderLength`

8.38.2.3 `uint8_t loc_LocApplicationInfo::appVersionLength`

8.38.2.4 `uint8_t loc_LocApplicationInfo::appVersionValid`

8.38.2.5 `uint8_t* loc_LocApplicationInfo::pAppName`

8.38.2.6 `uint8_t* loc_LocApplicationInfo::pAppProvider`

8.38.2.7 `uint8_t* loc_LocApplicationInfo::pAppVersion`

8.39 loc_precisionDilution Struct Reference

Data Fields

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

8.39.1 Detailed Description

This structure contains Dilution of precision associated with this position.

Parameters

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

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

8.39.2 Field Documentation

8.39.2.1 uint32_t loc_precisionDilution::HDOP

8.39.2.2 uint32_t loc_precisionDilution::PDOP

8.39.2.3 uint32_t loc_precisionDilution::VDOP

8.40 loc_sensorDataUsage Struct Reference

Data Fields

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

8.40.1 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

8.40.2 Field Documentation

8.40.2.1 `uint32_t loc_sensorDataUsage::aidingIndicatorMask`

8.40.2.2 `uint32_t loc_sensorDataUsage::usageMask`

8.41 `loc_SV` Struct Reference

Data Fields

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

8.41.1 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>

8.41.2 Field Documentation

8.41.2.1 `uint16_t loc_SV::id`

8.41.2.2 `uint8_t loc_SV::mask`

8.41.2.3 `uint32_t loc_SV::system`

8.42 `loc_SVInfo` Struct Reference

Data Fields

- `uint8_t len`
- `loc_SV * pSV`

8.42.1 Detailed Description

This structure contains the elements of Delete LOC SV Info

Parameters

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

8.42.2 Field Documentation

8.42.2.1 `uint8_t loc_SVInfo::len`

8.42.2.2 `loc_SV* loc_SVInfo::pSV`

8.43 loc_svUsedforFix Struct Reference

Data Fields

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

8.43.1 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

8.43.2 Field Documentation

8.43.2.1 `uint16_t loc_svUsedforFix::gnssSvUsedList[255]`

8.43.2.2 `uint8_t loc_svUsedforFix::gnssSvUsedList_len`

8.44 lteSSInfo Struct Reference

Data Fields

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

8.44.1 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.

8.44.2 Field Documentation

8.44.2.1 `int16_t lteSSInfo::rsrp`

8.44.2.2 `int8_t lteSSInfo::rsrq`

8.44.2.3 `int8_t lteSSInfo::rssi`

8.44.2.4 `int16_t lteSSInfo::snr`

8.45 messageModeTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSMessageModelInfo MessageModelInfo`

8.45.1 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

8.45.2 Field Documentation

8.45.2.1 `sMSMessageModelInfo` `messageModeTlv::MessageModelInfo`

8.45.2.2 `uint8_t` `messageModeTlv::TlvPresent`

8.46 nas_acqOrderPref Struct Reference

Data Fields

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

8.46.1 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

8.46.2 Field Documentation

8.46.2.1 `uint8_t` `nas_acqOrderPref::acqOrdeLen`

8.46.2.2 `uint8_t*` `nas_acqOrderPref::pAcqOrder`

8.47 nas_AddCDMASysInfo Struct Reference

Data Fields

- `uint16_t` `geoSysIdx`
- `uint16_t` `regPrd`

8.47.1 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.

8.47.2 Field Documentation

8.47.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

8.47.2.2 uint16_t nas_AddCDMASysInfo::regPrd

8.48 nas_AddSysInfo Struct Reference

Data Fields

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

8.48.1 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

8.48.2 Field Documentation

8.48.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

8.48.2.2 uint16_t nas_AddSysInfo::geoSysIdx

8.49 nas_CallBarringSysInfo Struct Reference

Data Fields

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

8.49.1 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

8.49.2 Field Documentation

8.49.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

8.49.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

8.50 nas_callBarStatus Struct Reference

Data Fields

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

8.50.1 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

8.50.2 Field Documentation

8.50.2.1 `uint32_t nas_callBarStatus::csBarStatus`8.50.2.2 `uint32_t nas_callBarStatus::psBarStatus`8.51 `nas_CDMAECIOThresh` Struct Reference

Data Fields

- `uint8_t CDMAECIOThreshListLen`
- `int16_t * pCDMAECIOThreshList`

8.51.1 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)

8.51.2 Field Documentation

8.51.2.1 `uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen`

8.51.2.2 int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList

8.52 nas_CDMAInfo Struct Reference

Data Fields

- uint16_t [sid](#)
- uint16_t [nid](#)
- uint16_t [baseId](#)
- uint16_t [refpn](#)
- uint32_t [baseLat](#)
- uint32_t [baseLong](#)

8.52.1 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

8.52.2 Field Documentation

8.52.2.1 uint16_t nas_CDMAInfo::baseId

8.52.2.2 uint32_t nas_CDMAInfo::baseLat

8.52.2.3 uint32_t nas_CDMAInfo::baseLong

8.52.2.4 `uint16_t nas_CDMAInfo::nid`

8.52.2.5 `uint16_t nas_CDMAInfo::refpn`

8.52.2.6 `uint16_t nas_CDMAInfo::sid`

8.53 `nas_CDMARSSIThresh` Struct Reference

Data Fields

- `uint8_t CDMARSSIThreshListLen`
- `int16_t * pCDMARSSIThreshList`

8.53.1 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).

8.53.2 Field Documentation

8.53.2.1 `uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen`

8.53.2.2 `int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList`

8.54 `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]

8.54.1 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>isSysPrIMatchValid</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}.

8.54.2 Field Documentation

8.54.2.1 `uint16_t nas_CDMA SysInfo::baseId`

8.54.2.2 `uint32_t nas_CDMA SysInfo::baseLat`

8.54.2.3 `uint32_t nas_CDMA SysInfo::baseLong`

8.54.2.4 `uint8_t nas_CDMA SysInfo::bsInfoValid`

8.54.2.5 `uint8_t nas_CDMA SysInfo::bsPRev`

8.54.2.6 `uint8_t nas_CDMA SysInfo::bsPRevValid`

8.54.2.7 `uint8_t nas_CDMA SysInfo::ccsSupported`

8.54.2.8 `uint8_t nas_CDMA SysInfo::ccsSupportedValid`

8.54.2.9 `uint8_t nas_CDMA SysInfo::cdmaSysIdValid`

8.54.2.10 `uint8_t nas_CDMA SysInfo::isSysPrIMatch`

8.54.2.11 `uint8_t nas_CDMA SysInfo::isSysPrIMatchValid`

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

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

8.54.2.14 `uint16_t nas_CDMA SysInfo::networkId`

- 8.54.2.15 `uint8_t nas_CDMA SysInfo::networkIdValid`
- 8.54.2.16 `uint16_t nas_CDMA SysInfo::packetZone`
- 8.54.2.17 `uint8_t nas_CDMA SysInfo::packetZoneValid`
- 8.54.2.18 `uint8_t nas_CDMA SysInfo::pRevInUse`
- 8.54.2.19 `uint8_t nas_CDMA SysInfo::pRevInUseValid`
- 8.54.2.20 `nas_sysInfoCommon nas_CDMA SysInfo::sysInfoCDMA`
- 8.54.2.21 `uint16_t nas_CDMA SysInfo::systemID`

8.55 nas_CDMA SysInfoExt Struct Reference

Data Fields

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

8.55.1 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

8.55.2 Field Documentation

- 8.55.2.1 `uint8_t nas_CDMA SysInfoExt::imsi_11_12`
- 8.55.2.2 `uint16_t nas_CDMA SysInfoExt::MCC`

8.56 nas_cellParams Struct Reference

Data Fields

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

8.56.1 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>rsqi</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.

8.56.2 Field Documentation

8.56.2.1 `uint16_t nas_cellParams::pci`

8.56.2.2 `int16_t nas_cellParams::rsrp`

8.56.2.3 `int16_t nas_cellParams::rsrq`

8.56.2.4 `int16_t nas_cellParams::rsqi`

8.56.2.5 `int16_t nas_cellParams::srxlev`

8.57 nas_CommInfo Struct Reference

Data Fields

- `int8_t temperature`
- `uint8_t modemMode`
- `uint8_t systemMode`
- `uint8_t imsRegState`
- `uint8_t psState`

8.57.1 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

8.57.2 Field Documentation

8.57.2.1 `uint8_t nas_CommInfo::imsRegState`

8.57.2.2 `uint8_t nas_CommInfo::modemMode`

8.57.2.3 `uint8_t nas_CommInfo::psState`

8.57.2.4 `uint8_t nas_CommInfo::systemMode`

8.57.2.5 `int8_t nas_CommInfo::temperature`

8.58 nas_CSGID Struct Reference

Data Fields

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

8.58.1 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

8.58.2 Field Documentation

8.58.2.1 `uint32_t nas_CSGID::id`

8.58.2.2 `uint16_t nas_CSGID::mcc`

8.58.2.3 `uint16_t nas_CSGID::mnc`

8.58.2.4 `uint8_t nas_CSGID::mncPcsDigits`

8.58.2.5 `uint8_t nas_CSGID::rat`

8.59 nas_currentPLMN Struct Reference

Data Fields

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

8.59.1 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

8.59.2 Field Documentation

8.59.2.1 `uint16_t nas_currentPLMN::MCC`

8.59.2.2 `uint16_t nas_currentPLMN::MNC`

8.59.2.3 `uint8_t nas_currentPLMN::netDescr[255]`

8.59.2.4 `uint8_t nas_currentPLMN::netDescrLength`

8.60 nas_dataSrvCapabilities Struct Reference

Data Fields

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

8.60.1 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+

8.60.2 Field Documentation

8.60.2.1 `uint8_t nas_dataSrvCapabilities::dataCapabilities[32]`

8.60.2.2 `uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen`

8.61 nas_detailSvcInfo Struct Reference

Data Fields

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

8.61.1 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
-----------------------	---

8.61.2 Field Documentation

8.61.2.1 uint8_t nas_detailSvcInfo::hdrHybrid

8.61.2.2 uint8_t nas_detailSvcInfo::hdrSrvStatus

8.61.2.3 uint8_t nas_detailSvcInfo::isSysForbidden

8.61.2.4 uint8_t nas_detailSvcInfo::srvCapability

8.61.2.5 uint8_t nas_detailSvcInfo::srvStatus

8.62 nas_ecioListElement Struct Reference

Data Fields

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

8.62.1 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

8.62.2 Field Documentation

8.62.2.1 int16_t nas_ecioListElement::ecio

8.62.2.2 uint8_t nas_ecioListElement::radiolf

8.63 nas_errorRateListElement Struct Reference

Data Fields

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

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

8.63.2 Field Documentation

8.63.2.1 uint16_t nas_errorRateListElement::errorRate

8.63.2.2 uint8_t nas_errorRateListElement::radioIrf

8.64 nas_GERANInfo Struct Reference

Data Fields

- uint32_t [cellID](#)
- uint8_t [plmn](#) [3]
- uint16_t [lac](#)
- uint16_t [arfcn](#)
- uint8_t [bsic](#)
- uint32_t [timingAdvance](#)
- uint16_t [rxLev](#)
- uint8_t [nrmInst](#)
- [nas_nmrCellInfo](#) [insNmrCellInfo](#) [255]

8.64.1 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.

8.64.2 Field Documentation

8.64.2.1 `uint16_t nas_GERANInfo::arfcn`

8.64.2.2 `uint8_t nas_GERANInfo::bsic`

8.64.2.3 `uint32_t nas_GERANInfo::cellID`

8.64.2.4 `nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]`

8.64.2.5 `uint16_t nas_GERANInfo::lac`

8.64.2.6 `uint8_t nas_GERANInfo::nmrInst`

8.64.2.7 `uint8_t nas_GERANInfo::plmn[3]`

8.64.2.8 `uint16_t nas_GERANInfo::rxLev`

8.64.2.9 `uint32_t nas_GERANInfo::timingAdvance`

8.65 nas_geranInstInfo Struct Reference

Data Fields

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

8.65.1 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.

8.65.2 Field Documentation

8.65.2.1 uint16_t nas_geranInstInfo::geranArfcn

8.65.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

8.65.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

8.65.2.4 int16_t nas_geranInstInfo::geranRssi

8.66 nas_gsmCellInfo Struct Reference

Data Fields

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

8.66.1 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.

8.66.2 Field Documentation

8.66.2.1 `uint16_t nas_gsmCellInfo::arfcn`

8.66.2.2 `uint8_t nas_gsmCellInfo::band1900`

8.66.2.3 `uint8_t nas_gsmCellInfo::bsicId`

8.66.2.4 `uint8_t nas_gsmCellInfo::cellIdValid`

8.66.2.5 `int16_t nas_gsmCellInfo::rsSI`

8.66.2.6 `int16_t nas_gsmCellInfo::srxlev`

8.67 nas_GSMRSSIThresh Struct Reference

Data Fields

- `uint8_t GSMRSSIThreshListLen`
- `int16_t * pGSMRSSIThreshList`

8.67.1 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)

8.67.2 Field Documentation

8.67.2.1 uint8_t nas_GSMRSSIthresh::GSMRSSIthreshListLen

8.67.2.2 int16_t* nas_GSMRSSIthresh::pGSMRSSIthreshList

8.68 nas_GSMsRvStatusInfo Struct Reference

Data Fields

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

8.68.1 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
-----------------------	---

8.68.2 Field Documentation

8.68.2.1 `uint8_t nas_GSMsSrvStatusInfo::isPrefDataPath`

8.68.2.2 `uint8_t nas_GSMsSrvStatusInfo::srvStatus`

8.68.2.3 `uint8_t nas_GSMsSrvStatusInfo::trueSrvStatus`

8.69 nas_GSMsSysInfo 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`

8.69.1 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

8.69.2 Field Documentation

8.69.2.1 `uint32_t nas_GSM SysInfo::cellId`

8.69.2.2 `uint8_t nas_GSM SysInfo::cellIdValid`

8.69.2.3 `uint8_t nas_GSM SysInfo::dtmSupp`

8.69.2.4 `uint8_t nas_GSM SysInfo::dtmSuppValid`

8.69.2.5 `uint8_t nas_GSM SysInfo::egprsSupp`

8.69.2.6 `uint8_t nas_GSM SysInfo::egprsSuppValid`

8.69.2.7 `uint16_t nas_GSM SysInfo::lac`

8.69.2.8 `uint8_t nas_GSM SysInfo::lacValid`

8.69.2.9 `uint8_t nas_GSM SysInfo::MCC[3]`

8.69.2.10 `uint8_t nas_GSM SysInfo::MNC[3]`

8.69.2.11 `uint8_t nas_GSM SysInfo::networkIdValid`

8.69.2.12 `uint8_t nas_GSM SysInfo::regRejectInfoValid`

8.69.2.13 `uint8_t nas_GSMsysInfo::rejCause`

8.69.2.14 `uint8_t nas_GSMsysInfo::rejectSrvDomain`

8.69.2.15 `nas_sysInfoCommon nas_GSMsysInfo::sysInfoGSM`

8.70 nas_HDRECIOTthresh Struct Reference

Data Fields

- `uint8_t HDRECIOTthreshListLen`
- `int16_t * pHDRECIOTthreshList`

8.70.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

Parameters

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

8.70.2 Field Documentation

8.70.2.1 `uint8_t nas_HDRECIOTthresh::HDRECIOTthreshListLen`

8.70.2.2 `int16_t* nas_HDRECIOTthresh::pHDRECIOTthreshList`

8.71 nas_HDRIOTthresh Struct Reference

Data Fields

- `uint8_t HDRIOTthreshListLen`
- `int16_t * pHDRIOTthreshList`

8.71.1 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).

8.71.2 Field Documentation

8.71.2.1 `uint8_t nas_HDRIOTresh::HDRIOTreshListLen`8.71.2.2 `int16_t* nas_HDRIOTresh::pHDRIOTreshList`8.72 `nas_HDRRSSIthresh` Struct Reference

Data Fields

- `uint8_t HDRRSSIthreshListLen`
- `int16_t * pHDRRSSIthreshList`

8.72.1 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).

8.72.2 Field Documentation

8.72.2.1 `uint8_t nas_HDRRSSIthresh::HDRRSSIthreshListLen`8.72.2.2 `int16_t* nas_HDRRSSIthresh::pHDRRSSIthreshList`8.73 `nas_HDRSINRThreshold` Struct Reference

Data Fields

- `uint8_t HDRSINRthreshListLen`
- `uint16_t * pHDRSINRthreshList`

8.73.1 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

8.73.2 Field Documentation

8.73.2.1 `uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen`

8.73.2.2 `uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList`

8.74 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]`

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

8.74.2 Field Documentation

8.74.2.1 `uint8_t nas_HDRSysInfo::hdrActiveProt`

8.74.2.2 `uint8_t nas_HDRSysInfo::hdrActiveProtValid`

8.74.2.3 `uint8_t nas_HDRSysInfo::hdrPersonality`

8.74.2.4 `uint8_t nas_HDRSysInfo::hdrPersonalityValid`

8.74.2.5 `uint8_t nas_HDRSysInfo::is856SysId[16]`

8.74.2.6 `uint8_t nas_HDRSysInfo::is856SysIdValid`

8.74.2.7 `uint8_t nas_HDRSysInfo::isSysPrIMatch`

8.74.2.8 `uint8_t nas_HDRSysInfo::isSysPrIMatchValid`

8.74.2.9 `nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR`

8.75 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]

8.75.1 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 <i>Srxlev</i> low threshold. Range: 0 to 31. When the serving cell does not exceed <i>thresh_serving_low</i>, 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.

8.75.2 Field Documentation

8.75.2.1 `uint8_t nas_infoInterFreq::cell_resel_priority`

8.75.2.2 `nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]`

8.75.2.3 `uint8_t nas_infoInterFreq::cells_len`

8.75.2.4 `uint16_t nas_infoInterFreq::earfcn`

8.75.2.5 `uint8_t nas_infoInterFreq::threshXHigh`

8.75.2.6 `uint8_t nas_infoInterFreq::threshXLow`

8.76 nas_lteGsmCellInfo Struct Reference

Data Fields

- `uint8_t cellReselPriority`
- `uint8_t threshGsmHigh`
- `uint8_t threshGsmLow`
- `uint8_t nccPermitted`
- `uint8_t cells_len`
- `nas_gsmCellInfo GsmCellInfo [255]`

8.76.1 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.

8.76.2 Field Documentation

8.76.2.1 `uint8_t nas_lteGsmCellInfo::cellReselPriority`8.76.2.2 `uint8_t nas_lteGsmCellInfo::cells_len`8.76.2.3 `nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]`8.76.2.4 `uint8_t nas_lteGsmCellInfo::nccPermitted`8.76.2.5 `uint8_t nas_lteGsmCellInfo::threshGsmHigh`8.76.2.6 `uint8_t nas_lteGsmCellInfo::threshGsmLow`

8.77 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](#)

8.77.1 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

8.77.2 Field Documentation

8.77.2.1 `uint8_t nas_LTEInfo::band`

8.77.2.2 `uint8_t nas_LTEInfo::bandwidth`

8.77.2.3 `uint8_t nas_LTEInfo::emmConnState`

8.77.2.4 `uint8_t nas_LTEInfo::emmState`

8.77.2.5 `uint8_t nas_LTEInfo::emmSubState`

8.77.2.6 `uint16_t nas_LTEInfo::RXChan`

8.77.2.7 `uint16_t nas_LTEInfo::TXChan`

8.78 nas_LTEInfoInterfreq Struct Reference

Data Fields

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

8.78.1 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.

8.78.2 Field Documentation

8.78.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

8.78.2.2 nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]

8.78.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

8.79 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]

8.79.1 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.

8.79.2 Field Documentation

8.79.2.1 `nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]`

8.79.2.2 `uint8_t nas_LTEInfoIntrafreq::cellReselPriority`

8.79.2.3 `uint8_t nas_LTEInfoIntrafreq::cellsLen`

8.79.2.4 `uint16_t nas_LTEInfoIntrafreq::earfcn`

8.79.2.5 `uint32_t nas_LTEInfoIntrafreq::globalCellId`

8.79.2.6 `uint8_t nas_LTEInfoIntrafreq::plmn[3]`

8.79.2.7 `uint16_t nas_LTEInfoIntrafreq::servingCellId`

8.79.2.8 `uint8_t nas_LTEInfoIntrafreq::sIntraSearch`

8.79.2.9 `uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch`

8.79.2.10 `uint16_t nas_LTEInfoIntrafreq::tac`

8.79.2.11 `uint8_t nas_LTEInfoIntrafreq::threshServingLow`

8.79.2.12 `uint8_t nas_LTEInfoIntrafreq::ueInIdle`

8.80 nas_LTEInfoNeighboringGSM Struct Reference

Data Fields

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

8.80.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

Parameters

<i>ueIdle</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.

8.80.2 Field Documentation

8.80.2.1 `uint8_t nas_LTEInfoNeighboringGSM::freqsLen`

8.80.2.2 `nas_LteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]`

8.80.2.3 `uint8_t nas_LTEInfoNeighboringGSM::ueIdle`

8.81 nas_LTEInfoNeighboringWCDMA Struct Reference

Data Fields

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

8.81.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

Parameters

<i>ueIdle</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.

8.81.2 Field Documentation

8.81.2.1 `uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen`

8.81.2.2 `nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]`

8.81.2.3 `uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle`

8.82 nas_lteRsrpinformation Struct Reference

Data Fields

- `int16_t rsrplevel`

8.82.1 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).
------------------	--

8.82.2 Field Documentation

8.82.2.1 `int16_t nas_lteRsrpinformation::rsrplevel`

8.83 nas_LTEsrpThresh Struct Reference

Data Fields

- `uint8_t LTERSRPThreshListLen`
- `int16_t * pLTERSRPThreshList`

8.83.1 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).

8.83.2 Field Documentation

8.83.2.1 `uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen`

8.83.2.2 `int16_t* nas_LTERSRPThresh::pLTERSRPThreshList`

8.84 nas_LTERSRQThresh Struct Reference

Data Fields

- `uint8_t LTERSRQThreshListLen`
- `int16_t* pLTERSRQThreshList`

8.84.1 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)

8.84.2 Field Documentation

8.84.2.1 `uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen`

8.84.2.2 `int16_t* nas_LTERSRQThresh::pLTERSRQThreshList`

8.85 nas_LTERSSIThresh Struct Reference

Data Fields

- `uint8_t LTERSSIThreshListLen`
- `int16_t* pLTERSSIThreshList`

8.85.1 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)

8.85.2 Field Documentation

8.85.2.1 `uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen`

8.85.2.2 `int16_t* nas_LTERSSIThresh::pLTERSSIThreshList`

8.86 nas_LTESigRptConfig Struct Reference

Data Fields

- `uint8_t rptRate`
- `uint8_t avgPeriod`

8.86.1 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

8.86.2 Field Documentation

8.86.2.1 `uint8_t nas_LTESigRptConfig::avgPeriod`

8.86.2.2 `uint8_t nas_LTESigRptConfig::rptRate`

8.87 nas_IteSnrinformation Struct Reference

Data Fields

- [int16_t snrlevel](#)

8.87.1 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.
-----------------	---

8.87.2 Field Documentation

8.87.2.1 [int16_t nas_lteSnrinformation::snrlevel](#)

8.88 nas_LTESNRThreshold Struct Reference

Data Fields

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

8.88.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

Parameters

<i>LTESNRThreshListLen</i>	<ul style="list-style-type: none"> • Length of the LTE SNR threshold list parameter to follow
<i>pLTESNRThreshList</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).

8.88.2 Field Documentation

8.88.2.1 [uint8_t nas_LTESNRThreshold::LTESNRThreshListLen](#)

8.88.2.2 [int16_t* nas_LTESNRThreshold::pLTESNRThreshList](#)

8.89 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](#)

8.89.1 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

8.89.2 Field Documentation

8.89.2.1 `uint32_t nas_LTESysInfo::cellId`

8.89.2.2 `uint8_t nas_LTESysInfo::cellIdValid`

8.89.2.3 `uint16_t nas_LTESysInfo::lac`

8.89.2.4 `uint8_t nas_LTESysInfo::lacValid`

8.89.2.5 `uint8_t nas_LTESysInfo::MCC[3]`

8.89.2.6 `uint8_t nas_LTESysInfo::MNC[3]`

8.89.2.7 `uint8_t nas_LTESysInfo::networkIdValid`

8.89.2.8 `uint8_t nas_LTESysInfo::regRejectInfoValid`

8.89.2.9 `uint8_t nas_LTESysInfo::rejCause`

8.89.2.10 `uint8_t nas_LTESysInfo::rejectSrvDomain`

8.89.2.11 `nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE`

8.89.2.12 `uint16_t nas_LTESysInfo::tac`

8.89.2.13 `uint8_t nas_LTESysInfo::tacValid`

8.90 `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]

8.90.1 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.

8.90.2 Field Documentation

8.90.2.1 `uint8_t nas_lteWcdmaCellInfo::cellReselPriority`

8.90.2.2 `uint8_t nas_lteWcdmaCellInfo::cellsLen`

8.90.2.3 `uint16_t nas_lteWcdmaCellInfo::threshXhigh`

8.90.2.4 `uint16_t nas_lteWcdmaCellInfo::threshXlow`

8.90.2.5 `uint16_t nas_lteWcdmaCellInfo::uarfcn`

8.90.2.6 `nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]`

8.91 nas_MNRInfo Struct Reference

Data Fields

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

8.91.1 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

8.91.2 Field Documentation

8.91.2.1 uint16_t nas_MNRInfo::mcc

8.91.2.2 uint16_t nas_MNRInfo::mnc

8.91.2.3 uint32_t nas_MNRInfo::rat

8.92 nas_netSelectionPref Struct Reference

Data Fields

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

8.92.1 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

8.92.2 Field Documentation

8.92.2.1 uint16_t nas_netSelectionPref::mcc

8.92.2.2 uint16_t nas_netSelectionPref::mnc

8.92.2.3 uint8_t nas_netSelectionPref::netReg

8.93 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](#)

8.93.1 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

8.93.2 Field Documentation

8.93.2.1 `uint16_t nas_nmrCellInfo::nmrArfcn`

8.93.2.2 `uint8_t nas_nmrCellInfo::nmrBsic`

8.93.2.3 `uint32_t nas_nmrCellInfo::nmrCellID`

8.93.2.4 `uint16_t nas_nmrCellInfo::nmrLac`

8.93.2.5 `uint8_t nas_nmrCellInfo::nmrPlmn[3]`

8.93.2.6 `uint16_t nas_nmrCellInfo::nmrRxLev`

8.94 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](#)

8.94.1 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.

8.94.2 Field Documentation

8.94.2.1 [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) [nas_PhyCaAggPcellInfo::dl_bw_value](#)

8.94.2.2 [uint16_t](#) [nas_PhyCaAggPcellInfo::freq](#)

8.94.2.3 [uint16_t](#) [nas_PhyCaAggPcellInfo::iLTEbandValue](#)

8.94.2.4 [uint16_t](#) [nas_PhyCaAggPcellInfo::pci](#)

8.94.2.5 [uint8_t](#) [nas_PhyCaAggPcellInfo::TlvPresent](#)

8.95 nas_PhyCaAggScellIDBw Struct Reference

Data Fields

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

- uint8_t [TlvPresent](#)

8.95.1 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.
--------------------	--

8.95.2 Field Documentation

8.95.2.1 [NAS_LTE_CPHY_CA_BW_NRB_LITE](#) nas_PhyCaAggScellIDBw::dl_bw_value

8.95.2.2 uint8_t nas_PhyCaAggScellIDBw::TlvPresent

8.96 nas_PhyCaAggScellIndex Struct Reference

Data Fields

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

8.96.1 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.

8.96.2 Field Documentation

8.96.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

8.96.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

8.97 nas_PhyCaAggScellIndType Struct Reference

Data Fields

- uint16_t [pci](#)
- uint16_t [freq](#)

- [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 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.

8.97.2 Field Documentation

8.97.2.1 `uint16_t` *nas_PhyCaAggScellIndType::freq*

8.97.2.2 `uint16_t` *nas_PhyCaAggScellIndType::pci*

8.97.2.3 [NAS_LTE_CPHY_SCELL_STATE_LITE](#) *nas_PhyCaAggScellIndType::scell_state*

8.97.2.4 `uint8_t` *nas_PhyCaAggScellIndType::TlvPresent*

8.98 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*

8.98.1 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.
-------------------	--

8.98.2 Field Documentation

8.98.2.1 **NAS_LTE_CPHY_CA_BW_NRB_LITE** nas_PhyCaAggScellInfo::dl_bw_value

8.98.2.2 uint16_t nas_PhyCaAggScellInfo::freq

8.98.2.3 uint16_t nas_PhyCaAggScellInfo::iLTEbandValue

8.98.2.4 uint16_t nas_PhyCaAggScellInfo::pci

8.98.2.5 **NAS_LTE_CPHY_SCELL_STATE_LITE** nas_PhyCaAggScellInfo::scell_state

8.98.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

8.99 nas_qaQmi3Gpp2TimeZone Struct Reference

Data Fields

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

8.99.1 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)

8.99.2 Field Documentation

8.99.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

8.99.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

8.99.2.3 `uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset`

8.100 `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]

8.100.1 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

8.100.2 Field Documentation

8.100.2.1 `char nas_QmiNas3GppNetworkInfo::Description[255]`

8.100.2.2 `uint32_t nas_QmiNas3GppNetworkInfo::Forbidden`

8.100.2.3 `uint32_t nas_QmiNas3GppNetworkInfo::InUse`

8.100.2.4 `uint16_t nas_QmiNas3GppNetworkInfo::MCC`

8.100.2.5 `uint16_t nas_QmiNas3GppNetworkInfo::MNC`

8.100.2.6 `uint32_t nas_QmiNas3GppNetworkInfo::Preferred`

8.100.2.7 `uint32_t nas_QmiNas3GppNetworkInfo::Roaming`

8.101 `nas_QmiNas3GppNetworkRAT` Struct Reference

Data Fields

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

8.101.1 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

8.101.2 Field Documentation

8.101.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

8.101.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

8.101.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

8.102 nas_QmisNasPcsDigit Struct Reference

Data Fields

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

8.102.1 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

8.102.2 Field Documentation

8.102.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

8.102.2.2 uint16_t nas_QmisNasPcsDigit::MCC

8.102.2.3 uint16_t nas_QmisNasPcsDigit::MNC

8.103 nas_RejectReasonTlv Struct Reference

Data Fields

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

8.103.1 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

8.103.2 Field Documentation

8.103.2.1 uint32_t nas_RejectReasonTlv::rejectCause

8.103.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

8.103.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

8.104 nas_RFInfoTlv Struct Reference

Data Fields

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

8.104.1 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

8.104.2 Field Documentation

8.104.2.1 uint32_t nas_RFInfoTlv::activeBandClass[255]

8.104.2.2 uint32_t nas_RFInfoTlv::activeChannel[255]

8.104.2.3 uint32_t nas_RFInfoTlv::radiolInterface[255]

8.104.2.4 uint8_t nas_RFInfoTlv::radiolInterfaceSize

8.104.2.5 uint8_t nas_RFInfoTlv::TlvPresent

8.105 nas_roamIndList Struct Reference

Data Fields

- uint8_t [numInstances](#)
- uint8_t [radiolInterface](#) [32]
- uint8_t [roamIndicator](#) [32]

8.105.1 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_1XEVD0 - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE
<i>roamIndicator</i>	<ul style="list-style-type: none"> • Roaming Indicator • Values: <ul style="list-style-type: none"> – 0x00 - Roaming – 0x01 - Home

8.105.2 Field Documentation

8.105.2.1 uint8_t nas_roamIndList::numInstances

8.105.2.2 uint8_t nas_roamIndList::radiolInterface[32]

8.105.2.3 `uint8_t nas_roamIndList::roamIndicator[32]`

8.106 `nas_rsrqInformation` Struct Reference

Data Fields

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

8.106.1 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

8.106.2 Field Documentation

8.106.2.1 `uint8_t nas_rsrqInformation::radiolf`

8.106.2.2 `int8_t nas_rsrqInformation::rsrq`

8.107 `nas_RxSigInfo` Struct Reference

Data Fields

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

8.107.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

Parameters

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

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

Note

If the radio is tuned, the instantaneous values are set for the fields below. If the radio is not tuned, the values set below may be invalid.

Parameters

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

8.107.2 Field Documentation

8.107.2.1 `uint8_t nas_RxSigInfo::isRadioTuned`

8.107.2.2 `int32_t nas_RxSigInfo::rsrp`

8.107.2.3 `uint8_t nas_RxSigInfo::rxChainIndex`

8.107.2.4 `int32_t nas_RxSigInfo::rxPower`

8.108 nas_rxSignalStrengthListElement Struct Reference**Data Fields**

- `int16_t rxSignalStrength`
- `uint8_t radiolfl`

8.108.1 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.

8.108.2 Field Documentation

8.108.2.1 `uint8_t nas_rxSignalStrengthListElement::radioIf`

8.108.2.2 `int16_t nas_rxSignalStrengthListElement::rxSignalStrength`

8.109 nas_SccRxInfo Struct Reference**Data Fields**

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

8.109.1 Detailed Description

This structure contains information about the SccRxInfo parameters.

Parameters

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

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

8.109.2 Field Documentation

8.109.2.1 `uint8_t nas_SccRxInfo::numInstances`

8.109.2.2 `int32_t nas_SccRxInfo::rsrq`

8.109.2.3 `nas_RxSigInfo nas_SccRxInfo::sigInfo[255]`

8.109.2.4 `int16_t nas_SccRxInfo::snr`

8.109.2.5 `uint8_t nas_SccRxInfo::TlvPresent`

8.110 nas_servSystem Struct Reference

Data Fields

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

8.110.1 Detailed Description

This structure contains the Serving System parameters

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

Parameters

<i>regState</i>	<ul style="list-style-type: none"> • Registration state - Registration state of the mobile • Values: <ul style="list-style-type: none"> – 0 - Not Registered; mobile is not currently searching for a new network to provide service – 1 - Registered with a network – 2 - Not registered, but mobile is currently searching for a new network to provide service – 3 - Registration denied by visible network – 4 - Registration state is unknown
-----------------	--

<i>csAttachState</i>	<ul style="list-style-type: none"> • CS Attach State - Circuit-switched domain attach state of the mobile • Values: <ul style="list-style-type: none"> – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
<i>psAttachState</i>	<ul style="list-style-type: none"> • PS Attach State - Packet-switched domain attach state of the mobile • Values: <ul style="list-style-type: none"> – 0 - Unknown or not applicable – 1 - Attached – 2 - Detached
<i>selNetwork</i>	<ul style="list-style-type: none"> • Selected Network - Type of selected radio access network • Values: <ul style="list-style-type: none"> – 0 - Unknown – 1 - 3GPP2 network – 2 - 3GPP network
<i>numRadio-Interfaces</i>	<ul style="list-style-type: none"> • In Use Radio Interfaces Number <ul style="list-style-type: none"> – Number of radio interfaces currently in use – defaults to zero
<i>radioInterface</i>	<ul style="list-style-type: none"> • Radio Interface currently in use • Values: <ul style="list-style-type: none"> – 0x00 - RADIO_IF_NO_SVC - None(no service) – 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X – 0x02 - RADIO_IF_CDMA_1xEVDO - cdma2000 HRPD (1xEV-DO) – 0x03 - RADIO_IF_AMPS - AMPS – 0x04 - RADIO_IF_GSM - GSM – 0x05 - RADIO_IF_UMTS - UMTS – 0x08 - RADIO_IF_LTE - LTE

8.110.2 Field Documentation

8.110.2.1 `uint8_t nas_servSystem::csAttachState`

8.110.2.2 `uint8_t nas_servSystem::numRadioInterfaces`

8.110.2.3 `uint8_t nas_servSystem::psAttachState`

8.110.2.4 `uint8_t nas_servSystem::radioInterface[32]`

8.110.2.5 `uint8_t nas_servSystem::regState`

8.110.2.6 `uint8_t nas_servSystem::selNetwork`

8.111 nas_SignalStrengthTlv Struct Reference

Data Fields

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

8.111.1 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

8.111.2 Field Documentation

8.111.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

8.111.2.2 int8_t nas_SignalStrengthTlv::signalStrength

8.111.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

8.112 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](#)

8.112.1 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

8.112.2 Field Documentation

- 8.112.2.1 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta`
- 8.112.2.2 `int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]`
- 8.112.2.3 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen`
- 8.112.2.4 `uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta`
- 8.112.2.5 `uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta`
- 8.112.2.6 `uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta`
- 8.112.2.7 `uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta`
- 8.112.2.8 `uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta`
- 8.112.2.9 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta`
- 8.112.2.10 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]`
- 8.112.2.11 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen`

8.113 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](#)

8.113.1 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

8.113.2 Field Documentation

8.113.2.1 `nas_ecioListElement` `nas_SLQSSignalStrengthsInformation::eciInfo`

8.113.2.2 `nas_errorRateListElement` `nas_SLQSSignalStrengthsInformation::errorRateInfo`

8.113.2.3 `uint32_t` `nas_SLQSSignalStrengthsInformation::io`

8.113.2.4 `nas_lteRsrpinformation` `nas_SLQSSignalStrengthsInformation::lteRsrpinfo`

8.113.2.5 `nas_lteSnrinformation` `nas_SLQSSignalStrengthsInformation::lteSnrinfo`

8.113.2.6 `nas_rsrqInformation` `nas_SLQSSignalStrengthsInformation::rsrqInfo`

8.113.2.7 `nas_rxSignalStrengthListElement` `nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo`

8.113.2.8 `uint8_t` `nas_SLQSSignalStrengthsInformation::sinr`

8.114 nas_SLQSSignalStrengthsTlv Struct Reference

Data Fields

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

8.114.1 Detailed Description

Parameters

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

8.114.2 Field Documentation

8.114.2.1 `nas_SLQSSignalStrengthsInformation` `nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo`

8.114.2.2 `uint8_t` `nas_SLQSSignalStrengthsTlv::TlvPresent`

8.115 nas_SrvStatusInfo Struct Reference

Data Fields

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

8.115.1 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

8.115.2 Field Documentation

8.115.2.1 `uint8_t nas_SrvStatusInfo::isPrefDataPath`

8.115.2.2 `uint8_t nas_SrvStatusInfo::srvStatus`

8.116 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`

8.116.1 Detailed Description

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

Parameters

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

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

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

8.116.2 Field Documentation

8.116.2.1 `uint8_t nas_sysInfoCommon::isSysForbidden`

8.116.2.2 `uint8_t nas_sysInfoCommon::isSysForbiddenValid`

8.116.2.3 `uint8_t nas_sysInfoCommon::roamStatus`

8.116.2.4 `uint8_t nas_sysInfoCommon::roamStatusValid`

8.116.2.5 `uint8_t nas_sysInfoCommon::srvCapability`

8.116.2.6 `uint8_t nas_sysInfoCommon::srvCapabilityValid`

8.116.2.7 `uint8_t nas_sysInfoCommon::srvDomain`

8.116.2.8 `uint8_t nas_sysInfoCommon::srvDomainValid`

8.117 nas_TDSCDMAECIOThresh Struct Reference

Data Fields

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

8.117.1 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.

8.117.2 Field Documentation

8.117.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.117.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

8.118 nas_TDSCDMARSCPThresh Struct Reference

Data Fields

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

8.118.1 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).

8.118.2 Field Documentation

8.118.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.118.2.2 `uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen`

8.119 nas_TDSCDMARSSIThresh Struct Reference

Data Fields

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

8.119.1 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.

8.119.2 Field Documentation

8.119.2.1 `float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList`

8.119.2.2 `uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen`

8.120 nas_TDSCDMASINRThresh Struct Reference

Data Fields

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

8.120.1 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

8.120.2 Field Documentation

8.120.2.1 float* nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.120.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

8.121 nas_timelInfo Struct Reference

Data Fields

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

8.121.1 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.

8.121.2 Field Documentation

8.121.2.1 uint8_t nas_timeInfo::day

8.121.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

8.121.2.3 uint8_t nas_timeInfo::dayOfWeek

8.121.2.4 uint8_t nas_timeInfo::hour

8.121.2.5 uint8_t nas_timeInfo::minute

8.121.2.6 uint8_t nas_timeInfo::month

8.121.2.7 uint8_t nas_timeInfo::radioInterface

8.121.2.8 uint8_t nas_timeInfo::second

8.121.2.9 int8_t nas_timeInfo::timeZone

8.121.2.10 uint8_t nas_timeInfo::TlvPresent

8.121.2.11 uint16_t nas_timeInfo::year

8.122 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]

8.122.1 Detailed Description

This structure contains information about the UMTS Network.

Parameters

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

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

8.122.2 Field Documentation

- 8.122.2.1 `uint16_t nas_UMTSInfo::cellID`
- 8.122.2.2 `int16_t nas_UMTSInfo::ecio`
- 8.122.2.3 `uint8_t nas_UMTSInfo::geranInst`
- 8.122.2.4 `nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]`
- 8.122.2.5 `uint16_t nas_UMTSInfo::lac`
- 8.122.2.6 `uint8_t nas_UMTSInfo::plmn[3]`
- 8.122.2.7 `uint16_t nas_UMTSInfo::psc`
- 8.122.2.8 `int16_t nas_UMTSInfo::rscp`
- 8.122.2.9 `uint16_t nas_UMTSInfo::uarfcn`
- 8.122.2.10 `uint8_t nas_UMTSInfo::umtsInst`
- 8.122.2.11 `nas_UMTSInstInfo nas_UMTSInfo::UMTSInstInfo[255]`

8.123 nas_UMTSInstInfo Struct Reference

Data Fields

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

8.123.1 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).

8.123.2 Field Documentation

8.123.2.1 int16_t nas_UMTSinstInfo::umtsEcio

8.123.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

8.123.2.3 int16_t nas_UMTSinstInfo::umtsRscp

8.123.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

8.124 nas_umtsLTENbrCell Struct Reference

Data Fields

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

8.124.1 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.

8.124.2 Field Documentation

8.124.2.1 uint8_t nas_umtsLTENbrCell::cellsTDD

8.124.2.2 uint16_t nas_umtsLTENbrCell::earfcn

8.124.2.3 uint16_t nas_umtsLTENbrCell::pci

8.124.2.4 uint32_t nas_umtsLTENbrCell::rsrp

8.124.2.5 uint32_t nas_umtsLTENbrCell::rsrq

8.124.2.6 int16_t nas_umtsLTENbrCell::srxlev

8.125 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](#)

8.125.1 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.

8.125.2 Field Documentation

8.125.2.1 uint8_t nas_UniversalTime::day

8.125.2.2 uint8_t nas_UniversalTime::dayOfWeek

8.125.2.3 uint8_t nas_UniversalTime::hour

8.125.2.4 uint8_t nas_UniversalTime::minute

8.125.2.5 uint8_t nas_UniversalTime::month

8.125.2.6 uint8_t nas_UniversalTime::second

8.125.2.7 uint16_t nas_UniversalTime::year

8.126 nas_wcdmaCellInfo Struct Reference

Data Fields

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

8.126.1 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.

8.126.2 Field Documentation

8.126.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

8.126.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

8.126.2.3 uint16_t nas_wcdmaCellInfo::psc

8.126.2.4 int16_t nas_wcdmaCellInfo::srxlev

8.127 nas_WCDMAECIOThresh Struct Reference

Data Fields

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

8.127.1 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)

8.127.2 Field Documentation

8.127.2.1 int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList

8.127.2.2 `uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen`

8.128 nas_WCDMAInfoLTENeighborCell Struct Reference

Data Fields

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

8.128.1 Detailed Description

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

Parameters

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

8.128.2 Field Documentation

8.128.2.1 `nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell`[255]

8.128.2.2 `uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen`

8.128.2.3 `uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCTest`

8.129 nas_WCDMARSSIThresh Struct Reference

Data Fields

- `uint8_t WCDMARSSIThreshListLen`
- `int16_t * pWCDMARSSIThreshList`

8.129.1 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)

8.129.2 Field Documentation

8.129.2.1 `int16_t` `nas_WCDMARSSIThresh::pWCDMARSSIThreshList`

8.129.2.2 `uint8_t` `nas_WCDMARSSIThresh::WCDMARSSIThreshListLen`

8.130 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`

8.130.1 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
------------	--

8.130.2 Field Documentation

- 8.130.2.1 `uint32_t nas_WCDMA SysInfo::cellId`
- 8.130.2.2 `uint8_t nas_WCDMA SysInfo::cellIdValid`
- 8.130.2.3 `uint8_t nas_WCDMA SysInfo::hsCallStatus`
- 8.130.2.4 `uint8_t nas_WCDMA SysInfo::hsCallStatusValid`
- 8.130.2.5 `uint8_t nas_WCDMA SysInfo::hsInd`
- 8.130.2.6 `uint8_t nas_WCDMA SysInfo::hsIndValid`
- 8.130.2.7 `uint16_t nas_WCDMA SysInfo::lac`
- 8.130.2.8 `uint8_t nas_WCDMA SysInfo::lacValid`
- 8.130.2.9 `uint8_t nas_WCDMA SysInfo::MCC[3]`
- 8.130.2.10 `uint8_t nas_WCDMA SysInfo::MNC[3]`
- 8.130.2.11 `uint8_t nas_WCDMA SysInfo::networkIdValid`
- 8.130.2.12 `uint16_t nas_WCDMA SysInfo::psc`
- 8.130.2.13 `uint8_t nas_WCDMA SysInfo::pscValid`
- 8.130.2.14 `uint8_t nas_WCDMA SysInfo::regRejectInfoValid`
- 8.130.2.15 `uint8_t nas_WCDMA SysInfo::rejCause`
- 8.130.2.16 `uint8_t nas_WCDMA SysInfo::rejectSrvDomain`
- 8.130.2.17 `nas_sysInfoCommon nas_WCDMA SysInfo::sysInfoWCDMA`

8.131 NASBandPreferenceTlv Struct Reference

Data Fields

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

8.131.1 Field Documentation

- 8.131.1.1 `uint64_t NASBandPreferenceTlv::band_pref`

8.131.1.2 `uint8_t NASBandPreferenceTlv::TlvPresent`

8.132 NASEmergencyModeTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint8_t EmerMode`

8.132.1 Field Documentation

8.132.1.1 `uint8_t NASEmergencyModeTlv::EmerMode`

8.132.1.2 `uint8_t NASEmergencyModeTlv::TlvPresent`

8.133 NasGetLTECphyCalInfo Struct Reference

Data Fields

- `NASPhyCaAggScellIndType PhyCaAggScellIndType`
- `NASPhyCaAggScellDIBw PhyCaAggScellDIBw`
- `NASPhyCaAggScellInfo PhyCaAggScellInfo`
- `NASPhyCaAggPcellInfo PhyCaAggPcellInfo`
- `NASPhyCaAggScellIndex PhyCaAggScellIndex`

8.133.1 Field Documentation

8.133.1.1 `NASPhyCaAggPcellInfo NasGetLTECphyCalInfo::PhyCaAggPcellInfo`

8.133.1.2 `NASPhyCaAggScellDIBw NasGetLTECphyCalInfo::PhyCaAggScellDIBw`

8.133.1.3 `NASPhyCaAggScellIndex NasGetLTECphyCalInfo::PhyCaAggScellIndex`

8.133.1.4 `NASPhyCaAggScellIndType NasGetLTECphyCalInfo::PhyCaAggScellIndType`

8.133.1.5 `NASPhyCaAggScellInfo NasGetLTECphyCalInfo::PhyCaAggScellInfo`

8.134 NASGWAcqOrderPrefTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `uint32_t GWAcqOrderPref`

8.134.1 Field Documentation

8.134.1.1 `uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref`

8.134.1.2 `uint8_t NASGWAcqOrderPrefTlv::TlvPresent`

8.135 NASLTEBandPreferenceTlv Struct Reference

Data Fields

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

8.135.1 Field Documentation

8.135.1.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

8.135.1.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

8.136 NASLteNasReleaseInfoTlv Struct Reference

Data Fields

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

8.136.1 Field Documentation

8.136.1.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

8.136.1.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

8.136.1.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

8.136.1.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

8.137 NASModePreferenceTlv Struct Reference

Data Fields

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

8.137.1 Field Documentation

8.137.1.1 uint16_t NASModePreferenceTlv::ModePref

8.137.1.2 uint8_t NASModePreferenceTlv::TlvPresent

8.138 NASNetSelPreferenceTlv Struct Reference

Data Fields

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

8.138.1 Field Documentation

8.138.1.1 `uint8_t NASNetSelPreferenceTlv::NetSelPref`

8.138.1.2 `uint8_t NASNetSelPreferenceTlv::TlvPresent`

8.139 NASOTAMessageTlv Struct Reference

Data Fields

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

8.139.1 Field Documentation

8.139.1.1 `uint8_t NASOTAMessageTlv::data_buf`[2048]

8.139.1.2 `uint16_t NASOTAMessageTlv::data_len`

8.139.1.3 `uint32_t NASOTAMessageTlv::message_type`

8.139.1.4 `uint8_t NASOTAMessageTlv::TlvPresent`

8.140 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`

8.140.1 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.

8.140.2 Field Documentation

8.140.2.1 `LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value`

8.140.2.2 `uint32_t NASPhyCaAggPcellInfo::freq`

8.140.2.3 `uint32_t NASPhyCaAggPcellInfo::ltebandValue`

8.140.2.4 `uint32_t NASPhyCaAggPcellInfo::pci`

8.140.2.5 `uint8_t NASPhyCaAggPcellInfo::TlvPresent`

8.141 NASPhyCaAggScellIDBw Struct Reference

Data Fields

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

8.141.1 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.
--------------------	--

8.141.2 Field Documentation

8.141.2.1 `LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellIDBw::dl_bw_value`

8.141.2.2 `uint8_t NASPhyCaAggScellIDBw::TlvPresent`

8.142 NASPhyCaAggScellIndex Struct Reference

Data Fields

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

8.142.1 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.

8.142.2 Field Documentation

8.142.2.1 `uint8_t NASPhyCaAggScellIndex::scell_idx`

8.142.2.2 `uint8_t NASPhyCaAggScellIndex::TlvPresent`

8.143 NASPhyCaAggScellIndType Struct Reference

Data Fields

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

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

8.143.2 Field Documentation

8.143.2.1 uint32_t NASPhyCaAggScellIndType::freq

8.143.2.2 uint32_t NASPhyCaAggScellIndType::pci

8.143.2.3 LIBPACK_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

8.143.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

8.144 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](#)

8.144.1 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.

8.144.2 Field Documentation

8.144.2.1 LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

8.144.2.2 uint32_t NASPhyCaAggScellInfo::freq

8.144.2.3 uint32_t NASPhyCaAggScellInfo::ltebandValue

8.144.2.4 uint32_t NASPhyCaAggScellInfo::pci

8.144.2.5 LIBPACK_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

8.144.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

8.145 NASPRLPreferenceTlv Struct Reference

Data Fields

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

8.145.1 Field Documentation

8.145.1.1 uint16_t NASPRLPreferenceTlv::PRLPref

8.145.1.2 uint8_t NASPRLPreferenceTlv::TlvPresent

8.146 NASQmiCbkJnasSwtOTAMessageInd Struct Reference

Data Fields

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

8.146.1 Field Documentation

8.146.1.1 [NASLteNasReleaseInfoTlv](#) NASQmiCbkJnasSwtOTAMessageInd::nasRelInfoTlv

8.146.1.2 [NASOTAMessageTlv](#) NASQmiCbkJnasSwtOTAMessageInd::otaMsgTlv

8.146.1.3 [NASTimeInfoTlv](#) NASQmiCbkJnasSwtOTAMessageInd::timeTlv

8.147 NASQmiCbkJnasSystemSelPrefInd Struct Reference

Data Fields

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

8.147.1 Field Documentation

- 8.147.1.1 **NASBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::BPTlv
- 8.147.1.2 **NASEmergencyModeTlv** NASQmiCbkNasSystemSelPrefInd::EMTlv
- 8.147.1.3 **NASGWAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::GWAOPTlv
- 8.147.1.4 **NASLTEBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::LBPTlv
- 8.147.1.5 **NASModePreferenceTlv** NASQmiCbkNasSystemSelPrefInd::MPTlv
- 8.147.1.6 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv
- 8.147.1.7 **NASPRLPReferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv
- 8.147.1.8 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv
- 8.147.1.9 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

8.148 NASRoamPreferenceTlv Struct Reference

Data Fields

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

8.148.1 Field Documentation

- 8.148.1.1 uint16_t NASRoamPreferenceTlv::RoamPref
- 8.148.1.2 uint8_t NASRoamPreferenceTlv::TlvPresent

8.149 NASServDomainPrefTlv Struct Reference

Data Fields

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

8.149.1 Field Documentation

- 8.149.1.1 uint32_t NASServDomainPrefTlv::SrvDomainPref
- 8.149.1.2 uint8_t NASServDomainPrefTlv::TlvPresent

8.150 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](#)

8.150.1 Detailed Description

This structure will hold the serving system parameters information

Parameters

<i>registrationState</i>	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - Type of selected radio access network <ul style="list-style-type: none"> • 0x00 - Unknown • 0x01 - 3GPP2 network • 0x02 - 3GPP network
<i>radioInterfaceNo</i>	<ul style="list-style-type: none"> - Number of radio interfaces currently in use; this indicates how many radio_if identifiers follow this field
<i>radioInterface-List</i>	<ul style="list-style-type: none"> - Radio interface currently in use (each is 1 byte) <ul style="list-style-type: none"> • 0x00 - None (no service) • 0x01 - cdma2000 1X • 0x02 - cdma2000 HRPD (1xEV-DO) • 0x03 - AMPS • 0x04 - GSM • 0x05 - UMTS • 0x08 - LTE
<i>hdrPersonality</i>	<ul style="list-style-type: none"> - HDR personality information (valid only for EVDO) <ul style="list-style-type: none"> • 0x00 - Unknown • 0x01 - HRPD • 0x02 - eHRPD

Note: None

8.150.2 Field Documentation

- 8.150.2.1 `uint8_t` `NASServingSystemInfo::csAttachState`
- 8.150.2.2 `uint8_t` `NASServingSystemInfo::hdrPersonality`
- 8.150.2.3 `uint8_t` `NASServingSystemInfo::psAttachState`
- 8.150.2.4 `uint8_t` `NASServingSystemInfo::radioInterfaceList[255]`
- 8.150.2.5 `uint8_t` `NASServingSystemInfo::radioInterfaceNo`
- 8.150.2.6 `uint8_t` `NASServingSystemInfo::registrationState`
- 8.150.2.7 `uint8_t` `NASServingSystemInfo::selectedNetwork`

8.151 NASTimeInfoTlv Struct Reference

Data Fields

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

8.151.1 Field Documentation

- 8.151.1.1 `uint64_t` `NASTimeInfoTlv::time`
- 8.151.1.2 `uint8_t` `NASTimeInfoTlv::TlvPresent`

8.152 newMTMessageTlv Struct Reference

Data Fields

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

8.152.1 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

8.152.2 Field Documentation

8.152.2.1 `sMSMTMessageInfo newMTMessageTlv::MTMessageInfo`

8.152.2.2 `uint8_t newMTMessageTlv::TlvPresent`

8.153 `pack_dms_GetCustFeaturesV2_t` Struct Reference

Data Fields

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

8.153.1 Detailed Description

This structure contains which customization id or the list type want to retrieve from modem. This TLV is only applicable for 9x30 modules so far

Parameters

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

8.153.2 Field Documentation

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

8.153.2.2 `uint8_t pack_dms_GetCustFeaturesV2_t::list_type`

8.153.2.3 `uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult`

8.154 `pack_dms_SetCrashAction_t` Struct Reference

Data Fields

- `uint8_t crashAction`

8.154.1 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
--------------------	---

8.154.2 Field Documentation

8.154.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

8.155 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](#)

8.155.1 Field Documentation

8.155.1.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

8.155.1.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

8.155.1.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

8.155.1.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

8.155.1.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

8.155.1.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

8.155.1.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

8.155.1.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

8.155.1.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

8.156 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](#)

8.156.1 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

8.156.2 Field Documentation

8.156.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

8.156.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

8.156.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

8.156.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

8.157 pack_dms_SetEventReport_t Struct Reference

Data Fields

- uint8_t [mode](#)

8.157.1 Field Documentation

8.157.1.1 uint8_t pack_dms_SetEventReport_t::mode

8.158 pack_dms_SetPower_t Struct Reference

Data Fields

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

8.158.1 Field Documentation

8.158.1.1 uint32_t pack_dms_SetPower_t::mode

8.158.1.2 uint16_t pack_dms_SetPower_t::Tlvresult

8.159 pack_dms_SetUSBComp_t Struct Reference

Data Fields

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

8.159.1 Field Documentation

8.159.1.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

8.159.1.2 uint8_t pack_dms_SetUSBComp_t::USBComp

8.160 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint8_t [resetInfoInd](#)

8.160.1 Detailed Description

Parameters

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

8.160.2 Field Documentation

8.160.2.1 uint8_t pack_dms_SLQSDmsSwiIndicationRegister_t::resetInfoInd

8.161 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

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

8.161.1 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.
-----------------------------	--

8.161.2 Field Documentation

8.161.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

8.161.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

8.162 pack_dms_UIMGetICCID_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.162.1 Detailed Description

This structure contains UIM Get ICCID pack

Parameters

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

8.162.2 Field Documentation

8.162.2.1 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

8.163 pack_fms_GetImagesPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.163.1 Detailed Description

This structure contains the Get Image Preference information pack

Parameters

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

8.163.2 Field Documentation

8.163.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

8.164 pack_fms_GetStoredImages_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.164.1 Detailed Description

This structure contains the Get Stored Images pack

Parameters

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

8.164.2 Field Documentation

8.164.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

8.165 pack_fms_SetImagesPreference_t Struct Reference

Data Fields

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

8.165.1 Detailed Description

This structure contains the Set Images Preference pack

Parameters

<i>imageListSize</i>	<ul style="list-style-type: none">• Image List Size
<i>pImageList</i>	<ul style="list-style-type: none">• Image List• See FMSPrefImageList
<i>bForceDownload</i>	<ul style="list-style-type: none">• 0 - Not Force Download.• 1 - Force Download.
<i>modemindex</i>	<ul style="list-style-type: none">• Modem Index.
<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack result

8.165.2 Field Documentation

8.165.2.1 `uint32_t pack_fms_SetImagesPreference_t::bForceDownload`

8.165.2.2 `uint32_t pack_fms_SetImagesPreference_t::imageListSize`

8.165.2.3 `uint8_t pack_fms_SetImagesPreference_t::modemindex`

8.165.2.4 `FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList`

8.165.2.5 `uint16_t pack_fms_SetImagesPreference_t::Tlvresult`

8.166 `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`

8.166.1 Detailed Description

This structure contains LOC delete assist data pack

Parameters

<i>pSVInfo</i>	<ul style="list-style-type: none"> • Pointer to struct <code>loc_SVInfo</code>. See <code>loc_SVInfo</code> for more information
<i>pGnssData</i>	<ul style="list-style-type: none"> • Pointer to struct <code>loc_GnssData</code>. See <code>loc_GnssData</code> for more information
<i>pCellDb</i>	<ul style="list-style-type: none"> • Pointer to struct <code>loc_CellDb</code>. See <code>loc_CellDb</code> for more information
<i>pClkInfo</i>	<ul style="list-style-type: none"> • Pointer to struct <code>loc_ClkInfo</code>. See <code>loc_ClkInfo</code> for more information
<i>pBdsSVInfo</i>	<ul style="list-style-type: none"> • Pointer to struct <code>loc_BdsSVInfo</code>. See <code>loc_BdsSVInfo</code> for more information
<i>Tlvresult</i>	<ul style="list-style-type: none"> • Pack delete assist data request result.

8.166.2 Field Documentation

8.166.2.1 `loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo`

8.166.2.2 `loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb`

8.166.2.3 loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo

8.166.2.4 loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData

8.166.2.5 loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo

8.166.2.6 uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult

8.167 pack_loc_EventRegister_t Struct Reference

Data Fields

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

8.167.1 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.
------------------	--

8.167.2 Field Documentation

8.167.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

8.167.2.2 uint16_t pack_loc_EventRegister_t::Tlvresult

8.168 pack_loc_SetExtPowerState_t Struct Reference

Data Fields

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

8.168.1 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.

8.168.2 Field Documentation

8.168.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

8.168.2.2 uint16_t pack_loc_SetExtPowerState_t::Tlvresult

8.169 pack_loc_SetOperationMode_t Struct Reference

Data Fields

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

8.169.1 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.

8.169.2 Field Documentation

8.169.2.1 uint32_t pack_loc_SetOperationMode_t::mode

8.169.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

8.170 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

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

8.170.1 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.

8.170.2 Field Documentation

8.170.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.170.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

8.171 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](#)

8.171.1 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

8.171.2 Field Documentation

8.171.2.1 `loc_LocApplicationInfo*` `pack_loc_Start_t::pApplicationInfo`

8.171.2.2 `uint32_t*` `pack_loc_Start_t::pConfigAltitudeAssumed`

8.171.2.3 `uint32_t*` `pack_loc_Start_t::pHorizontalAccuracyLvl`

8.171.2.4 `uint32_t*` `pack_loc_Start_t::pIntermediateReportState`

8.171.2.5 `uint32_t*` `pack_loc_Start_t::pMinIntervalTime`

8.171.2.6 `uint32_t*` `pack_loc_Start_t::pRecurrenceType`

8.171.2.7 `uint8_t` `pack_loc_Start_t::SessionId`

8.171.2.8 `uint16_t` `pack_loc_Start_t::Tlvresult`

8.172 `pack_loc_Stop_t` Struct Reference

Data Fields

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

8.172.1 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.

8.172.2 Field Documentation

8.172.2.1 `uint8_t` `pack_loc_Stop_t::SessionId`

8.172.2.2 `uint16_t` `pack_loc_Stop_t::Tlvresult`

8.173 `pack_nas_SetACCOLC_t` Struct Reference

Data Fields

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

8.173.1 Detailed Description

Parameters

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

8.173.2 Field Documentation

8.173.2.1 `uint8_t pack_nas_SetACCOLC_t::accolc`

8.173.2.2 `int8_t pack_nas_SetACCOLC_t::spc[6]`

8.174 pack_nas_SetNetworkPreference_t Struct Reference

Data Fields

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

8.174.1 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

8.174.2 Field Documentation

8.174.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

8.174.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

8.174.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

8.175 pack_nas_SLQSGetPLMNName_t Struct Reference

Data Fields

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

8.175.1 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.

8.175.2 Field Documentation

8.175.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

8.175.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

8.175.2.3 uint8_t* pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

8.176 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

Data Fields

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

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

8.176.1 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.

8.176.2 Field Documentation

8.176.2.1 uint32_t* pack_nas_SLQSInitiateNetworkRegistration_t::pChangeDuration

8.176.2.2 uint8_t* pack_nas_SLQSInitiateNetworkRegistration_t::pMncPcsDigitStatus

8.176.2.3 nas_MNRInfo* pack_nas_SLQSInitiateNetworkRegistration_t::pMNRInfo

8.176.2.4 uint32_t pack_nas_SLQSInitiateNetworkRegistration_t::regAction

8.177 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

8.177.1 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>pHdRECIO- Thresh</i>	<ul style="list-style-type: none"> HDR ECIO Threshold List
<i>pHdRECIODelta</i>	<ul style="list-style-type: none"> ECIO delta (in units of 0.1 dB) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pHdRSINR- Thresh</i>	<ul style="list-style-type: none"> HDR SINR Threshold List
<i>pHdRSINRDelta</i>	<ul style="list-style-type: none"> SINR delta (in units of 1 SINR level) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pHdRIOTresh</i>	<ul style="list-style-type: none"> HDR IO Threshold List
<i>pHdRIODelta</i>	<ul style="list-style-type: none"> IO delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pGSMRSSI- Thresh</i>	<ul style="list-style-type: none"> GSM RSSI Threshold List See GSMRSSIThresh for more details
<i>pGSMRSSIDelta</i>	<ul style="list-style-type: none"> RSSI delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pWCDMARSSI- Thresh</i>	<ul style="list-style-type: none"> WCDMA RSSI Threshold List See WCDMARSSIThresh for more details
<i>pWCDMARSSI- Delta</i>	<ul style="list-style-type: none"> RSSI delta (in units of 0.1 dBm). A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pWCDMAECIO- Thresh</i>	<ul style="list-style-type: none"> WCDMA ECIO Threshold List
<i>pWCDMAECIO- Delta</i>	<ul style="list-style-type: none"> ECIO delta (in units of 0.1 dB) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSSI- Thresh</i>	<ul style="list-style-type: none"> LTE RSSI Threshold List
<i>pLTERSSIDelta</i>	<ul style="list-style-type: none"> RSSI delta (in units of 0.1 dBm) A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.

<i>pLTERSNR- Thresh</i>	<ul style="list-style-type: none"> • LTE SNR Threshold List
<i>pLTERSNRDelta</i>	<ul style="list-style-type: none"> • SNR delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSRQ- Thresh</i>	<ul style="list-style-type: none"> • LTE RSRQ Threshold List
<i>pLTERSRQ- Delta</i>	<ul style="list-style-type: none"> • RSRQ delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSRP- Thresh</i>	<ul style="list-style-type: none"> • LTE RSRP Threshold List
<i>pLTERSRPDelta</i>	<ul style="list-style-type: none"> • RSRP delta (in units of 0.1 dBm). • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pLTERSigRpt- Config</i>	<ul style="list-style-type: none"> • LTE Signal Report Config
<i>pTDSCDMARS- CPThresh</i>	<ul style="list-style-type: none"> • TDSCDMA RSCP Threshold List
<i>pTDSCDMARS- CPDelta</i>	<ul style="list-style-type: none"> • RSCP delta (in units of 0.1 dBm) • A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.
<i>pTDSCDMARS- SIThresh</i>	<ul style="list-style-type: none"> • TDSCDMA RSSI Threshold List
<i>pTDSCDMARS- SIDelta</i>	<ul style="list-style-type: none"> • RSSI delta (in dBm) used by TD-SCDMA.
<i>pTDSCDMAECI- OThresh</i>	<ul style="list-style-type: none"> • TDSCDMA ECIO Threshold List
<i>pTDSCDMAECI- ODelta</i>	<ul style="list-style-type: none"> • ECIO delta (in dB) used by TD-SCDMA
<i>pTDSCDMASIN- RThresh</i>	<ul style="list-style-type: none"> • TDSCDMA SINR Threshold List
<i>pTDSCDMASIN- RDelta</i>	<ul style="list-style-type: none"> • SINR delta (in dB) used by TD-SCDMA.

8.177.2 Field Documentation

- 8.177.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta
- 8.177.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh
- 8.177.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta
- 8.177.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh
- 8.177.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta
- 8.177.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh
- 8.177.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta
- 8.177.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh
- 8.177.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta
- 8.177.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh
- 8.177.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIDelta
- 8.177.2.12 nas_HDRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIThresh
- 8.177.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRDelta
- 8.177.2.14 nas_HDRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRThresh
- 8.177.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta
- 8.177.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh
- 8.177.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta
- 8.177.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh
- 8.177.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta
- 8.177.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh
- 8.177.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig
- 8.177.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta
- 8.177.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh
- 8.177.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta
- 8.177.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh
- 8.177.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta
- 8.177.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh
- 8.177.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

- 8.177.2.29 `nas_TDSCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh`
- 8.177.2.30 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta`
- 8.177.2.31 `nas_TDSCDMASINRThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh`
- 8.177.2.32 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIODelta`
- 8.177.2.33 `nas_WCDMAECIOThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh`
- 8.177.2.34 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta`
- 8.177.2.35 `nas_WCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh`

8.178 `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](#)

8.178.1 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. <p>tFNBandPreference</p> <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. <p>tFNDualStandByPref</p> <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. <p>tFNSubscriptionInfo</p> <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. <p>tFNNetworkTime</p> <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. <p>tFNSysInfo</p> <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. <p>tFNSigInfo</p> <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. <p>tFNErrRate</p> <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

8.178.2 Field Documentation

8.178.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

8.178.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

8.178.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

8.178.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

8.178.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

8.178.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

8.178.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

8.178.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

8.178.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

8.178.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

8.178.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

8.178.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

8.178.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

8.179 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](#)

8.179.1 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

8.179.2 Field Documentation

- 8.179.2.1 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsDI`
- 8.179.2.2 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsUI`
- 8.179.2.3 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmDI`
- 8.179.2.4 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmUI`
- 8.179.2.5 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmDI`
- 8.179.2.6 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmUI`
- 8.179.2.7 `uint8_t* pack_nas_SLQSNasSwiOTAMessageCallback_t::pRankIndicatorInd`

8.180 `pack_nas_SLQSSetSignalStrengthsCallback_t` Struct Reference

Data Fields

- `uint8_t bEnable`
- `nas_SLQSSignalStrengthsIndReq * pSigIndReq`

8.180.1 Detailed Description

Parameters

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

8.180.2 Field Documentation

- 8.180.2.1 `uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable`
- 8.180.2.2 `nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq`

8.181 `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`

8.181.1 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

8.181.2 Field Documentation

8.181.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

8.181.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

8.181.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

8.181.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

8.181.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

8.181.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

- 8.181.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref
- 8.181.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat
- 8.181.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref
- 8.181.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref
- 8.181.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref
- 8.181.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT
- 8.181.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref
- 8.181.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref
- 8.181.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction
- 8.181.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdsdmaBandPref

8.182 pack_qmi_t Struct Reference

Data Fields

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

8.182.1 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

8.182.2 Field Documentation

- 8.182.2.1 uint16_t pack_qmi_t::msgid
- 8.182.2.2 uint8_t pack_qmi_t::svc
- 8.182.2.3 int pack_qmi_t::timeout
- 8.182.2.4 uint16_t pack_qmi_t::xid

8.183 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.183.1 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
-------------------	--

8.183.2 Field Documentation

8.183.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.184 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

Data Fields

- uint32_t [apnId](#)

8.184.1 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
-------------------	--

8.184.2 Field Documentation

8.184.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

8.185 pack_qos_SLQSSetQosEventCallback_t Struct Reference

Data Fields

- uint8_t [enable](#)

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

8.185.2 Field Documentation

8.185.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

8.186 pack_sms_SendSMS_t Struct Reference

Data Fields

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

8.186.1 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

8.186.2 Field Documentation

8.186.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

8.186.2.2 uint32_t pack_sms_SendSMS_t::messageSize

8.186.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

8.186.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

8.187 pack_sms_SetNewSMSCallback_t Struct Reference

Data Fields

- enum [eqmiCbkJetStatus status](#)

8.187.1 Detailed Description

Parameters

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

8.187.2 Field Documentation

8.187.2.1 enum [eqmiCbkJetStatus pack_sms_SetNewSMSCallback_t::status](#)

8.188 pack_sms_SLQSDelateSMS_t Struct Reference

Data Fields

- uint32_t [storageType](#)
- uint32_t * [pMessageIndex](#)
- uint32_t * [pMessageTag](#)
- 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>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)

8.188.2 Field Documentation

8.188.2.1 uint32_t* pack_sms_SLQSDelSDeleteSMS_t::pMessageIndex

8.188.2.2 uint8_t* pack_sms_SLQSDelSDeleteSMS_t::pMessageMode

8.188.2.3 uint32_t* pack_sms_SLQSDelSDeleteSMS_t::pMessageTag

8.188.2.4 uint32_t pack_sms_SLQSDelSDeleteSMS_t::storageType

8.189 pack_sms_SLQSGetSMS_t Struct Reference

Data Fields

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

8.189.1 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)

8.189.2 Field Documentation

8.189.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

8.189.2.2 uint8_t* pack_sms_SLQSGetSMS_t::pMessageMode

8.189.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

8.190 pack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

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

8.190.1 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)

8.190.2 Field Documentation

8.190.2.1 `uint8_t*` `pack_sms_SLQSGetSMSList_t::pMessageMode`8.190.2.2 `uint32_t*` `pack_sms_SLQSGetSMSList_t::pRequestedTag`8.190.2.3 `uint32_t` `pack_sms_SLQSGetSMSList_t::storageType`8.191 `pack_sms_SLQSMModifySMSStatus_t` Struct Reference

Data Fields

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

8.191.1 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)
---------------------	--

8.191.2 Field Documentation

8.191.2.1 uint32_t pack_sms_SLQSMModifySMSStatus_t::messageIndex

8.191.2.2 uint32_t pack_sms_SLQSMModifySMSStatus_t::messageTag

8.191.2.3 uint8_t* pack_sms_SLQSMModifySMSStatus_t::pMessageMode

8.191.2.4 uint32_t pack_sms_SLQSMModifySMSStatus_t::storageType

8.192 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](#)

8.192.1 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

8.192.2 Field Documentation

8.192.2.1 `uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate`

8.192.2.2 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type`

8.192.2.3 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::function`

8.192.2.4 `uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist`

8.192.2.5 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time`

8.192.2.6 `int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate`

8.192.2.7 `int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type`

8.192.2.8 `int pack_swiloc_SwiLocSetAutoStart_t::set_function`

8.192.2.9 `int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist`

8.192.2.10 `int pack_swiloc_SwiLocSetAutoStart_t::set_max_time`

8.193 `pack_swioama_SLQSOMADMCancelSession_t` Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.193.1 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
------------------------	--

8.193.2 Field Documentation

8.193.2.1 uint32_t pack_swisma_SLQSOMADMCancelSession_t::sessionType

8.194 pack_swisma_SLQSOMADMGetSessionInfo_t Struct Reference

Data Fields

- uint32_t [SessionType](#)

8.194.1 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
------------------------	--

8.194.2 Field Documentation

8.194.2.1 uint32_t pack_swisma_SLQSOMADMGetSessionInfo_t::SessionType

8.195 pack_swisma_SLQSOMADMSendSelection_t Struct Reference

Data Fields

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

8.195.1 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.

8.195.2 Field Documentation

8.195.2.1 uint32_t* pack_swioama_SLQSOMADMSendSelection_t::pDeferTime

8.195.2.2 uint32_t* pack_swioama_SLQSOMADMSendSelection_t::pRejectReason

8.195.2.3 uint32_t pack_swioama_SLQSOMADMSendSelection_t::selection

8.196 pack_swioama_SLQSOMADMSetSettings_t Struct Reference

Data Fields

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

8.196.1 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

8.196.2 Field Documentation

8.196.2.1 uint8_t pack_swisma_SLQSOMADMSetSettings_t::FOTAdownload

8.196.2.2 uint8_t pack_swisma_SLQSOMADMSetSettings_t::FOTAUpdate

8.196.2.3 uint8_t* pack_swisma_SLQSOMADMSetSettings_t::pAutosdm

8.196.2.4 uint8_t* pack_swisma_SLQSOMADMSetSettings_t::pFwAutoCheck

8.197 pack_swisma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [sessionType](#)

8.197.1 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
------------------------	---

8.197.2 Field Documentation

8.197.2.1 uint32_t pack_swisma_SLQSOMADMStartSession_t::sessionType

8.198 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](#)

8.198.1 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.

8.198.2 Field Documentation

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

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

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

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

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

8.198.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

8.199 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](#)

8.199.1 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.

8.199.2 Field Documentation

8.199.2.1 uim_fileInfo pack_uim_ReadTransparent_t::fileIndex

8.199.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

8.199.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

8.199.2.4 `uim_readTransparentInfo` `pack_uim_ReadTransparent_t::readTransparent`

8.199.2.5 `uim_sessionInformation` `pack_uim_ReadTransparent_t::sessionInfo`

8.199.2.6 `uint16_t` `pack_uim_ReadTransparent_t::Tlvresult`

8.200 `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`

8.200.1 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 <code>uim_sessionInformation</code> for more information.
<i>pinProtection</i>	<ul style="list-style-type: none"> • See <code>uim_setPINProtection</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.

8.200.2 Field Documentation

8.200.2.1 `uim_encryptedPIN1` `pack_uim_SetPinProtection_t::EncryptedPIN1`

8.200.2.2 `uint32_t*` `pack_uim_SetPinProtection_t::pIndicationToken`

8.200.2.3 uim_setPINProtection pack_uim_SetPinProtection_t::pinProtection

8.200.2.4 uint8_t* pack_uim_SetPinProtection_t::pKeyReferenceID

8.200.2.5 uim_sessionInformation pack_uim_SetPinProtection_t::sessionInfo

8.200.2.6 uint16_t pack_uim_SetPinProtection_t::Tlvresult

8.201 pack_uim_SLQSUIMEventRegister_t Struct Reference

Data Fields

- uint32_t [eventMask](#)

8.201.1 Detailed Description

Parameters

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

8.201.2 Field Documentation

8.201.2.1 uint32_t pack_uim_SLQSUIMEventRegister_t::eventMask

8.202 pack_uim_SLQSUIMPowerDown_t Struct Reference

Data Fields

- uint8_t [slot](#)

8.202.1 Detailed Description

This structure contains information of the request parameters associated with a Power Down.

Parameters

<i>slot</i>	<ul style="list-style-type: none">• Indicates the slot to be used.<ul style="list-style-type: none">– 1 - Slot 1– 2 - Slot 2
-------------	---

8.202.2 Field Documentation

8.202.2.1 uint8_t pack_uim_SLQSUIMPowerDown_t::slot

8.203 pack_uim_SLQSUIMPowerUp_t Struct Reference

Data Fields

- uint8_t [slot](#)
- uint8_t * [plgnoreHotSwapSwitch](#)

8.203.1 Detailed Description

This structure contains information of the request parameters associated with a Power Down.

Parameters

<i>slot</i>	<ul style="list-style-type: none"> • Indicates the slot to be used. <ul style="list-style-type: none"> – 1 - Slot 1 – 2 - Slot 2
<i>plgnoreHot-Swap-Switch(optional)</i>	<ul style="list-style-type: none"> • Hot-swap switch status. <ul style="list-style-type: none"> – 0 - Checks the hot-swap switch status – 1 - Ignores the hot-swap switch status

8.203.2 Field Documentation

8.203.2.1 uint8_t* [pack_uim_SLQSUIMPowerUp_t::plgnoreHotSwapSwitch](#)

8.203.2.2 uint8_t [pack_uim_SLQSUIMPowerUp_t::slot](#)

8.204 pack_uim_SLQSUIMSwitchSlot_t Struct Reference

Data Fields

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

8.204.1 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
-----------------------	--

8.204.2 Field Documentation

8.204.2.1 `uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot`

8.204.2.2 `uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot`

8.205 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`

8.205.1 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
-----------------------------------	---

8.205.2 Field Documentation

8.205.2.1 `uim_encryptedPIN1` `pack_uim_UnblockPin_t::EncryptedPIN1`

8.205.2.2 `uint32_t*` `pack_uim_UnblockPin_t::pIndicationToken`

8.205.2.3 `uim_unblockUIMPIN` `pack_uim_UnblockPin_t::pinProtection`

8.205.2.4 `uint8_t*` `pack_uim_UnblockPin_t::pKeyReferenceID`

8.205.2.5 `uim_sessionInformation` `pack_uim_UnblockPin_t::sessionInfo`

8.205.2.6 `uint16_t` `pack_uim_UnblockPin_t::Tlvresult`

8.206 `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`

8.206.1 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.

8.206.2 Field Documentation

8.206.2.1 uim_encryptedPIN1* pack_uim_VerifyPin_t::pEncryptedPIN1

8.206.2.2 uint32_t* pack_uim_VerifyPin_t::pIndicationToken

8.206.2.3 uint8_t* pack_uim_VerifyPin_t::pKeyReferenceID

8.206.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

8.206.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

8.206.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

8.207 pack_wds_GetDefaultProfile_t Struct Reference**Data Fields**

- uint32_t [profiletype](#)

8.207.1 Detailed Description**Parameters**

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

8.207.2 Field Documentation

8.207.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

8.208 pack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

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

8.208.1 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

8.208.2 Field Documentation

8.208.2.1 [uint8_t pack_wds_GetDefaultProfileNum_t::family](#)

8.208.2.2 [uint8_t pack_wds_GetDefaultProfileNum_t::type](#)

8.209 [pack_wds_GetDormancyState_t](#) Struct Reference

8.210 [pack_wds_GetLastMobileIPError_t](#) Struct Reference

8.211 [pack_wds_GetMobileIP_t](#) Struct Reference

8.212 [pack_wds_GetMobileIPProfile_t](#) Struct Reference

Data Fields

- [uint8_t index](#)

8.212.1 Detailed Description

Parameters

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

8.212.2 Field Documentation

8.212.2.1 [uint8_t pack_wds_GetMobileIPProfile_t::index](#)

8.213 [pack_wds_GetPacketStatistics_t](#) Struct Reference

Data Fields

- uint32_t * [pStatMask](#)

8.213.1 Detailed Description

Parameters

<i>pStatMask</i>	<ul style="list-style-type: none"> • Packet Statistics Mask 0x00000001 - Tx packets OK 0x00000002 - Rx packets OK 0x00000004 - Tx packet errors 0x00000008 - Rx packet errors 0x00000010 - Tx overflows 0x00000020 - Rx overflows 0x00000040 - Tx bytes OK 0x00000080 - Rx bytes OK
------------------	--

8.213.2 Field Documentation

8.213.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

8.214 pack_wds_GetPacketStatus_t Struct Reference

Data Fields

- uint32_t [statmask](#)

8.214.1 Detailed Description

Parameters

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

8.214.2 Field Documentation

8.214.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

8.215 pack_wds_GetSessionDuration_t Struct Reference

8.216 pack_wds_RMSetTransferStatistics_t Struct Reference

Data Fields

- [rmTrasnferStaticsReq](#) [RmTrasnferStaticsReq](#)

8.216.1 Detailed Description

Parameters

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

8.216.2 Field Documentation

8.216.2.1 `rmTrasnferStaticsReq` `pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq`

8.217 `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](#)

8.217.1 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

8.217.2 Field Documentation

8.217.2.1 `uint32_t` `pack_wds_SetDefaultProfile_t::authentication`

8.217.2.2 `uint32_t` `pack_wds_SetDefaultProfile_t::ipAddress`

8.217.2.3 `uint8_t*` `pack_wds_SetDefaultProfile_t::pApnname`

8.217.2.4 `uint32_t` `pack_wds_SetDefaultProfile_t::pdpType`

8.217.2.5 `uint8_t*` `pack_wds_SetDefaultProfile_t::pName`

8.217.2.6 `uint8_t*` `pack_wds_SetDefaultProfile_t::pPassword`

8.217.2.7 `uint32_t` `pack_wds_SetDefaultProfile_t::primaryDNS`

8.217.2.8 `uint32_t` `pack_wds_SetDefaultProfile_t::profileType`

8.217.2.9 `uint8_t*` `pack_wds_SetDefaultProfile_t::pUsername`

8.217.2.10 `uint32_t` `pack_wds_SetDefaultProfile_t::secondaryDNS`

8.218 pack_wds_SetDefaultProfileNum_t Struct Reference

Data Fields

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

8.218.1 Field Documentation

8.218.1.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

8.218.1.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

8.218.1.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

8.219 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](#)

8.219.1 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

8.219.2 Field Documentation

- 8.219.2.1 `uint8_t pack_wds_SetMobileIPProfile_t::index`
- 8.219.2.2 `uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI`
- 8.219.2.3 `uint32_t* pack_wds_SetMobileIPProfile_t::pAddress`
- 8.219.2.4 `uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled`
- 8.219.2.5 `uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI`
- 8.219.2.6 `int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA`
- 8.219.2.7 `int8_t* pack_wds_SetMobileIPProfile_t::pMNHA`
- 8.219.2.8 `int8_t* pack_wds_SetMobileIPProfile_t::pNAI`
- 8.219.2.9 `uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA`
- 8.219.2.10 `uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling`
- 8.219.2.11 `uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA`
- 8.219.2.12 `int8_t pack_wds_SetMobileIPProfile_t::spc[10]`

8.220 `pack_wds_SLQSCreateProfile_t` Struct Reference

Data Fields

- `uint8_t * pProfileId`
- `uint8_t * pProfileType`
- `wds_profileInfo * pCurProfile`

8.220.1 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

8.220.2 Field Documentation

8.220.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

8.220.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

8.220.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

8.221 pack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

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

8.221.1 Detailed Description

Parameters

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

8.221.2 Field Documentation

8.221.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

8.221.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

8.222 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

8.223 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

8.224 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](#)

8.224.1 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

8.224.2 Field Documentation

8.224.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

8.224.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

8.224.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

8.224.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

8.224.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

8.224.2.6 transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

8.225 pack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

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

8.225.1 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

8.225.2 Field Documentation

8.225.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

8.225.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

8.226 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

Data Fields

- uint32_t * [pReqSettings](#)

8.226.1 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
---------------------	--

8.226.2 Field Documentation

8.226.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

8.227 pack_wds_SLQSModifyProfile_t Struct Reference

Data Fields

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

8.227.1 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

8.227.2 Field Documentation

8.227.2.1 [wds_profileInfo](#) pack_wds_SLQSMModifyProfile_t::curProfile

8.227.2.2 uint8_t* pack_wds_SLQSMModifyProfile_t::pProfileId

8.227.2.3 uint8_t* pack_wds_SLQSMModifyProfile_t::pProfileType

8.228 pack_wds_SLQSSet3GPPConfigItem_t Struct Reference

Data Fields

- uint16_t * [pLTEAttachProfile](#)
- uint16_t * [pProfileList](#)
- uint8_t * [pDefaultPDNEnabled](#)
- uint8_t * [p3gppRelease](#)
- uint16_t * [pLTEAttachProfileList](#)
- uint16_t [LTEAttachProfileListLen](#)

8.228.1 Detailed Description

Parameters

<i>pLTEAttach-Profile</i>	<ul style="list-style-type: none"> • Optional parameter • LTE Attach Profile <ul style="list-style-type: none"> – points to a single WORD Value indicating the attached LTE Profile – Optional parameter with possible values 1-16 (EM/MC73xx or earlier) • This setting is deprecated on MC/EM74xx
<i>ProfileList</i>	<p>Profile List</p> <ul style="list-style-type: none"> • an array of 4 profile configurations • Each element points to a single WORD value indicating profile • Optional parameter with possible values <ul style="list-style-type: none"> – 1 - 16 (MC/EM73xx and before) – 1 - 24 (MC/EM74xx and onwards) • function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present
<i>pDefaultPDN-Enabled</i>	<ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – 0 - disabled – 1 - enabled
<i>p3gppRelease</i>	<p>3GPP release</p> <ul style="list-style-type: none"> • Optional parameter <ul style="list-style-type: none"> – 0 - Release_99 – 1 - Release_5 – 2 - Release_6 – 3 - Release_7 – 4 - Release_8 • In 9x30 and onwards <ul style="list-style-type: none"> – 5 - Release 9 – 6 - Release 10 – 7 - Release 11
<i>pLTEAttach-ProfileList</i>	<ul style="list-style-type: none"> • pointer to WORD array indicating LTE Attach Profile List <ul style="list-style-type: none"> – Optional parameter – possible values: 1-24 – This setting is only supported for MC/EM74xx onwards – Please provide attach profiles in order of decreasing priority in this list.
<i>LTEAttach-ProfileListLen</i>	<ul style="list-style-type: none"> • Number of element in pLTEAttachProfileList <ul style="list-style-type: none"> – valid range: 1-24 – This setting is only supported for MC/EM74xx onwards

8.228.2 Field Documentation

8.228.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

8.228.2.2 `uint8_t*` `pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease`

8.228.2.3 `uint8_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled`

8.228.2.4 `uint16_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile`

8.228.2.5 `uint16_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList`

8.228.2.6 `uint16_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pProfileList`

8.229 `pack_wds_SLQSSetIPFamilyPreference_t` Struct Reference

Data Fields

- `uint8_t` [IPFamilyPreference](#)

8.229.1 Detailed Description

Parameters

<i>IPFamily-Preference</i>	IP Family preference <ul style="list-style-type: none"> • <code>PACK_WDS_IPV4</code> IP Version 4 • <code>PACK_WDS_IPV6</code> IP Version 6
----------------------------	---

8.229.2 Field Documentation

8.229.2.1 `uint8_t` `pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference`

8.230 `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](#)

8.230.1 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

8.230.2 Field Documentation

8.230.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

8.230.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

8.230.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

8.230.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

8.230.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

8.230.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

8.230.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

8.231 pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

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

8.231.1 Detailed Description

Parameters

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

8.231.2 Field Documentation

8.231.2.1 [wdsDhcpv4ProfileId](#)* [pack_wds_SLQSSetDHCPv4ClientConfig_t::pProfileId](#)

8.232 pack_wds_SLQSSetLoopback_t Struct Reference

Data Fields

- uint8_t [loopbackMode](#)
- uint8_t [loopbackMultiplier](#)

8.232.1 Detailed Description

Parameters

<i>loopbackMode</i>	<ul style="list-style-type: none"> • Loopback Mode. <ul style="list-style-type: none"> – 0 - Disable – 1 - Enable
<i>loopback-Multiplier</i>	<ul style="list-style-type: none"> • Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.232.2 Field Documentation

8.232.2.1 `uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode`

8.232.2.2 `uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier`

8.233 `pack_wds_SLQSStartDataSession_t` Struct Reference

Data Fields

- `uint8_t * pTech`
- `uint32_t * pprofileid3gpp`
- `uint32_t * pprofileid3gpp2`
- `uint32_t * pAuth`
- `char * pUser`
- `char * pPass`

8.233.1 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

8.233.2 Field Documentation

8.233.2.1 uint32_t* pack_wds_SLQSSStartDataSession_t::pAuth

8.233.2.2 char* pack_wds_SLQSSStartDataSession_t::pPass

8.233.2.3 uint32_t* pack_wds_SLQSSStartDataSession_t::pprofileid3gpp

8.233.2.4 uint32_t* pack_wds_SLQSSStartDataSession_t::pprofileid3gpp2

8.233.2.5 uint8_t* pack_wds_SLQSSStartDataSession_t::pTech

8.233.2.6 char* pack_wds_SLQSSStartDataSession_t::pUser

8.234 pack_wds_SLQSSStopDataSession_t Struct Reference

Data Fields

- uint32_t * [psid](#)

8.234.1 Detailed Description

Parameters

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

8.234.2 Field Documentation

8.234.2.1 uint32_t* pack_wds_SLQSSStopDataSession_t::psid

8.235 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

Data Fields

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

8.235.1 Detailed Description

Parameters

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

8.235.2 Field Documentation

8.235.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

8.235.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

8.236 PackCreateProfileOut Struct Reference

Data Fields

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

8.236.1 Field Documentation

8.236.1.1 uint16_t PackCreateProfileOut::ExtErrorCode

8.236.1.2 uint8_t PackCreateProfileOut::ProfileIndex

8.236.1.3 uint8_t PackCreateProfileOut::ProfileType

8.237 packgetDyingGaspCfg Struct Reference

Data Fields

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

8.237.1 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 CContent as a string of 8 bit ASCII text characters Max 160 chars. • Optional parameter.

8.237.2 Field Documentation

8.237.2.1 uint8_t* packgetDyingGaspCfg::pDestSMSContent

8.237.2.2 uint8_t* packgetDyingGaspCfg::pDestSMSNum

8.238 packgetDyingGaspStatistics Struct Reference

Data Fields

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

8.238.1 Detailed Description

Parameters

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

8.238.2 Field Documentation

8.238.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.238.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

8.239 qmiSmsMessageList Struct Reference

Data Fields

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

8.239.1 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"> Message tag

8.239.2 Field Documentation

8.239.2.1 uint32_t qmiSmsMessageList::messageIndex

8.239.2.2 uint32_t qmiSmsMessageList::messageTag

8.240 qmiWDSDataBearerTechnology Struct Reference

Data Fields

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

8.240.1 Detailed Description

Parameters

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

8.240.2 Field Documentation

8.240.2.1 `uint8_t qmiWSDDataBearerTechnology::currentNetwork`

8.240.2.2 `uint32_t qmiWSDDataBearerTechnology::ratMask`

8.240.2.3 `uint32_t qmiWSDDataBearerTechnology::soMask`

8.241 RFBandInfoElements Struct Reference

Data Fields

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

8.241.1 Detailed Description

Parameters

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

8.241.2 Field Documentation

8.241.2.1 `uint16_t RFBandInfoElements::activeBandClass`

8.241.2.2 `uint16_t RFBandInfoElements::activeChannel`

8.241.2.3 `uint8_t RFBandInfoElements::radioInterface`

8.242 rmTrasnferStaticsReq Struct Reference

Data Fields

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

8.242.1 Detailed Description

Parameters

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

8.242.2 Field Documentation

8.242.2.1 `uint8_t rmTrasnferStaticsReq::bResetStatistics`

8.242.2.2 `uint32_t rmTrasnferStaticsReq::ulMask`

8.243 slot_t Struct Reference

Data Fields

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

8.243.1 Detailed Description

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

Parameters

<i>uPhyCardStatus</i>	<ul style="list-style-type: none"> • State of the card in the Pyhsical Slot Status. <ul style="list-style-type: none"> – 0x00 - Unknown. – 0x01 - Absent. – 0x02 - Present.
<i>uPhySlotStatus</i>	<ul style="list-style-type: none"> • State of the Physical Slot status. <ul style="list-style-type: none"> – 0x00 Inactive. – 0x01 Activate.
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> • Logical Slot associated with this physical slot. THis is valid if the physical slot is active. <ul style="list-style-type: none"> – 1 - Slot 1. – 2 - Slot 2. – 3 - Slot 3. – 4 - Slot 4. – 5 - Slot 5.
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> • Number of sets the sets of ICCID
<i>bICCID[MAX_ICCID_LENGTH]</i>	<ul style="list-style-type: none"> • Contains the ICCID of the card in the physical slot.

8.243.2 Field Documentation

8.243.2.1 uint8_t slot_t::bICCID[255]

8.243.2.2 uint8_t slot_t::bICCIDLength

8.243.2.3 uint8_t slot_t::bLogicalSlot

8.243.2.4 uint32_t slot_t::uPhyCardStatus

8.243.2.5 uint32_t slot_t::uPhySlotStatus

8.244 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]

8.244.1 Detailed Description

This structure contains information about the SLOTS present.

Parameters

<i>cardState</i>	<ul style="list-style-type: none"> • Indicates the state of the card for each slot. <ul style="list-style-type: none"> – 0 - Absent – 1 - Present – 2 - Error
<i>upinState</i>	<ul style="list-style-type: none"> • Indicates the state of UPIN. <ul style="list-style-type: none"> – 0 - Unknown – 1 - Enabled and not verified – 2 - Enabled and verified – 3 - Disabled – 4 - Blocked – 5 - Permanently blocked – 0xFF - Not Available
<i>upinRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to verify the UPIN. • If 0xFF, information not available.
<i>upukRetries</i>	<ul style="list-style-type: none"> • Indicates the number of retries remaining to unblock the UPIN. • If 0xFF, information not available.

<i>errorState</i>	<ul style="list-style-type: none"> Indicates the reason for the card error, and is valid only when the card state is Error <ul style="list-style-type: none"> 0 - Unknown 1 - Power down 2 - Poll error 3 - No ATR received 4 - Volt mismatch 5 - Parity error 6 - Unknown; possibly removed 7 - Card returned technical problems 0xFF - Not Available Other values are possible and reserved for future use. When an unknown value is received, it is to be handled as "Unknown".
<i>numApp</i>	<ul style="list-style-type: none"> Indicates the number of applications available on the card. The following block is repeated for each application. i.e. AppStatus. If zero(0) then no AppStatus information exists.
<i>AppStatus[MAX_NO_OF_APPLICATIONS]</i>	<ul style="list-style-type: none"> See appStats for more information.

8.244.2 Field Documentation

8.244.2.1 `appStats slotInf::AppStatus[10]`

8.244.2.2 `uint8_t slotInf::cardState`

8.244.2.3 `uint8_t slotInf::errorState`

8.244.2.4 `uint8_t slotInf::numApp`

8.244.2.5 `uint8_t slotInf::upinRetries`

8.244.2.6 `uint8_t slotInf::upinState`

8.244.2.7 `uint8_t slotInf::upukRetries`

8.245 slots_t Struct Reference

Data Fields

- [slot_t uimSlotStatus](#) [255]

8.245.1 Field Documentation

8.245.1.1 `slot_t slots_t::uimSlotStatus[255]`

8.246 sMSCAddress Struct Reference

Data Fields

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

8.246.1 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

8.246.2 Field Documentation

8.246.2.1 uint8_t sMSCAddress::data[256]

8.246.2.2 uint8_t sMSCAddress::length

8.247 sMSCAddressTlv Struct Reference

Data Fields

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

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>SMSCInfo</i>	<ul style="list-style-type: none">• SMSC Address• See sMSCAddressInfo for more information

8.247.2 Field Documentation

8.247.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

8.247.2.2 uint8_t sMSCAddressTlv::TlvPresent

8.248 sMSEtwsMessage Struct Reference

Data Fields

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

- `uint8_t data` [1254]

8.248.1 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

8.248.2 Field Documentation

8.248.2.1 `uint8_t sMSEtwsMessage::data`[1254]

8.248.2.2 `uint16_t sMSEtwsMessage::length`

8.248.2.3 `uint8_t sMSEtwsMessage::notificationType`

8.249 sMSEtwsMessageTlv Struct Reference

Data Fields

- `uint8_t TlvPresent`
- `sMSEtwsMessageInfo EtwsMessageInfo`

8.249.1 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

8.249.2 Field Documentation

8.249.2.1 `sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo`

8.249.2.2 `uint8_t sMSEtwsMessageTlv::TlvPresent`

8.250 sMSEtwsPlmn Struct Reference

Data Fields

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

8.250.1 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

8.250.2 Field Documentation

8.250.2.1 uint16_t sMSEtwsPlmn::mobileCountryCode

8.250.2.2 uint16_t sMSEtwsPlmn::mobileNetworkCode

8.251 sMSMessageMode Struct Reference

Data Fields

- uint8_t [messageMode](#)

8.251.1 Detailed Description

Parameters

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

8.251.2 Field Documentation

8.251.2.1 uint8_t sMSMessageMode::messageMode

8.252 sMSMTMessage Struct Reference

Data Fields

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

8.252.1 Detailed Description

Parameters

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

8.252.2 Field Documentation

8.252.2.1 `uint32_t sMSMTMessage::messageIndex`

8.252.2.2 `uint32_t sMSMTMessage::storageType`

8.253 sMSOnIMS Struct Reference

Data Fields

- `uint8_t` [smsOnIMS](#)

8.253.1 Detailed Description

Parameters

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

8.253.2 Field Documentation

8.253.2.1 `uint8_t sMSOnIMS::smsOnIMS`

8.254 sMSOnIMSTlv Struct Reference

Data Fields

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

8.254.1 Detailed Description

Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none">• Boolean indicating the presence of the TLV in the QMI response
<i>IMSInfo</i>	<ul style="list-style-type: none">• SMS on IMS• See sMSOnIMSInfo for more information

8.254.2 Field Documentation

8.254.2.1 `sMSOnIMSInfo sMSOnIMSTlv::IMSInfo`

8.254.2.2 `uint8_t sMSOnIMSTlv::TlvPresent`

8.255 sMSTransferRouteMTMessage Struct Reference

Data Fields

- `uint8_t` [ackIndicator](#)

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

8.255.1 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

8.255.2 Field Documentation

8.255.2.1 uint8_t sMSTransferRouteMTMessage::ackIndicator

8.255.2.2 uint8_t sMSTransferRouteMTMessage::data[256]

8.255.2.3 uint8_t sMSTransferRouteMTMessage::format

8.255.2.4 uint16_t sMSTransferRouteMTMessage::length

8.255.2.5 uint32_t sMSTransferRouteMTMessage::transactionID

8.256 tdscdmaSigInfoExt Struct Reference

Data Fields

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

8.256.1 Detailed Description

Parameters

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

8.256.2 Field Documentation

8.256.2.1 float tdsdmaSigInfoExt::ecio

8.256.2.2 float tdsdmaSigInfoExt::rscp

8.256.2.3 float tdsdmaSigInfoExt::rsi

8.256.2.4 float tdsdmaSigInfoExt::sinr

8.257 transferRouteMessageTlv Struct Reference

Data Fields

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

8.257.1 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

8.257.2 Field Documentation

8.257.2.1 uint8_t transferRouteMessageTlv::TlvPresent

8.257.2.2 [sMSTransferRouteMTMessageInfo](#) transferRouteMessageTlv::TransferRouteMTMessageInfo

8.258 transferStatInd Struct Reference

Data Fields

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

8.258.1 Detailed Description

Parameters

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

8.258.2 Field Documentation

8.258.2.1 uint32_t transferStatInd::StatsMask

8.258.2.2 uint8_t transferStatInd::StatsPeriod

8.259 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](#)

8.259.1 Detailed Description

This structure contains Application Status Information loaded on the card.

Parameters

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

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

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

8.259.2 Field Documentation

8.259.2.1 `uint8_t uim_appStatus::aidLength`

8.259.2.2 `uint8_t uim_appStatus::aidVal[255]`

8.259.2.3 `uint8_t uim_appStatus::appState`

8.259.2.4 `uint8_t uim_appStatus::appType`

8.259.2.5 `uint8_t uim_appStatus::persoFeature`

8.259.2.6 `uint8_t uim_appStatus::persoRetries`

8.259.2.7 `uint8_t uim_appStatus::persoState`

8.259.2.8 uint8_t uim_appStatus::persoUnblockRetries

8.259.2.9 uint8_t uim_appStatus::pin1Retries

8.259.2.10 uint8_t uim_appStatus::pin1State

8.259.2.11 uint8_t uim_appStatus::pin2Retries

8.259.2.12 uint8_t uim_appStatus::pin2State

8.259.2.13 uint8_t uim_appStatus::puk1Retries

8.259.2.14 uint8_t uim_appStatus::puk2Retries

8.259.2.15 uint8_t uim_appStatus::univPin

8.260 uim_cardResult Struct Reference

Data Fields

- uint8_t [sw1](#)
- uint8_t [sw2](#)

8.260.1 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.

8.260.2 Field Documentation

8.260.2.1 uint8_t uim_cardResult::sw1

8.260.2.2 uint8_t uim_cardResult::sw2

8.261 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]

8.261.1 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.

8.261.2 Field Documentation

8.261.2.1 `uint16_t uim_cardStatus::index1xPri`

8.261.2.2 `uint16_t uim_cardStatus::index1xSec`

8.261.2.3 `uint16_t uim_cardStatus::indexGwPri`

8.261.2.4 `uint16_t uim_cardStatus::indexGwSec`

8.261.2.5 `uint8_t uim_cardStatus::numSlot`

8.261.2.6 `uim_slotInfo uim_cardStatus::SlotInfo[5]`

8.262 uim_changeUIMPIN Struct Reference

Data Fields

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

8.262.1 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.

8.262.2 Field Documentation

8.262.2.1 uint8_t uim_changeUIMPIN::oldPINLen

8.262.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

8.262.2.3 uint8_t uim_changeUIMPIN::pinID

8.262.2.4 uint8_t uim_changeUIMPIN::pinLen

8.262.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

8.263 uim_encryptedPIN1 Struct Reference

Data Fields

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

8.263.1 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.

8.263.2 Field Documentation

8.263.2.1 uint8_t uim_encryptedPIN1::pin1Len

8.263.2.2 uint8_t uim_encryptedPIN1::pin1Val[255]

8.264 uim_fileInfo Struct Reference

Data Fields

- uint16_t [fileID](#)
- uint8_t [pathLen](#)
- uint16_t [path](#) [255]

8.264.1 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).

8.264.2 Field Documentation

8.264.2.1 uint16_t uim_fileInfo::fileID

8.264.2.2 uint16_t uim_fileInfo::path[255]

8.264.2.3 uint8_t uim_fileInfo::pathLen

8.265 uim_hotSwapStatus Struct Reference

Data Fields

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

8.265.1 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

8.265.2 Field Documentation

8.265.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

8.265.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

8.266 uim_readResult Struct Reference

Data Fields

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

8.266.1 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.

8.266.2 Field Documentation

8.266.2.1 `uint8_t uim_readResult::content[255]`

8.266.2.2 `uint16_t uim_readResult::contentLen`

8.267 `uim_readTransparentInfo` Struct Reference

Data Fields

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

8.267.1 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.

8.267.2 Field Documentation

8.267.2.1 `uint16_t uim_readTransparentInfo::length`

8.267.2.2 `uint16_t uim_readTransparentInfo::offset`

8.268 `uim_remainingRetries` Struct Reference

Data Fields

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

8.268.1 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>unlockLeft</i>	<ul style="list-style-type: none">• Number of remaining attempts to unlock 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.

8.268.2 Field Documentation

8.268.2.1 `uint8_t uim_remainingRetries::unlockLeft`

8.268.2.2 `uint8_t uim_remainingRetries::verifyLeft`

8.269 uim_sessionInformation Struct Reference

Data Fields

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

8.269.1 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.

8.269.2 Field Documentation

8.269.2.1 `uint8_t uim_sessionInformation::aid[255]`

8.269.2.2 `uint8_t uim_sessionInformation::aidLength`

8.269.2.3 `uint8_t uim_sessionInformation::sessionType`

8.270 uim_setPINProtection Struct Reference

Data Fields

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

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

8.270.2 Field Documentation

- 8.270.2.1 `uint8_t uim_setPINProtection::pinID`
- 8.270.2.2 `uint8_t uim_setPINProtection::pinLength`
- 8.270.2.3 `uint8_t uim_setPINProtection::pinOperation`
- 8.270.2.4 `uint8_t uim_setPINProtection::pinValue[255]`

8.271 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]`

8.271.1 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.

8.271.2 Field Documentation

8.271.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.271.2.2 `uint8_t uim_slotInfo::cardState`

8.271.2.3 `uint8_t uim_slotInfo::errorState`

8.271.2.4 `uint8_t uim_slotInfo::numApp`

8.271.2.5 `uint8_t uim_slotInfo::upinRetries`

8.271.2.6 `uint8_t uim_slotInfo::upinState`

8.271.2.7 `uint8_t uim_slotInfo::upukRetries`

8.272 uim_UIMSessionInformation Struct Reference

Data Fields

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

8.272.1 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.

8.272.2 Field Documentation

8.272.2.1 uint8_t uim_UIMSessionInformation::aid[255]

8.272.2.2 uint8_t uim_UIMSessionInformation::aidLength

8.272.2.3 uint8_t uim_UIMSessionInformation::sessionType

8.273 uim_unblockUIMPIN Struct Reference

Data Fields

- uint8_t [pinID](#)
- uint8_t [pukLen](#)
- uint8_t [pukVal](#) [255]
- uint8_t [newPINLen](#)
- uint8_t [newPINVal](#) [255]

8.273.1 Detailed Description

This structure contains the information about the unblock pin parameters.

Parameters

<i>pinID</i>	<ul style="list-style-type: none"> Indicates the PIN ID to be changed. <ul style="list-style-type: none"> 1 - PIN1 (also called PIN) 2 - PIN2 3 - Universal PIN
--------------	--

<i>pukLen</i>	<ul style="list-style-type: none"> Length of the following elements i.e. puk value.
<i>pukVal[UIM_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> PIN Unlock Key value. This value is a sequence of ASCII characters.
<i>newPINLen</i>	<ul style="list-style-type: none"> Length of the following elements i.e. new pin value.
<i>newPINVal[UIM_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> New PIN value. This value is a sequence of ASCII characters.

8.273.2 Field Documentation

8.273.2.1 `uint8_t uim_unblockUIMPIN::newPINLen`

8.273.2.2 `uint8_t uim_unblockUIMPIN::newPINVal[255]`

8.273.2.3 `uint8_t uim_unblockUIMPIN::pinID`

8.273.2.4 `uint8_t uim_unblockUIMPIN::pukLen`

8.273.2.5 `uint8_t uim_unblockUIMPIN::pukVal[255]`

8.274 uim_verifyUIMPIN Struct Reference

Data Fields

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

8.274.1 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.
--	---

8.274.2 Field Documentation

8.274.2.1 uint8_t uim_verifyUIMPIN::pinID

8.274.2.2 uint8_t uim_verifyUIMPIN::pinLen

8.274.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

8.275 unpack_dms_GetActivationState_t Struct Reference

Data Fields

- uint8_t [state](#)

8.275.1 Detailed Description

Parameters

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

8.275.2 Field Documentation

8.275.2.1 uint8_t unpack_dms_GetActivationState_t::state

8.276 unpack_dms_GetBandCapability_t Struct Reference

Data Fields

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

8.276.1 Field Documentation

8.276.1.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

8.276.1.2 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

8.277 unpack_dms_GetCrashAction_t Struct Reference

Data Fields

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

8.277.1 Field Documentation

8.277.1.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

8.277.1.2 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

8.278 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](#)

8.278.1 Field Documentation

8.278.1.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

8.278.1.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

8.278.1.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

8.278.1.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

8.278.1.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

8.278.1.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

8.278.1.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

8.278.1.8 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

8.278.1.9 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

8.278.1.10 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

8.279 unpack_dms_GetCustFeaturesV2_t Struct Reference

Data Fields

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

8.279.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

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

8.279.2 Field Documentation

8.279.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

8.279.2.2 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

8.280 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](#)

8.280.1 Field Documentation

8.280.1.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

8.280.1.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

8.280.1.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

8.280.1.4 uint8_t unpack_dms_GetDeviceCap_t::Radiolfaces[64]

8.280.1.5 uint32_t unpack_dms_GetDeviceCap_t::RadiolfacesSize

8.280.1.6 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

8.280.1.7 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

8.281 unpack_dms_GetDeviceCapabilities_t Struct Reference

Data Fields

- uint32_t [maxTxChannelRate](#)
- uint32_t [maxRxChannelRate](#)
- uint32_t [dataServiceCaCapability](#)
- uint32_t [simCapability](#)
- uint32_t [radiofacesSize](#)
- uint8_t [Radiofaces](#) [255]

8.281.1 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

8.281.2 Field Documentation

8.281.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

8.281.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

8.281.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

8.281.2.4 uint8_t unpack_dms_GetDeviceCapabilities_t::Radiofaces[255]

8.281.2.5 uint32_t unpack_dms_GetDeviceCapabilities_t::radiofacesSize

8.281.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

8.282 unpack_dms_GetDeviceHardwareRev_t Struct Reference

Data Fields

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

8.282.1 Field Documentation

8.282.1.1 char unpack_dms_GetDeviceHardwareRev_t::String[255]

8.282.1.2 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

8.282.1.3 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

8.283 unpack_dms_GetDeviceMfr_t Struct Reference

Data Fields

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

8.283.1 Field Documentation

8.283.1.1 char unpack_dms_GetDeviceMfr_t::String[255]

8.283.1.2 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

8.283.1.3 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

8.284 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](#)

8.284.1 Field Documentation

8.284.1.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

8.284.1.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

8.284.1.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

8.284.1.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

8.284.1.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

8.284.1.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

8.284.1.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

8.284.1.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

8.284.1.9 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

8.285 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](#)

8.285.1 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.

8.285.2 Field Documentation

8.285.2.1 char `unpack_dms_GetFirmwareInfo_t::appversion_str`[85]

8.285.2.2 char `unpack_dms_GetFirmwareInfo_t::bootversion_str`[85]

8.285.2.3 char `unpack_dms_GetFirmwareInfo_t::carrier_str`[20]

8.285.2.4 char `unpack_dms_GetFirmwareInfo_t::cur_carr_name`[17]

8.285.2.5 char `unpack_dms_GetFirmwareInfo_t::cur_carr_rev`[13]

8.285.2.6 char `unpack_dms_GetFirmwareInfo_t::modelid_str`[20]

8.285.2.7 char `unpack_dms_GetFirmwareInfo_t::packageid_str`[85]

8.285.2.8 char `unpack_dms_GetFirmwareInfo_t::priversion_str`[16]

8.285.2.9 char `unpack_dms_GetFirmwareInfo_t::sku_str`[15]

8.285.2.10 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

8.286 unpack_dms_GetFirmwareRevision_t Struct Reference

Data Fields

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

8.286.1 Field Documentation

8.286.1.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

8.286.1.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

8.286.1.3 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

8.286.1.4 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

8.287 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](#)

8.287.1 Detailed Description

Parameters

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

8.287.2 Field Documentation

8.287.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

8.287.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

8.287.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

8.287.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

8.287.2.5 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

8.287.2.6 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

8.287.2.7 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

8.288 unpack_dms_GetFSN_t Struct Reference

Data Fields

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

8.288.1 Field Documentation

8.288.1.1 char unpack_dms_GetFSN_t::String[255]

8.288.1.2 uint16_t unpack_dms_GetFSN_t::Tlvresult

8.289 unpack_dms_GetHardwareRevision_t Struct Reference

Data Fields

- char [hwVer](#) [255]

8.289.1 Detailed Description

Parameters

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

8.289.2 Field Documentation

8.289.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

8.290 unpack_dms_GetIMSI_t Struct Reference

Data Fields

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

8.290.1 Field Documentation

8.290.1.1 char unpack_dms_GetIMSI_t::imsi[255]

8.290.1.2 uint16_t unpack_dms_GetIMSI_t::Tlvresult

8.291 unpack_dms_GetManufacturer_t Struct Reference

Data Fields

- char [manufacturer](#) [255]

- uint16_t [Tlvresult](#)

8.291.1 Detailed Description

This structure is used to store device manufacturer information.

Parameters

<i>manufacturer[O-UT]</i>	<ul style="list-style-type: none">• NULL terminated string
<i>Tlvresult</i>	<ul style="list-style-type: none">• Unpack Result

8.291.2 Field Documentation

8.291.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

8.291.2.2 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

8.292 unpack_dms_GetModelID_t Struct Reference

Data Fields

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

8.292.1 Detailed Description

Parameters

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

8.292.2 Field Documentation

8.292.2.1 char unpack_dms_GetModelID_t::modelid[255]

8.292.2.2 uint16_t unpack_dms_GetModelID_t::Tlvresult

8.293 unpack_dms_GetNetworkTime_t Struct Reference

Data Fields

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

8.293.1 Detailed Description

Parameters

<i>source</i>	<ul style="list-style-type: none"> Source of timestamp <ul style="list-style-type: none"> 0 - 32 kHz device clock 1 - CDMA network 2 - cdma2000 1xEV-DO network
<i>timestamp</i>	<ul style="list-style-type: none"> Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980)
<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.

8.293.2 Field Documentation

8.293.2.1 `uint16_t unpack_dms_GetNetworkTime_t::source`

8.293.2.2 `uint64_t unpack_dms_GetNetworkTime_t::timestamp`

8.293.2.3 `uint16_t unpack_dms_GetNetworkTime_t::Tlvresult`

8.294 `unpack_dms_GetOfflineReason_t` Struct Reference

Data Fields

- `uint32_t * pReasonMask`
- `uint32_t * pbPlatform`
- `uint16_t Tlvresult`

8.294.1 Detailed Description

This structure is used to store reason why the operating mode of the device is currently offline.

Parameters

<i>pReasonMask[OUT]</i>	<ul style="list-style-type: none"> Optional parameter Bitmask of offline reasons <ul style="list-style-type: none"> 0x00000001 - Host image configuration issue 0x00000002 - PRI image configuration issue 0x00000004 - PRI version incompatible 0x00000008 - PRI copy issue All others - Reserved
-------------------------	--

<i>pbPlatform[OUT]</i>	<ul style="list-style-type: none"> Optional parameter Is the device offline due to a platform restriction? <ul style="list-style-type: none"> 0 - No 1 - Yes
<i>Tlvresult</i>	<ul style="list-style-type: none"> Unpack Result

8.294.2 Field Documentation

8.294.2.1 uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform

8.294.2.2 uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask

8.294.2.3 uint16_t unpack_dms_GetOfflineReason_t::Tlvresult

8.295 unpack_dms_GetPower_t Struct Reference

Data Fields

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

8.295.1 Detailed Description

Parameters

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

8.295.2 Field Documentation

8.295.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

8.295.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

8.295.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

8.295.2.4 uint16_t unpack_dms_GetPower_t::Tlvresult

8.296 unpack_dms_GetPRLVersion_t Struct Reference

Data Fields

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

- uint16_t [Tlvresult](#)

8.296.1 Field Documentation

8.296.1.1 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

8.296.1.2 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

8.296.1.3 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

8.297 unpack_dms_GetSerialNumbers_t Struct Reference

Data Fields

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

8.297.1 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

8.297.2 Field Documentation

8.297.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

8.297.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

8.297.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

8.297.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

8.298 unpack_dms_GetUSBComp_t Struct Reference

Data Fields

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

8.298.1 Field Documentation

8.298.1.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

8.298.1.2 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

8.298.1.3 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

8.298.1.4 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

8.299 unpack_dms_GetVoiceNumber_t Struct Reference

Data Fields

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

8.299.1 Field Documentation

8.299.1.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

8.299.1.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

8.299.1.3 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

8.299.1.4 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

8.299.1.5 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

8.300 unpack_dms_SetCrashAction_t Struct Reference

Data Fields

- uint8_t [notused](#)

8.300.1 Detailed Description

Modem response. Not used

8.300.2 Field Documentation

8.300.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

8.301 unpack_dms_SetCustFeature_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.301.1 Field Documentation

8.301.1.1 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

8.302 unpack_dms_SetCustFeaturesV2_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.302.1 Detailed Description

This structure contains customization settings set to modem unpack

Parameters

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

8.302.2 Field Documentation

8.302.2.1 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

8.303 unpack_dms_SetEventReport_ind_t Struct Reference

Data Fields

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

8.303.1 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

8.303.2 Field Documentation

8.303.2.1 dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv

8.303.2.2 dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv

8.303.2.3 uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult

8.304 unpack_dms_SetEventReport_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.304.1 Field Documentation

8.304.1.1 uint16_t unpack_dms_SetEventReport_t::Tlvresult

8.305 unpack_dms_SetFirmwarePreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.305.1 Field Documentation

8.305.1.1 uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult

8.306 unpack_dms_SetPower_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.306.1 Field Documentation

8.306.1.1 uint16_t unpack_dms_SetPower_t::Tlvresult

8.307 unpack_dms_SetUSBComp_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.307.1 Field Documentation

8.307.1.1 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

8.308 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

Data Fields

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

8.308.1 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

8.308.2 Field Documentation

8.308.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

8.308.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

8.308.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

8.309 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

Data Fields

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

8.309.1 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

8.309.2 Field Documentation

8.309.2.1 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

8.309.2.2 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

8.309.2.3 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

8.310 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.310.1 Detailed Description

This structure contains set registration state for different indication unpack

Parameters

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

8.310.2 Field Documentation

8.310.2.1 uint16_t unpack_dms_SLQSDmsSwilIndicationRegister_t::Tlvresult

8.311 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](#)

8.311.1 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
---------------------------------	--

8.311.2 Field Documentation

8.311.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

8.311.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

8.311.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

8.311.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

8.311.2.5 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

8.312 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.312.1 Detailed Description

This structure contains Clear Dying GASP unpack

Parameters

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

8.312.2 Field Documentation

8.312.2.1 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult

8.313 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

Data Fields

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

8.313.1 Detailed Description

This structure contains Get Dying GASP Config unpack

Parameters

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

8.313.2 Field Documentation

8.313.2.1 `packgetDyingGaspCfg* unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg`

8.313.2.2 `uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult`

8.314 `unpack_dms_SLQSSwiGetDyingGaspStatistics_t` Struct Reference

Data Fields

- `packgetDyingGaspStatistics* pGetDyingGaspStatistics`
- `uint16_t Tlvresult`

8.314.1 Detailed Description

This structure contains Get Dying GASP Statistics.

Parameters

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

8.314.2 Field Documentation

8.314.2.1 `packgetDyingGaspStatistics* unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics`

8.314.2.2 `uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

8.315 `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]`

8.315.1 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

8.315.2 Field Documentation

8.315.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

8.315.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

8.315.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

8.315.2.4 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

8.315.2.5 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

8.315.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

8.316 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](#)

8.316.1 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

8.316.2 Field Documentation

8.316.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

8.316.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

8.316.2.3 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

8.316.2.4 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

8.316.2.5 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

8.316.2.6 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult

8.317 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.317.1 Detailed Description

This structure contains set Dying GASP Config unpack

Parameters

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

8.317.2 Field Documentation

8.317.2.1 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

8.318 unpack_dms_UIMGetICCID_t Struct Reference

Data Fields

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

8.318.1 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.

8.318.2 Field Documentation

8.318.2.1 uint8_t unpack_dms_UIMGetICCID_t::String[255]

8.318.2.2 uint8_t unpack_dms_UIMGetICCID_t::stringSize

8.318.2.3 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

8.319 unpack_fms_GetImagesPreference_t Struct Reference

Data Fields

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

8.319.1 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

8.319.2 Field Documentation

8.319.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

8.319.2.2 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

8.319.2.3 uint16_t unpack_fms_GetImagesPreference_t::Tlvresult

8.320 unpack_fms_GetStoredImages_t Struct Reference

Data Fields

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

8.320.1 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

8.320.2 Field Documentation

8.320.2.1 [FMSImageList](#) unpack_fms_GetStoredImages_t::imageList

8.320.2.2 [uint32_t](#) unpack_fms_GetStoredImages_t::imagelistSize

8.320.2.3 [uint16_t](#) unpack_fms_GetStoredImages_t::Tlvresult

8.321 unpack_fms_SetImagesPreference_t Struct Reference

Data Fields

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

8.321.1 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

8.321.2 Field Documentation

8.321.2.1 [uint8_t](#) unpack_fms_SetImagesPreference_t::ImageTypes[255]

8.321.2.2 [uint32_t](#) unpack_fms_SetImagesPreference_t::ImageTypesSize

8.321.2.3 [uint16_t](#) unpack_fms_SetImagesPreference_t::Tlvresult

8.322 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](#)

8.322.1 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

8.322.2 Field Documentation

8.322.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

8.322.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

8.322.2.3 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

8.322.2.4 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

8.322.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

8.322.2.6 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

8.322.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

8.322.2.8 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

8.322.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

8.322.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

8.322.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

- 8.322.2.12 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor
- 8.322.2.13 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude
- 8.322.2.14 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude
- 8.322.2.15 uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation
- 8.322.2.16 loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution
- 8.322.2.17 loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage
- 8.322.2.18 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal
- 8.322.2.19 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc
- 8.322.2.20 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical
- 8.322.2.21 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc
- 8.322.2.22 loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix
- 8.322.2.23 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask
- 8.322.2.24 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc
- 8.322.2.25 uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc
- 8.322.2.26 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc
- 8.322.2.27 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence
- 8.322.2.28 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability
- 8.322.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc
- 8.322.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid
- 8.322.2.31 uint32_t unpack_loc_BestAvailPos_Ind_t::status
- 8.322.2.32 uint16_t unpack_loc_BestAvailPos_Ind_t::Tlvresult

8.323 unpack_loc_Delete_Assist_Data_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.323.1 Detailed Description

This structure contains LOC delete assist data 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_Delete_Assist_Data_t::Tlvresult

8.324 unpack_loc_EngineState_Ind_t Struct Reference

Data Fields

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

8.324.1 Detailed Description

This structure contains LOC Engine State field.

Parameters

<i>engineState</i>	<ul style="list-style-type: none"> Location engine state. Valid values <ul style="list-style-type: none"> 1 - Location engine is on 2 - Location engine is off
<i>Tlvresult</i>	<ul style="list-style-type: none"> unpack result

8.324.2 Field Documentation

8.324.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

8.324.2.2 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

8.325 unpack_loc_EventRegister_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.325.1 Detailed Description

This structure contains Event Register unpack

Parameters

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

8.325.2 Field Documentation

8.325.2.1 uint16_t unpack_loc_EventRegister_t::Tlvresult

8.326 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](#)

8.326.1 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

8.326.2 Field Documentation

8.326.2.1 `uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed`

8.326.2.2 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid`

8.326.2.3 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel`

8.326.2.4 `uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId`

8.326.2.5 `loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime`

- 8.326.2.6 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading`
- 8.326.2.7 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc`
- 8.326.2.8 `uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence`
- 8.326.2.9 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability`
- 8.326.2.10 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular`
- 8.326.2.11 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.326.2.12 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor`
- 8.326.2.13 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor`
- 8.326.2.14 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude`
- 8.326.2.15 `uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds`
- 8.326.2.16 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude`
- 8.326.2.17 `uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation`
- 8.326.2.18 `loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution`
- 8.326.2.19 `loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage`
- 8.326.2.20 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal`
- 8.326.2.21 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc`
- 8.326.2.22 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical`
- 8.326.2.23 `loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix`
- 8.326.2.24 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask`
- 8.326.2.25 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc`
- 8.326.2.26 `uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc`
- 8.326.2.27 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc`
- 8.326.2.28 `uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence`
- 8.326.2.29 `uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability`
- 8.326.2.30 `uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc`
- 8.326.2.31 `uint8_t unpack_loc_PositionRpt_Ind_t::sessionId`
- 8.326.2.32 `uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus`
- 8.326.2.33 `uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult`

8.327 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

Data Fields

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

8.327.1 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

8.327.2 Field Documentation

8.327.2.1 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

8.327.2.2 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

8.328 unpack_loc_SetExtPowerState_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.328.1 Detailed Description

This structure contains Set Ext Power State unpack

Parameters

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

8.328.2 Field Documentation

8.328.2.1 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

8.329 unpack_loc_SetOperationMode_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.329.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

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

8.329.2 Field Documentation

8.329.2.1 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

8.330 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.330.1 Detailed Description

This structure contains Set Operation Mode unpack

Parameters

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

8.330.2 Field Documentation

8.330.2.1 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

8.331 unpack_loc_Start_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.331.1 Detailed Description

This structure contains Start LOC unpack

Parameters

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

8.331.2 Field Documentation

8.331.2.1 uint16_t unpack_loc_Start_t::Tlvresult

8.332 unpack_loc_Stop_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.332.1 Detailed Description

This structure contains Stop LOC unpack

Parameters

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

8.332.2 Field Documentation

8.332.2.1 uint16_t unpack_loc_Stop_t::Tlvresult

8.333 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](#)

8.333.1 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

8.333.2 Field Documentation

8.333.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

8.333.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

8.333.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

8.333.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

8.333.2.5 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

8.333.2.6 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

8.333.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

8.333.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

8.333.2.9 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

8.333.2.10 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

8.333.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

8.334 unpack_nas_GetHomeNetwork_t Struct Reference

Data Fields

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

8.334.1 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
------------	--

8.334.2 Field Documentation

8.334.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

8.334.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

8.334.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

8.334.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

8.334.2.5 uint16_t unpack_nas_GetHomeNetwork_t::sid

8.335 unpack_nas_GetNetworkPreference_t Struct Reference

Data Fields

- uint32_t [ActiveTechPref](#)
- uint32_t [Duration](#)
- uint32_t [PersistentTechPref](#)
- 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 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

8.335.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

8.335.2.3 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

8.335.2.4 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

8.336 unpack_nas_GetRFInfo_t Struct Reference

Data Fields

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

8.336.1 Detailed Description

Parameters

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

8.336.2 Field Documentation

8.336.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

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

8.337 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]

8.337.1 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

8.337.2 Field Documentation

8.337.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

8.337.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

8.337.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

8.337.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

8.337.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

8.337.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

8.337.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

8.337.2.8 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

8.337.2.9 uint8_t unpack_nas_GetServingNetwork_t::Radiolfaces[255]

8.337.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadiolfacesSize

8.337.2.11 uint32_t unpack_nas_GetServingNetwork_t::RAN

8.337.2.12 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

8.337.2.13 uint32_t unpack_nas_GetServingNetwork_t::Roaming

8.338 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

Data Fields

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

8.338.1 Detailed Description

Parameters

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

8.338.2 Field Documentation

8.338.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

8.338.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

8.339 unpack_nas_GetSignalStrengths_t Struct Reference

Data Fields

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

8.339.1 Detailed Description

Parameters

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

8.339.2 Field Documentation

8.339.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

8.339.2.2 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

8.339.2.3 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

8.340 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](#)

8.340.1 Detailed Description

Parameters

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

8.340.2 Field Documentation

8.340.2.1 nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

8.340.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

8.340.2.3 nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance

8.340.2.4 uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize

8.340.2.5 nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATInstance

8.340.2.6 uint8_t* unpack_nas_PerformNetworkScan_t::pRATInstanceSize

8.340.2.7 uint32_t* unpack_nas_PerformNetworkScan_t::pScanResult

8.341 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

Data Fields

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

8.341.1 Detailed Description

Parameters

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

8.341.2 Field Documentation

8.341.2.1 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]

8.341.2.2 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize

8.342 unpack_nas_SetEventReportInd_t Struct Reference

Data Fields

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

8.342.1 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

8.342.2 Field Documentation

8.342.2.1 `nas_RFInfoTlv` `unpack_nas_SetEventReportInd_t::RFTlv`

8.342.2.2 `nas_RejectReasonTlv` `unpack_nas_SetEventReportInd_t::RRTlv`

8.342.2.3 `nas_SLQSSignalStrengthsTlv` `unpack_nas_SetEventReportInd_t::SLQSSSTlv`

8.342.2.4 `nas_SignalStrengthTlv` `unpack_nas_SetEventReportInd_t::SSTlv`

8.343 `unpack_nas_SetNasLTECphyCalndCallback_ind_t` Struct Reference

Data Fields

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

8.343.1 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.

8.343.2 Field Documentation

8.343.2.1 `nas_PhyCaAggPcellInfo` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggPcellInfo`

8.343.2.2 nas_PhyCaAggScellIDIBw unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIDIBw

8.343.2.3 nas_PhyCaAggScellIndex unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndex

8.343.2.4 nas_PhyCaAggScellIndType unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndType

8.343.2.5 nas_PhyCaAggScellInfo unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellInfo

8.344 unpack_nas_SetNetworkPreference_t Struct Reference

Data Fields

- uint16_t [Tlvresult](#)

8.344.1 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

8.344.2 Field Documentation

8.344.2.1 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

8.345 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

Data Fields

- uint8_t [roaming](#)

8.345.1 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
----------------	---

8.345.2 Field Documentation

8.345.2.1 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

8.346 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

Data Fields

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

8.346.1 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

8.346.2 Field Documentation

8.346.2.1 [NAServingSystemInfo](#) unpack_nas_SetServingSystemCallback_ind_t::SSInfo

8.346.2.2 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

8.347 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

Data Fields

- [NasGetLTECphyCAInfo](#) LTECphyCAInfo

- uint16_t [Tlvresult](#)

8.347.1 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

8.347.2 Field Documentation

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

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

8.348 unpack_nas_SLQSGetNetworkTime_t Struct Reference

Data Fields

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

8.348.1 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

8.348.2 Field Documentation

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

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

8.349 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]

8.349.1 Field Documentation

- 8.349.1.1 char [unpack_nas_SLQSGetPLMNName_t::longName](#)[255]
- 8.349.1.2 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameCI](#)
- 8.349.1.3 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameEn](#)
- 8.349.1.4 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameLen](#)
- 8.349.1.5 uint8_t [unpack_nas_SLQSGetPLMNName_t::longNameSB](#)
- 8.349.1.6 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortName](#)[255]
- 8.349.1.7 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameCI](#)
- 8.349.1.8 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameEn](#)
- 8.349.1.9 char [unpack_nas_SLQSGetPLMNName_t::shortNameLen](#)
- 8.349.1.10 uint8_t [unpack_nas_SLQSGetPLMNName_t::shortNameSB](#)
- 8.349.1.11 char [unpack_nas_SLQSGetPLMNName_t::spn](#)[255]
- 8.349.1.12 uint8_t [unpack_nas_SLQSGetPLMNName_t::spnEncoding](#)
- 8.349.1.13 uint8_t [unpack_nas_SLQSGetPLMNName_t::spnLength](#)

8.350 [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 [CDMASystemInfoExt](#)
- uint8_t [HdrPersonality](#)
- uint16_t [TrackAreaCode](#)
- nas_callBarStatus [CallBarStatus](#)

8.350.1 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>CDMASystem-InfoExt</i>	extra cdma system info
<i>HdrPersonality</i>	hdr personality
<i>TrackAreaCode</i>	track area code
<i>CallBarStatus</i>	call barring status

8.350.2 Field Documentation

- 8.350.2.1 `uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID`
- 8.350.2.2 `uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude`
- 8.350.2.3 `uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude`
- 8.350.2.4 `nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus`
- 8.350.2.5 `uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev`
- 8.350.2.6 `nas_CDMA SysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMA SystemInfoExt`
- 8.350.2.7 `uint32_t unpack_nas_SLQSGetServingSystem_t::CellID`
- 8.350.2.8 `uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo`
- 8.350.2.9 `nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN`
- 8.350.2.10 `nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities`
- 8.350.2.11 `uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd`
- 8.350.2.12 `nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo`
- 8.350.2.13 `uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd`
- 8.350.2.14 `nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone`
- 8.350.2.15 `uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment`
- 8.350.2.16 `uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone`
- 8.350.2.17 `uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality`
- 8.350.2.18 `uint16_t unpack_nas_SLQSGetServingSystem_t::Lac`
- 8.350.2.19 `uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID`
- 8.350.2.20 `uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd`
- 8.350.2.21 `uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal`
- 8.350.2.22 `nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList`
- 8.350.2.23 `nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem`
- 8.350.2.24 `uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID`
- 8.350.2.25 `uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode`

8.351 `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](#)

8.351.1 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

8.351.2 Field Documentation

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

8.351.2.2 [uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen](#)

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

8.351.2.4 [uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen](#)

8.351.2.5 [int32_t unpack_nas_SLQSGetSignalStrength_t::lo](#)

8.351.2.6 [int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp](#)

8.351.2.7 [int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr](#)

8.351.2.8 [nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo](#)

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

8.351.2.10 [uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen](#)

8.351.2.11 [uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask](#)

8.351.2.12 [uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr](#)

8.352 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](#)

8.352.1 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>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

8.352.2 Field Documentation

- 8.352.2.1 `nas_AddCDMASysInfo*` `unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo`
- 8.352.2.2 `nas_AddSysInfo*` `unpack_nas_SLQSGetSysInfo_t::pAddGSM SysInfo`
- 8.352.2.3 `uint16_t*` `unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo`
- 8.352.2.4 `uint16_t*` `unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo`
- 8.352.2.5 `nas_AddSysInfo*` `unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo`
- 8.352.2.6 `nas_SrvStatusInfo*` `unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo`
- 8.352.2.7 `nas_CDMASysInfo*` `unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo`
- 8.352.2.8 `nas_CallBarringSysInfo*` `unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo`
- 8.352.2.9 `uint8_t*` `unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo`
- 8.352.2.10 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSGetSysInfo_t::pGSM SrvStatusInfo`
- 8.352.2.11 `nas_GSM SysInfo*` `unpack_nas_SLQSGetSysInfo_t::pGSM SysInfo`
- 8.352.2.12 `nas_SrvStatusInfo*` `unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo`
- 8.352.2.13 `nas_HDRSysInfo*` `unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo`
- 8.352.2.14 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo`
- 8.352.2.15 `nas_LTESysInfo*` `unpack_nas_SLQSGetSysInfo_t::pLTESysInfo`
- 8.352.2.16 `uint8_t*` `unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo`
- 8.352.2.17 `nas_CallBarringSysInfo*` `unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo`
- 8.352.2.18 `uint8_t*` `unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo`
- 8.352.2.19 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo`
- 8.352.2.20 `nas_WCDMASysInfo*` `unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo`

8.353 `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`

8.353.1 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>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

8.353.2 Field Documentation

8.353.2.1 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

8.353.2.2 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

8.353.2.3 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

8.353.2.4 uint64_t* unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

8.353.2.5 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pModePref

8.353.2.6 uint8_t* unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

8.353.2.7 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

8.353.2.8 uint16_t* unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

8.353.2.9 uint32_t* unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

8.354 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](#)

8.354.1 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.

8.354.2 Field Documentation

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

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

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

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

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

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

8.354.2.7 uint32_t* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

8.354.2.8 nas_UMTSInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

8.354.2.9 nas_WCDMAInfoLTENeighborCell* unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

8.355 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

Data Fields

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

8.355.1 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

8.355.2 Field Documentation

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

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

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

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

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

8.356 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

Data Fields

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

8.356.1 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_1XEVD0 - cdma2000 HRPD (1xEV-DO) 0x04 - NAS_RADIO_IF_GSM - GSM 0x05 - NAS_RADIO_IF_UMTS - UMTS 0x08 - NAS_RADIO_IF_LTE - LTE 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA

8.356.2 Field Documentation

8.356.2.1 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

8.356.2.2 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

8.356.2.3 uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

8.356.2.4 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

8.357 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference

Data Fields

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

8.357.1 Detailed Description

Parameters

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

8.357.2 Field Documentation

8.357.2.1 **cdmaSSInfo*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo`8.357.2.2 **int8_t*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo`8.357.2.3 **hdrSSInfo*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo`8.357.2.4 **lteSSInfo*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo`8.357.2.5 **int8_t*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp`8.357.2.6 **tdscdmaSigInfoExt*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt`8.357.2.7 **cdmaSSInfo*** `unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo`8.358 `unpack_nas_SLQSNasSwiModemStatus_t` Struct Reference

Data Fields

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

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

8.358.2 Field Documentation

8.358.2.1 **nas_CommInfo** `unpack_nas_SLQSNasSwiModemStatus_t::commonInfo`8.358.2.2 **nas_LTEInfo*** `unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo`

8.359 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference

Data Fields

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

8.359.1 Detailed Description

Parameters

<i>Info</i>	<ul style="list-style-type: none">• Structure used to store all QMI Notification Info.<ul style="list-style-type: none">– See NASQmiCbkNasSwiOTAMessageInd for more details
<i>Tlvresult</i>	<ul style="list-style-type: none">• unpack result

8.359.2 Field Documentation

8.359.2.1 [NASQmiCbkNasSwiOTAMessageInd](#) unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

8.359.2.2 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult

8.360 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

Data Fields

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

8.360.1 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

8.360.2 Field Documentation

8.360.2.1 [NASQmiCbkNasSystemSelPrefInd](#) unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

8.360.2.2 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult

8.361 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

Data Fields

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

8.361.1 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

8.361.2 Field Documentation

8.361.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

8.361.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

8.361.2.3 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

8.361.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

8.362 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

Data Fields

- [nas_SccRxInfo](#) * [pSccRxInfo](#)

8.362.1 Detailed Description

Parameters

<i>pSccRxInfo</i>	Secondary carrier Rx signal level info
-------------------	--

8.362.2 Field Documentation

8.362.2.1 nas_SccRxInfo* unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

8.363 unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference

Data Fields

- nas_SrvStatusInfo * pCDMASrvStatusInfo
- nas_SrvStatusInfo * pHDRSrvStatusInfo
- nas_GSMStatusInfo * pGSMStatusInfo
- nas_GSMStatusInfo * pWCDMASrvStatusInfo
- nas_GSMStatusInfo * pLTESrvStatusInfo
- nas_CDMASysInfo * pCDMASysInfo
- nas_HDRSysInfo * pHDRSysInfo
- nas_GSMStatusInfo * pGSMStatusInfo
- nas_WCDMASysInfo * pWCDMASysInfo
- nas_LTESysInfo * pLTESysInfo
- nas_AddCDMASysInfo * pAddCDMASysInfo
- uint16_t * pAddHDRSysInfo
- nas_AddSysInfo * pAddGSMStatusInfo
- nas_AddSysInfo * pAddWCDMASysInfo
- uint16_t * pAddLTESysInfo
- nas_CallBarringSysInfo * pGSMCallBarringSysInfo
- nas_CallBarringSysInfo * pWCDMACallBarringSysInfo
- uint8_t * pLTEVoiceSupportSysInfo
- uint8_t * pGSMCipherDomainSysInfo
- uint8_t * pWCDMACipherDomainSysInfo
- uint8_t * pSysInfoNoChange

8.363.1 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>pGSMStatusInfo</i>	<ul style="list-style-type: none"> • See GSMStatusInfo for more information.
<i>pWCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> • See GSMStatusInfo for more information.
<i>pLTESrvStatusInfo</i>	<ul style="list-style-type: none"> • See GSMStatusInfo for more information.
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> • See CDMAStatusInfo for more information.
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> • See HDRStatusInfo for more information.

<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> • See GSMSysInfo for more information.
<i>pWCDMASys-Info</i>	<ul style="list-style-type: none"> • See WCDMASysInfo for more information.
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> • See LTESysInfo for more information.
<i>pAddCDMASys-Info</i>	<ul style="list-style-type: none"> • See AddCDMASysInfo for more information.
<i>pAddHDRSys-Info</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
<i>pAddGSMSys-Info</i>	<ul style="list-style-type: none"> • See AddSysInfo for more information.
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> • See AddSysInfo for more information.
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> • System table index referencing the beginning of the geo in which the current serving system is present. • When the system index is not known, 0xFFFF is used.
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See CallBarringSysInfo for more information.
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> • See CallBarringSysInfo for more information.
<i>pLTEVoice-SupportSysInfo</i>	<ul style="list-style-type: none"> • Indicates voice support status on LTE. <ul style="list-style-type: none"> – 0x00 - Voice is not supported – 0x01 - Voice is supported
<i>pGSMCipher-DomainSysInfo</i>	<ul style="list-style-type: none"> • Ciphering on the service domain. <ul style="list-style-type: none"> – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched
<i>pWCDMA-CipherDomain-SysInfo</i>	<ul style="list-style-type: none"> • Ciphering on the service domain. <ul style="list-style-type: none"> – 0x00 - No service – 0x01 - Circuit-switched only – 0x02 - Packet-switched only – 0x03 - Circuit-switched and packet-switched

<i>pSysInfoNo-Change</i>	<ul style="list-style-type: none"> • System Info No Change. • Flag used to notify clients that a request to select a network ended with no change in the PLMN. <ul style="list-style-type: none"> – 0x01 - No change in system information
--------------------------	--

8.363.2 Field Documentation

8.363.2.1 **nas_AddCDMASysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

8.363.2.2 **nas_AddSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

8.363.2.3 **uint16_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

8.363.2.4 **uint16_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

8.363.2.5 **nas_AddSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo

8.363.2.6 **nas_SrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo

8.363.2.7 **nas_CDMASysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo

8.363.2.8 **nas_CallBarringSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo

8.363.2.9 **uint8_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo

8.363.2.10 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo

8.363.2.11 **nas_GSMSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo

8.363.2.12 **nas_SrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pHDSrvStatusInfo

8.363.2.13 **nas_HDRSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pHDSysInfo

8.363.2.14 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo

8.363.2.15 **nas_LTESysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo

8.363.2.16 **uint8_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo

8.363.2.17 **uint8_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange

8.363.2.18 **nas_CallBarringSysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo

8.363.2.19 **uint8_t*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo

8.363.2.20 **nas_GSMSrvStatusInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo

8.363.2.21 **nas_WCDMASysInfo*** unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo

8.364 unpack_omaDmConfigTlv_t Struct Reference

Data Fields

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

8.364.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

Parameters

<i>state</i>	<ul style="list-style-type: none"> • 0x01 - OMA-DM Read Request • 0x02 - OMA-DM Change Request • 0x03 - OMA-DM Config Complete
<i>user_input_req</i>	- Bit mask of available user inputs <ul style="list-style-type: none"> • 0x00 - No user input required. Informational indication • 0x01 - Accept • 0x02 - Reject
<i>user_input_timeout</i>	<ul style="list-style-type: none"> • Timeout for user input in minutes. A value of 0 means no time-out
<i>alertmsglength</i>	<ul style="list-style-type: none"> • Length of Alert message string in bytes
<i>alertmsg</i>	<ul style="list-style-type: none"> • Alert message in UCS2 (Max 256 characters)

8.364.2 Field Documentation

8.364.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

8.364.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

8.364.2.3 uint8_t unpack_omaDmConfigTlv_t::state

8.364.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

8.364.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

8.365 unpack_omaDmFotaTlv_t Struct Reference

Data Fields

- uint8_t [state](#)
- uint8_t [userInputReq](#)
- uint16_t [userInputTimeout](#)
- uint32_t [fwddownloadsize](#)

- 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

8.365.1 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

8.365.2 Field Documentation

- 8.365.2.1 `uint8_t unpack_omaDmFotaTlv_t::description[256]`
- 8.365.2.2 `uint16_t unpack_omaDmFotaTlv_t::descriptionlength`
- 8.365.2.3 `uint32_t unpack_omaDmFotaTlv_t::fwdownloadsize`
- 8.365.2.4 `uint32_t unpack_omaDmFotaTlv_t::fwloadComplete`
- 8.365.2.5 `uint16_t unpack_omaDmFotaTlv_t::namelength`
- 8.365.2.6 `uint8_t unpack_omaDmFotaTlv_t::package_name[256]`
- 8.365.2.7 `uint8_t unpack_omaDmFotaTlv_t::sessionType`
- 8.365.2.8 `uint8_t unpack_omaDmFotaTlv_t::severity`
- 8.365.2.9 `uint8_t unpack_omaDmFotaTlv_t::state`
- 8.365.2.10 `uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus`
- 8.365.2.11 `uint8_t unpack_omaDmFotaTlv_t::userInputReq`
- 8.365.2.12 `uint16_t unpack_omaDmFotaTlv_t::userInputTimeout`
- 8.365.2.13 `uint8_t unpack_omaDmFotaTlv_t::version[256]`
- 8.365.2.14 `uint16_t unpack_omaDmFotaTlv_t::versionlength`

8.366 `unpack_omaDmNotificationsTlv_t` Struct Reference

Data Fields

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

8.366.1 Field Documentation

8.366.1.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

8.366.1.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

8.367 unpack_qmi_t Struct Reference

Data Fields

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

8.367.1 Detailed Description

qmi response context

Parameters

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

8.367.2 Field Documentation

8.367.2.1 uint16_t unpack_qmi_t::msgid

8.367.2.2 enum [msgtype](#) unpack_qmi_t::type

8.367.2.3 uint16_t unpack_qmi_t::xid

8.368 unpack_qos_dataRate_t Struct Reference

Data Fields

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

8.368.1 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)

8.368.2 Field Documentation

8.368.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

8.368.2.2 `uint32_t unpack_qos_dataRate_t::guaranteedRate`

8.369 `unpack_qos_IPv4Addr_t` Struct Reference

Data Fields

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

8.369.1 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

8.369.2 Field Documentation

8.369.2.1 `uint32_t unpack_qos_IPv4Addr_t::addr`

8.369.2.2 `uint32_t unpack_qos_IPv4Addr_t::subnetMask`

8.370 `unpack_qos_IPv6Addr_t` Struct Reference

Data Fields

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

8.370.1 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.

8.370.2 Field Documentation

8.370.2.1 `uint8_t unpack_qos_IPv6Addr_t::addr[16]`

8.370.2.2 `uint8_t unpack_qos_IPv6Addr_t::prefixLen`

8.371 unpack_qos_IPv6TrafCls_t Struct Reference

Data Fields

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

8.371.1 Detailed Description

This structure contains the IPv6 filter traffic class

Parameters

<i>val</i>	The traffic class value
<i>mask</i>	The packet matches the traffic class filter if: (IPv6_filter_traffic_class_val and IPv6_filter_traffic_class_mask) == (Traffic class value in the IP packet & IPv6_filter_traffic_class_mask) Example: <ul style="list-style-type: none"> • IPv6_filter_tc_val = 00101000 • IPv6_filter_tc_mask = 11111100 Filter will compare only the first 6 bits in IPv6_filter_traffic_class with the first 6 bits in the traffic class field of the IP packet; first 6 bits in the traffic class field of the IP packet must be 001010 to match filter; last 2 bits can be anything, since they are ignored by filtering

8.371.2 Field Documentation

8.371.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

8.371.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

8.372 unpack_qos_pktErrRate_t Struct Reference

Data Fields

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

8.372.1 Detailed Description

This structure contains the IP flow packet error rate

Parameters

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

8.372.2 Field Documentation

8.372.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

8.372.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

8.373 unpack_qos_Port_t Struct Reference

Data Fields

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

8.373.1 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

8.373.2 Field Documentation

8.373.2.1 uint16_t unpack_qos_Port_t::port

8.373.2.2 uint16_t unpack_qos_Port_t::range

8.374 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](#)

8.374.1 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.

8.374.2 Field Documentation

8.374.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`

8.374.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`

8.374.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`

8.374.2.4 `uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters`

8.374.2.5 `uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters`

8.374.2.6 `unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState`

8.374.2.7 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]`

8.374.2.8 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted`

8.374.2.9 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]`

8.374.2.10 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted`

8.375 unpack_qos_QosFlowInfoState_t Struct Reference

Data Fields

- uint32_t [id](#)
- uint8_t [isNewFlow](#)
- uint8_t [state](#)

8.375.1 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

8.375.2 Field Documentation

8.375.2.1 uint32_t [unpack_qos_QosFlowInfoState_t::id](#)

8.375.2.2 uint8_t [unpack_qos_QosFlowInfoState_t::isNewFlow](#)

8.375.2.3 uint8_t [unpack_qos_QosFlowInfoState_t::state](#)

8.376 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

Data Fields

- uint8_t [NWQoSStatus](#)

8.376.1 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
--------------------	--

8.376.2 Field Documentation

8.376.2.1 uint8_t unpack_qos_SLQSQoSGetNetworkStatus_t::NWQoSStatus

8.377 unpack_qos_SLQSQoSwiReadApnExtraParams_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](#)

8.377.1 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

8.377.2 Field Documentation

8.377.2.1 uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl

8.377.2.2 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext

8.377.2.3 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2

8.377.2.4 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul

8.377.2.5 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext

8.377.2.6 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2

8.377.2.7 uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

8.378 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]

8.378.1 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

8.378.2 Field Documentation

8.378.2.1 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId`

8.378.2.2 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow`

8.378.2.3 `unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]`

8.378.2.4 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes`

8.378.2.5 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt`

8.378.2.6 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes`

8.378.2.7 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp`

8.378.2.8 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt`

8.378.2.9 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp`

8.379 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

Data Fields

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

8.379.1 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

8.379.2 Field Documentation

8.379.2.1 `uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows`

8.379.2.2 `unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]`

8.380 `unpack_qos_SLQSSetQosNWStatusCallback_ind_t` Struct Reference

Data Fields

- `uint8_t` [status](#)

8.380.1 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

8.380.2 Field Documentation

8.380.2.1 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

8.381 `unpack_qos_SLQSSetQosPriEventCallback_ind_t` Struct Reference

Data Fields

- `uint16_t` [event](#)

8.381.1 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
--------------	--

8.381.2 Field Documentation

8.381.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

8.382 `unpack_qos_SLQSSetQosStatusCallback_ind_t` Struct Reference

Data Fields

- uint32_t [id](#)
- uint8_t [status](#)
- uint8_t [event](#)
- uint8_t [reason](#)

8.382.1 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

8.382.2 Field Documentation

8.382.2.1 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event`

8.382.2.2 `uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id`

8.382.2.3 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason`

8.382.2.4 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status`

8.383 `unpack_qos_swiQosFilter_t` Struct Reference

Data Fields

- `uint8_t index`
- `uint8_t version`
- `uint8_t is_IPv4SrcAddr_Available`
- `unpack_qos_IPv4Addr_t IPv4SrcAddr`
- `uint8_t is_IPv4DstAddr_Available`
- `unpack_qos_IPv4Addr_t IPv4DstAddr`
- `uint8_t is_NxtHdrProto_Available`
- `uint8_t NxtHdrProto`
- `uint8_t is_IPv4Tos_Available`
- `unpack_qos_Tos_t IPv4Tos`
- `uint8_t is_IPv6SrcAddr_Available`
- `unpack_qos_IPv6Addr_t IPv6SrcAddr`
- `uint8_t is_IPv6DstAddr_Available`
- `unpack_qos_IPv6Addr_t IPv6DstAddr`
- `uint8_t is_IPv6TrafCls_Available`
- `unpack_qos_IPv6TrafCls_t IPv6TrafCls`
- `uint8_t is_IPv6Label_Available`
- `uint32_t IPv6Label`
- `uint8_t is_TCPSrcPort_Available`
- `unpack_qos_Port_t TCPSrcPort`
- `uint8_t is_TCPDstPort_Available`
- `unpack_qos_Port_t TCPDstPort`
- `uint8_t is_UDPSrcPort_Available`
- `unpack_qos_Port_t UDPSrcPort`
- `uint8_t is_UDPDstPort_Available`
- `unpack_qos_Port_t UDPDstPort`
- `uint8_t is_EspSpi_Available`
- `uint32_t EspSpi`
- `uint8_t is_Precedence_Available`
- `uint16_t Precedence`
- `uint8_t is_Id_Available`
- `uint16_t Id`
- `uint8_t is_TranSrcPort_Available`
- `unpack_qos_Port_t TranSrcPort`
- `uint8_t is_TranDstPort_Available`
- `unpack_qos_Port_t TranDstPort`

8.383.1 Detailed Description

This structure contains the QoS Filter Request.

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

Parameters

<i>index</i>	Mandatory parameter IP filter index Integer that uniquely identifies each filter instance This TLV must be present in the request
<i>version</i>	Mandatory parameter IP filter version Identifies whether the filter is associated with IPv4 or IPv6; value specified also implies that only TLVs defined for that IP version, i.e., TLVs with IPv4 or IPv6 in the name, can be specified <ul style="list-style-type: none"> • 0x04 – IPv4 • 0x06 – Ipv6
<i>IPv4SrcAddr</i>	IPv4 filter soruce address See unpack_qos_IPv4Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>IPv4DstAddr</i>	IPv4 filter destination address See unpack_qos_IPv4Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>NxtHdrProto</i>	IP filter next header protocol This TLV must be present if any non-IP filter TLV(s) are provided If this field is specified, only IP packets belonging to specified higher layer protocol are considered when filtering The following protocols may be specified: <ul style="list-style-type: none"> • 0x01 = ICMP • 0x06 = TCP • 0x11 = UDP • 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP & UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified.
<i>IPv4Tos</i>	IPv4 filter type of service See unpack_qos_Tos_t for more information
<i>IPv6SrcAddr</i>	IPv6 filter soruce address See unpack_qos_IPv6Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>IPv6DstAddr</i>	IPv6 filter destination address See unpack_qos_IPv6Addr_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>IPv6TrafCls</i>	IPv6 filter traffic class See unpack_qos_IPv6TrafCls_t for more information
<i>IPv6Label</i>	IPv6 flow label Packet matches the IPv6 flow label filter if: (*pIPv6Label == flow label in the IPv6 header) <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>TCPSrcPort</i>	TCP filter source port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>TCPDstPort</i>	TCP filter destination port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>UDPSrcPort</i>	UDP filter source port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication
<i>UDPDstPort</i>	UDP filter destination port filter See unpack_qos_Port_t for more information <ul style="list-style-type: none"> • Implemented only for unsolicited indication

<i>EspSpi</i>	ESP filter security policy index Security policy index to uniquely identify each IP flow for filtering encrypted packets for encapsulating security payload <ul style="list-style-type: none"> Implemented only for unsolicited indication
<i>Precedence</i>	Filter Precedence Specifies the order in which filters are applied; lower numerical value has higher precedence Note: This TLV only applies to network-initiated QoS; QoS requests containing this TLV from control points will be ignored
<i>Id</i>	Filter ID Unique identifier for each filter; filter ID is assigned by the modem Note: This TLV only applies to network-initiated QoS; QoS requests containing this TLV from control points will be ignored
<i>TranSrcPort</i>	Transport protocol filter source port See unpack_qos_Port_t for more information <ul style="list-style-type: none"> Implemented only for unsolicited indication
<i>UDPDstPort</i>	Transport protocol filter destination port See unpack_qos_Port_t for more information <ul style="list-style-type: none"> Implemented only for unsolicited indication

8.383.2 Field Documentation

- 8.383.2.1 `uint32_t unpack_qos_swiQosFilter_t::EspSpi`
- 8.383.2.2 `uint16_t unpack_qos_swiQosFilter_t::Id`
- 8.383.2.3 `uint8_t unpack_qos_swiQosFilter_t::index`
- 8.383.2.4 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr`
- 8.383.2.5 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr`
- 8.383.2.6 `unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos`
- 8.383.2.7 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr`
- 8.383.2.8 `uint32_t unpack_qos_swiQosFilter_t::IPv6Label`
- 8.383.2.9 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr`
- 8.383.2.10 `unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls`
- 8.383.2.11 `uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available`
- 8.383.2.12 `uint8_t unpack_qos_swiQosFilter_t::is_Id_Available`
- 8.383.2.13 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available`
- 8.383.2.14 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available`
- 8.383.2.15 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available`
- 8.383.2.16 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available`
- 8.383.2.17 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available`
- 8.383.2.18 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available`

- 8.383.2.19 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available`
- 8.383.2.20 `uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available`
- 8.383.2.21 `uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available`
- 8.383.2.22 `uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available`
- 8.383.2.23 `uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available`
- 8.383.2.24 `uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available`
- 8.383.2.25 `uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available`
- 8.383.2.26 `uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available`
- 8.383.2.27 `uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available`
- 8.383.2.28 `uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto`
- 8.383.2.29 `uint16_t unpack_qos_swiQosFilter_t::Precedence`
- 8.383.2.30 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort`
- 8.383.2.31 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort`
- 8.383.2.32 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort`
- 8.383.2.33 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort`
- 8.383.2.34 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort`
- 8.383.2.35 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort`
- 8.383.2.36 `uint8_t unpack_qos_swiQosFilter_t::version`

8.384 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](#)

8.384.1 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 <code>flow_spec</code> 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

8.384.2 Field Documentation

8.384.2.1 `unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate`

8.384.2.2 `uint8_t unpack_qos_swiQosFlow_t::index`

8.384.2.3 `uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.384.2.4 `uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.384.2.5 `uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available`

8.384.2.6 `uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.384.2.7 `uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.384.2.8 `uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.384.2.9 `uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.384.2.10 `uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.384.2.11 `uint8_t unpack_qos_swiQosFlow_t::is-TokenBucket_Available`

8.384.2.12 `uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.384.2.13 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.384.2.14 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

- 8.384.2.15 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available
- 8.384.2.16 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available
- 8.384.2.17 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available
- 8.384.2.18 uint32_t unpack_qos_swiQosFlow_t::Jitter
- 8.384.2.19 uint32_t unpack_qos_swiQosFlow_t::Latency
- 8.384.2.20 uint8_t unpack_qos_swiQosFlow_t::LteQci
- 8.384.2.21 uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz
- 8.384.2.22 uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz
- 8.384.2.23 unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate
- 8.384.2.24 uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2
- 8.384.2.25 unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket
- 8.384.2.26 uint8_t unpack_qos_swiQosFlow_t::TrafficClass
- 8.384.2.27 uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri
- 8.384.2.28 uint8_t unpack_qos_swiQosFlow_t::val_3GPPIImCn
- 8.384.2.29 uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER
- 8.384.2.30 uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd
- 8.384.2.31 uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri

8.385 unpack_qos_tokenBucket_t Struct Reference

Data Fields

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

8.385.1 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

8.385.2 Field Documentation

8.385.2.1 `uint32_t unpack_qos_tokenBucket_t::bucketSz`

8.385.2.2 `uint32_t unpack_qos_tokenBucket_t::peakRate`

8.385.2.3 `uint32_t unpack_qos_tokenBucket_t::tokenRate`

8.386 `unpack_qos_Tos_t` Struct Reference

Data Fields

- `uint8_t val`
- `uint8_t mask`

8.386.1 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: (<code>IPv4_filter_tos_val</code> and <code>IPv4_filter_tos_mask</code>) == (TOS value in the IP packet & <code>IPv4_filter_tos_mask</code>) Example: <ul style="list-style-type: none"> • <code>IPv4_filter_tos_val</code> = 00101000 • <code>IPv4_filter_tos_mask</code> = 11111100 The filter will compare only the first 6 bits in the <code>IPv4_filter_type_of_service</code> 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.

8.386.2 Field Documentation

8.386.2.1 `uint8_t unpack_qos_Tos_t::mask`

8.386.2.2 `uint8_t unpack_qos_Tos_t::val`

8.387 `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`

8.387.1 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

8.387.2 Field Documentation

8.387.2.1 uint32_t unpack_QosFlowStat_t::bearerId

8.387.2.2 uint64_t unpack_QosFlowStat_t::tx_bytes

8.387.2.3 uint64_t unpack_QosFlowStat_t::tx_bytes_drp

8.387.2.4 uint32_t unpack_QosFlowStat_t::tx_pkt

8.387.2.5 uint32_t unpack_QosFlowStat_t::tx_pkt_drp

8.388 unpack_sms_SendSMS_t Struct Reference

Data Fields

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

8.388.1 Detailed Description

Parameters

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

8.388.2 Field Documentation

8.388.2.1 uint32_t unpack_sms_SendSMS_t::messageFailureCode

8.388.2.2 uint16_t unpack_sms_SendSMS_t::messageID

8.389 unpack_sms_SetNewSMSCallback_ind_t Struct Reference

Data Fields

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

8.389.1 Detailed Description

Parameters

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

8.389.2 Field Documentation

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

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

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

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

8.389.2.5 struct newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv

8.389.2.6 struct sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSTlv

8.389.2.7 struct transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv

8.390 unpack_sms_SetNewSMSCallback_t Struct Reference

8.391 unpack_sms_SLQSDDeleteSMS_t Struct Reference

8.392 unpack_sms_SLQSGetSMS_t Struct Reference

Data Fields

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

8.392.1 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
----------------	--

8.392.2 Field Documentation

8.392.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

8.392.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

8.392.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

8.392.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

8.393 unpack_sms_SLQSGetSMSList_t Struct Reference

Data Fields

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

8.393.1 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

8.393.2 Field Documentation

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

8.393.2.2 uint32_t [unpack_sms_SLQSGetSMSList_t::messageListSize](#)

8.394 unpack_sms_SLQSModifySMSStatus_t Struct Reference

8.395 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

Data Fields

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

8.395.1 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)
--------------------	--

8.395.2 Field Documentation

8.395.2.1 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode

8.395.2.2 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType

8.396 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](#)

8.396.1 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

8.396.2 Field Documentation

8.396.2.1 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate`

8.396.2.2 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported`

8.396.2.3 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type`

8.396.2.4 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported`

8.396.2.5 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function`

8.396.2.6 `int unpack_swiloc_SwiLocGetAutoStart_t::function_reported`

8.396.2.7 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist`

8.396.2.8 `int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported`

8.396.2.9 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time`

8.396.2.10 `int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported`

8.397 `unpack_swima_SLQSOMADMAAlertCallback_ind_t` Struct Reference

Data Fields

- `uint32_t eventType`
- `unpack_omaDmFotaTlv_t SessionInfoFota`

- [unpack_omaDmConfigTlv_t SessionInfoConfig](#)
- [unpack_omaDmNotificationsTlv_t SessionInfoNotification](#)

8.397.1 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

8.397.2 Field Documentation

8.397.2.1 `uint32_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::eventType`

8.397.2.2 `unpack_omaDmConfigTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoConfig`

8.397.2.3 `unpack_omaDmFotaTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoFota`

8.397.2.4 `unpack_omaDmNotificationsTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoNotification`

8.398 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](#)

8.398.1 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

8.398.2 Field Documentation

8.398.2.1 `uint8_t unpack_swioama_SLQSOMADMGetSessionInfo_t::Date[255]`

8.398.2.2 `uint16_t unpack_swioama_SLQSOMADMGetSessionInfo_t::DateLength`

8.398.2.3 `uint16_t unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgDescLength`

8.398.2.4 `uint8_t unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgDescription[255]`

8.398.2.5 `uint8_t unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgName[255]`

8.398.2.6 `uint16_t unpack_swioama_SLQSOMADMGetSessionInfo_t::PkgNameLength`

8.398.2.7 `uint16_t unpack_swioama_SLQSOMADMGetSessionInfo_t::RetryCount`

8.398.2.8 `uint8_t unpack_swioama_SLQSOMADMGetSessionInfo_t::SessionState`

8.398.2.9 `uint8_t unpack_swioama_SLQSOMADMGetSessionInfo_t::SessionType`

- 8.398.2.10 `uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Severity`
- 8.398.2.11 `uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Source[255]`
- 8.398.2.12 `uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::SourceLength`
- 8.398.2.13 `uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Status`
- 8.398.2.14 `uint8_t unpack_swima_SLQSOMADMGetSessionInfo_t::Time[255]`
- 8.398.2.15 `uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::TimeLength`
- 8.398.2.16 `uint16_t unpack_swima_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus`

8.399 `unpack_swima_SLQSOMADMGetSettings_t` Struct Reference

Data Fields

- `uint32_t OMADMEEnabled`
- `uint8_t FOTAdownload`
- `uint8_t FOTAUpdate`
- `uint8_t Autosdm`
- `uint8_t FwAutoCheck`

8.399.1 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 <code>SLQSOMADMGetSettings2()</code> 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 <code>SLQSOMADMGetSettings2()</code> returns a default value 0xFF in case this parameter is not returned by the modem.

<i>FOTAUpdate[O-UT]</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[O-UT]</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.

8.399.2 Field Documentation

8.399.2.1 uint8_t unpack_swisma_SLQSOMADMGetSettings_t::Autosdm

8.399.2.2 uint8_t unpack_swisma_SLQSOMADMGetSettings_t::FOTAdownload

8.399.2.3 uint8_t unpack_swisma_SLQSOMADMGetSettings_t::FOTAUpdate

8.399.2.4 uint8_t unpack_swisma_SLQSOMADMGetSettings_t::FwAutoCheck

8.399.2.5 uint32_t unpack_swisma_SLQSOMADMGetSettings_t::OMADMEabled

8.400 unpack_swisma_SLQSOMADMStartSession_t Struct Reference

Data Fields

- uint32_t [FwAvailability](#)

8.400.1 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
-----------------------------	---

8.400.2 Field Documentation

8.400.2.1 uint32_t unpack_swima_SLQSOMADMStartSession_t::FwAvailability

8.401 unpack_uim_ChangePin_t Struct Reference

Data Fields

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

8.401.1 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.

8.401.2 Field Documentation

8.401.2.1 uim_encryptedPIN1* unpack_uim_ChangePin_t::pEncryptedPIN1

8.401.2.2 uint32_t* unpack_uim_ChangePin_t::pIndicationToken

8.401.2.3 uim_remainingRetries* unpack_uim_ChangePin_t::pRemainingRetries

8.401.2.4 uint16_t unpack_uim_ChangePin_t::Tlvresult

8.402 unpack_uim_GetCardStatus_t Struct Reference

Data Fields

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

8.402.1 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.

8.402.2 Field Documentation

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

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

8.402.2.3 uint16_t [unpack_uim_GetCardStatus_t::Tlvresult](#)

8.403 unpack_uim_ReadTransparent_t Struct Reference

Data Fields

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

8.403.1 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 <code>cardResult</code> for more information.
<i>pReadResult</i>	<ul style="list-style-type: none"> • See <code>readResult</code> 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 <code>read_result</code> is encrypted.

Note

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

8.403.2 Field Documentation

8.403.2.1 `uim_cardResult*` `unpack_uim_ReadTransparent_t::pCardResult`

8.403.2.2 `uint8_t*` `unpack_uim_ReadTransparent_t::pEncryptedData`

8.403.2.3 `uint32_t*` `unpack_uim_ReadTransparent_t::pIndicationToken`

8.403.2.4 `uim_readResult*` `unpack_uim_ReadTransparent_t::pReadResult`

8.403.2.5 `uint16_t` `unpack_uim_ReadTransparent_t::Tlvresult`

8.404 `unpack_uim_SetPinProtection_t` Struct Reference

Data Fields

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

8.404.1 Detailed Description

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

Parameters

<i>pRemaining-Retries(optional)</i>	<ul style="list-style-type: none"> • 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>pIndication-Token(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.

8.404.2 Field Documentation

8.404.2.1 uim_encryptedPIN1* unpack_uim_SetPinProtection_t::pEncryptedPIN1

8.404.2.2 uint32_t* unpack_uim_SetPinProtection_t::pIndicationToken

8.404.2.3 uim_remainingRetries* unpack_uim_SetPinProtection_t::pRemainingRetries

8.404.2.4 uint16_t unpack_uim_SetPinProtection_t::Tlvresult

8.405 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference**Data Fields**

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

8.405.1 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)

8.405.2 Field Documentation

8.405.2.1 uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots

8.405.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

8.406 unpack_uim_SLQSUIMEventRegister_t Struct Reference**Data Fields**

- uint32_t [eventMask](#)

8.406.1 Detailed Description

Parameters

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

8.406.2 Field Documentation

8.406.2.1 `uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask`

8.407 `unpack_uim_SLQSUIMGetSlotsStatus_t` Struct Reference

Data Fields

- `uint8_t * pNumberOfPhySlot`
- `slots_t * pUimSlotsStatus`

8.407.1 Detailed Description

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

Parameters

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

8.407.2 Field Documentation

8.407.2.1 `uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot`

8.407.2.2 `slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus`

8.408 `unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t` Struct Reference

Data Fields

- `uim_cardStatus * pCardStatus`

8.408.1 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.
--------------------	--

8.408.2 Field Documentation

8.408.2.1 uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus

8.409 unpack_uim_UnblockPin_t Struct Reference

Data Fields

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

8.409.1 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

8.409.2 Field Documentation

8.409.2.1 uim_encryptedPIN1* unpack_uim_UnblockPin_t::pEncryptedPIN1

8.409.2.2 uint32_t* unpack_uim_UnblockPin_t::pIndicationToken

8.409.2.3 uim_remainingRetries* unpack_uim_UnblockPin_t::pRemainingRetries

8.409.2.4 uint16_t unpack_uim_UnblockPin_t::Tlvresult

8.410 unpack_uim_VerifyPin_t Struct Reference

Data Fields

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

8.410.1 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.

8.410.2 Field Documentation

8.410.2.1 `uim_encryptedPIN1*` `unpack_uim_VerifyPin_t::pEncryptedPIN1`

8.410.2.2 `uint32_t*` `unpack_uim_VerifyPin_t::pIndicationToken`

8.410.2.3 `uim_remainingRetries*` `unpack_uim_VerifyPin_t::pRemainingRetries`

8.410.2.4 `uint16_t` `unpack_uim_VerifyPin_t::Tlvresult`

8.411 `unpack_wds_GetByteTotals_t` Struct Reference

Data Fields

- `uint64_t *` `pTXTotalBytes`
- `uint64_t *` `pRXTotalBytes`

8.411.1 Detailed Description

Parameters

<i>pTXTotalBytes</i>	<ul style="list-style-type: none"> Bytes transmitted without error
<i>pRXTotalBytes</i>	<ul style="list-style-type: none"> Bytes received without error

8.411.2 Field Documentation

8.411.2.1 `uint64_t*` `unpack_wds_GetByteTotals_t::pRXTotalBytes`

8.411.2.2 `uint64_t*` `unpack_wds_GetByteTotals_t::pTXTotalBytes`

8.412 unpack_wds_GetConnectionRate_t Struct Reference

Data Fields

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

8.412.1 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

8.412.2 Field Documentation

8.412.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

8.412.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

8.412.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

8.412.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

8.413 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]

8.413.1 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

8.413.2 Field Documentation

8.413.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

8.413.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

8.413.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

8.413.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr

8.413.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6

8.413.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]

8.413.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize

8.413.2.8 uint32_t unpack_wds_GetDefaultProfile_t::pdptype

8.413.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pridns

8.413.2.10 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6

8.413.2.11 uint32_t unpack_wds_GetDefaultProfile_t::secdns

8.413.2.12 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6

8.413.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

8.413.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

8.414 unpack_wds_GetDefaultProfileNum_t Struct Reference

Data Fields

- uint8_t [index](#)

8.414.1 Detailed Description

Parameters

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

8.414.2 Field Documentation

8.414.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

8.415 unpack_wds_GetDormancyState_t Struct Reference

Data Fields

- uint32_t [dormancyState](#)

8.415.1 Detailed Description

Parameters

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

8.415.2 Field Documentation

8.415.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

8.416 unpack_wds_GetLastMobileIPError_t Struct Reference

Data Fields

- uint32_t [error](#)

8.416.1 Detailed Description

Parameters

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

8.416.2 Field Documentation

8.416.2.1 uint32_t unpack_wds_GetLastMobileIPError_t::error

8.417 unpack_wds_GetMobileIP_t Struct Reference

Data Fields

- uint32_t [mipMode](#)

8.417.1 Detailed Description

Parameters

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

8.417.2 Field Documentation

8.417.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

8.418 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](#)

8.418.1 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

8.418.2 Field Documentation

8.418.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

8.418.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAASState

8.418.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

8.418.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

8.418.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

8.418.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HASState

8.418.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]

8.418.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize

8.418.2.9 uint32_t unpack_wds_GetMobilePPProfile_t::primaryHA

8.418.2.10 uint8_t unpack_wds_GetMobilePPProfile_t::revTunneling

8.418.2.11 uint32_t unpack_wds_GetMobilePPProfile_t::secondaryHA

8.419 unpack_wds_GetPacketStatistics_t Struct Reference

Data Fields

- uint32_t * [pTXPacketSuccesses](#)
- uint32_t * [pRXPacketSuccesses](#)
- uint32_t * [pTXPacketErrors](#)
- uint32_t * [pRXPacketErrors](#)
- uint32_t * [pTXPacketOverflows](#)
- uint32_t * [pRXPacketOverflows](#)
- uint64_t * [pTXOkBytesCount](#)
- uint64_t * [pRXOkBytesCount](#)
- uint64_t * [pTXOKBytesLastCall](#)
- uint64_t * [pRXOKBytesLastCall](#)
- uint32_t * [pTXDroppedCount](#)
- uint32_t * [pRXDroppedCount](#)

8.419.1 Detailed Description

Parameters

<i>pTXPacket-Successes</i>	<ul style="list-style-type: none"> • No of transmitted Packets without error.
<i>pRXPacket-Successes</i>	<ul style="list-style-type: none"> • No of received Packets without error.
<i>pTXPacketErrors</i>	<ul style="list-style-type: none"> • Number of outgoing packets with framing errors.
<i>pRXPacket-Errors</i>	<ul style="list-style-type: none"> • Number of incoming packets with framing errors.
<i>pTXPacket-Overflows</i>	<ul style="list-style-type: none"> • Number of packets dropped because Tx buffer overflowed (out of memory).
<i>pRXPacket-Overflows</i>	<ul style="list-style-type: none"> • Number of packets dropped because Rx buffer overflowed (out of memory).
<i>pTXOkBytes-Count</i>	<ul style="list-style-type: none"> • No of bytes transmitted without error.
<i>pRXOkBytes-Count</i>	<ul style="list-style-type: none"> • No of bytes received without error.
<i>pTXOKBytes-LastCall</i>	<ul style="list-style-type: none"> • No of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support

<i>pRXOKBytes-LastCall</i>	<ul style="list-style-type: none"> Number of bytes received without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support
<i>pTXDropped-Count</i>	<ul style="list-style-type: none"> Number of outgoing packets dropped.
<i>pRXDropped-Count</i>	<ul style="list-style-type: none"> Number of incoming packets dropped.

8.419.2 Field Documentation

8.419.2.1 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount`

8.419.2.2 `uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount`

8.419.2.3 `uint64_t* unpack_wds_GetPacketStatistics_t::pRXOKBytesLastCall`

8.419.2.4 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors`

8.419.2.5 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows`

8.419.2.6 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses`

8.419.2.7 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount`

8.419.2.8 `uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount`

8.419.2.9 `uint64_t* unpack_wds_GetPacketStatistics_t::pTXOKBytesLastCall`

8.419.2.10 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors`

8.419.2.11 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows`

8.419.2.12 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses`

8.420 `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](#)

8.420.1 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

8.420.2 Field Documentation

- 8.420.2.1 `uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount`
- 8.420.2.2 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount`
- 8.420.2.3 `uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall`
- 8.420.2.4 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors`
- 8.420.2.5 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows`
- 8.420.2.6 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses`
- 8.420.2.7 `uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount`
- 8.420.2.8 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount`
- 8.420.2.9 `uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall`
- 8.420.2.10 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors`
- 8.420.2.11 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows`
- 8.420.2.12 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses`

8.421 unpack_wds_GetSessionDuration_t Struct Reference

Data Fields

- `uint64_t callDuration`

8.421.1 Detailed Description

Parameters

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

8.421.2 Field Documentation

8.421.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

8.422 unpack_wds_GetSessionState_t Struct Reference

Data Fields

- uint32_t [connectionStatus](#)

8.422.1 Detailed Description

Parameters

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

8.422.2 Field Documentation

8.422.2.1 uint32_t unpack_wds_GetSessionState_t::connectionStatus

8.423 unpack_wds_RMSetTransferStatistics_t Struct Reference

8.424 unpack_wds_SetMobileIPProfile_t Struct Reference

8.425 unpack_wds_SLQSCreateProfile_t Struct Reference

Data Fields

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

8.425.1 Detailed Description

Parameters

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

8.425.2 Field Documentation

8.425.2.1 [PackCreateProfileOut](#)* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut

8.425.2.2 uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID

8.425.2.3 uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult

8.426 unpack_wds_SLQSDeleteProfile_t Struct Reference

Data Fields

- uint16_t [extendedErrorCode](#)

8.426.1 Detailed Description

Parameters

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

8.426.2 Field Documentation

8.426.2.1 uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode

8.427 unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference

Data Fields

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

8.427.1 Detailed Description

Parameters

	<i>pLTEAttach-Profile</i>	<ul style="list-style-type: none"> Optional parameter LTE Attach Profile <ul style="list-style-type: none"> points to a single WORD Value indicating the attached LTE Profile Optional parameter with possible values 1-16 (EM/MC73xx or earlier) This setting is deprecated on MC/EM74xx
	<i>profileList</i>	Profile List <ul style="list-style-type: none"> an array of 4 profile configurations Each element points to a single WORD value indicating profile Optional parameter with possible values <ul style="list-style-type: none"> 1 - 16 (MC/EM73xx and before) 1 - 24 (MC/EM74xx and onwards) function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present

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

8.427.2 Field Documentation

8.427.2.1 `uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease`

8.427.2.2 `uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled`

8.427.2.3 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile`

8.427.2.4 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]`

8.427.2.5 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen`

8.427.2.6 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]`

8.428 `unpack_wds_SLQSGetCurrDataSystemStat_t` Struct Reference

Data Fields

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

8.428.1 Detailed Description

Parameters

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

8.428.2 Field Documentation

8.428.2.1 `currNetworkInfo` `unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]`8.428.2.2 `uint8_t` `unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen`8.428.2.3 `uint8_t` `unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork`

8.429 unpack_wds_SLQSGetCurrentChannelRate_t Struct Reference

Data Fields

- `uint32_t` [current_channel_tx_rate](#)
- `uint32_t` [current_channel_rx_rate](#)
- `uint32_t` [max_channel_tx_rate](#)
- `uint32_t` [max_channel_rx_rate](#)

8.429.1 Detailed Description

Parameters

<i>current_channel_tx_rate</i>	<ul style="list-style-type: none"> • Current Channel Tx Rate. • Instantaneous channel Tx rate in bits per second. • In 9x15, this is the total current channel rate for all PDNs combined. • In 9x30 and later, this is the channel rate for a specific PDN.
<i>current_channel_rx_rate</i>	<ul style="list-style-type: none"> • Current Channel Rx Rate. • Instantaneous channel Rx rate in bits per second. • In 9x15, this is the total current channel rate for all PDNs combined. • In 9x30 and later, this is the channel rate for a specific PDN
<i>max_channel_tx_rate</i>	<ul style="list-style-type: none"> • Max Channel Tx Rate. • Maximum total Tx rate that modem is able to support in current serving system in bits per second. • In 9x15, this is a default hard coded value for the current serving system.
<i>max_channel_rx_rate</i>	<ul style="list-style-type: none"> • Max Channel Rx Rate. • Maximum total Rx rate that modem is able to support in current serving system in bits per second. • In 9x15, this is a default hard coded value for the current serving system.

8.429.2 Field Documentation

8.429.2.1 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::current_channel_rx_rate`

8.429.2.2 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::current_channel_tx_rate`

8.429.2.3 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::max_channel_rx_rate`

8.429.2.4 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::max_channel_tx_rate`

8.430 `unpack_wds_SLQSGGetDataBearerTechnology_t` Struct Reference

Data Fields

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

8.430.1 Detailed Description

Parameters

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

8.430.2 Field Documentation

8.430.2.1 `qmiWSDDataBearerTechnology unpack_wds_SLQSGGetDataBearerTechnology_t::curDataBearerTechnology`

8.430.2.2 `uint8_t unpack_wds_SLQSGGetDataBearerTechnology_t::dataBearerMask`

8.430.2.3 `qmiWSDDataBearerTechnology unpack_wds_SLQSGGetDataBearerTechnology_t::lastCallDataBearer-Technology`

8.431 `unpack_wds_SLQSGGetDUNCallInfo_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`

8.431.1 Detailed Description

Parameters

<i>connection-Status</i>	Connection Status
<i>callEndReason</i>	Last Modem Call End Reason
<i>txOKBytesCount</i>	Tx Bytes OK
<i>rxOKBytesCount</i>	Rx Bytes OK
<i>dormancyStatus</i>	Dormancy Status
<i>dataBearerTech</i>	data bearer technology
<i>channelRate</i>	data Channel Rate
<i>lastCallTXOK-BytesCnt</i>	Last Call Tx Bytes OK
<i>lastCallRXOK-BytesCnt</i>	Last Call Rx Bytes OK
<i>mdmCall-DurationActive</i>	Call active duration
<i>lastCallData-BearerTech</i>	Last Call Data Bearer Technology

8.431.2 Field Documentation

- 8.431.2.1 `uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason`
- 8.431.2.2 `dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate`
- 8.431.2.3 `connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus`
- 8.431.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`
- 8.431.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`
- 8.431.2.6 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech`
- 8.431.2.7 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt`
- 8.431.2.8 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt`
- 8.431.2.9 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive`
- 8.431.2.10 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount`
- 8.431.2.11 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount`

8.432 unpack_wds_SLQSGetProfileSettings_t Struct Reference

Data Fields

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

8.432.1 Field Documentation

8.432.1.1 **UnPackGetProfileSettingOut*** `unpack_wds_SLQSGetProfileSettings_t::pProfileSettings`

8.432.1.2 `uint8_t` `unpack_wds_SLQSGetProfileSettings_t::ProfileType`

8.432.1.3 `uint16_t` `unpack_wds_SLQSGetProfileSettings_t::Tlvresult`

8.433 `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](#)

8.433.1 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>PCSCFAddrPCO</i>	
<i>PrimaryDNSV6</i>	Primary DNS IPV6
<i>SecondaryDNSV6</i>	Secondary DNS IPV6
<i>UMTSGrantedQoS</i>	UMTS Granted Qos
<i>SecondaryDNSV4</i>	
<i>Mtu</i>	Maximum Transfer Unit
<i>DomainList</i>	
<i>IPFamilyPreference</i>	

8.433.2 Field Documentation

- 8.433.2.1 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]`
- 8.433.2.2 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication`
- 8.433.2.3 `struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList`
- 8.433.2.4 `struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS`
- 8.433.2.5 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4`
- 8.433.2.6 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag`
- 8.433.2.7 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference`
- 8.433.2.8 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4`
- 8.433.2.9 `struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6AddrInfo`
- 8.433.2.10 `struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6GWAddrInfo`
- 8.433.2.11 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu`
- 8.433.2.12 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO`
- 8.433.2.13 `struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList`
- 8.433.2.14 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPTtype`
- 8.433.2.15 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4`
- 8.433.2.16 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]`
- 8.433.2.17 `struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID`
- 8.433.2.18 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]`

- 8.433.2.19 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4`
- 8.433.2.20 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]`
- 8.433.2.21 `struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList`
- 8.433.2.22 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4`
- 8.433.2.23 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology`
- 8.433.2.24 `LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS`
- 8.433.2.25 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]`

8.434 `unpack_wds_SLQSModifyProfile_t` Struct Reference

Data Fields

- `uint16_t * pExtErrorCode`

8.434.1 Detailed Description

Parameters

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

8.434.2 Field Documentation

- 8.434.2.1 `uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode`

8.435 `unpack_wds_SLQSSetIPFamilyPreference_t` Struct Reference

Data Fields

- `uint16_t Tlvresult`

8.435.1 Detailed Description

Parameters

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

8.435.2 Field Documentation

- 8.435.2.1 `uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult`

8.436 `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](#)

8.436.1 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>sessionEnd-Reason</i>	Call End Reason
<i>verboseSessn-EndReasonType</i>	Verbose call end reason type
<i>verboseSessn-EndReason</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.

8.436.2 Field Documentation

- 8.436.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID
- 8.436.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status
- 8.436.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily
- 8.436.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd
- 8.436.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason
- 8.436.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName
- 8.436.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason
- 8.436.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

8.437 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]

8.437.1 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

8.437.2 Field Documentation

8.437.2.1 [uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail](#)

8.437.2.2 [wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo](#)[255]

8.437.2.3 [uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail](#)

8.437.2.4 [uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail](#)

8.437.2.5 [uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology](#)

8.437.2.6 [uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail](#)

8.437.2.7 [uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus](#)

8.437.2.8 [uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail](#)

- 8.437.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus
- 8.437.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen
- 8.437.2.11 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork
- 8.437.2.12 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask
- 8.437.2.13 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes
- 8.437.2.14 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts
- 8.437.2.15 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask
- 8.437.2.16 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes
- 8.437.2.17 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts
- 8.437.2.18 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

8.438 unpack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference

Data Fields

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

8.438.1 Detailed Description

Parameters

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

8.438.2 Field Documentation

- 8.438.2.1 [wdsDhcpv4HwConfig](#)* [unpack_wds_SLQSSetDHCPv4ClientConfig_t::pHwConfig](#)
- 8.438.2.2 [wdsDhcpv4OptionList](#)* [unpack_wds_SLQSSetDHCPv4ClientConfig_t::pRequestOptionList](#)

8.439 unpack_wds_SLQSSetLoopback_t Struct Reference

Data Fields

- uint8_t [ByteLoopbackMode](#)
- uint8_t [ByteLoopbackMultiplier](#)

8.439.1 Detailed Description

Parameters

<i>ByteLoopback-Mode</i>	<ul style="list-style-type: none"> • Loopback Mode. <ul style="list-style-type: none"> – 0 - Disable – 1 - Enable
<i>ByteLoopback-Multiplier</i>	<ul style="list-style-type: none"> • Loopback multiplier. Number of downlink bytes to send for each uplink byte.

8.439.2 Field Documentation

8.439.2.1 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode

8.439.2.2 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier

8.440 unpack_wds_SLQSStartDataSession_t Struct Reference

Data Fields

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

8.440.1 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

8.440.2 Field Documentation

8.440.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

8.440.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

8.440.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

8.440.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

8.441 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]

8.441.1 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

8.441.2 Field Documentation

8.441.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

- 8.441.2.2 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId`
- 8.441.2.3 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`
- 8.441.2.4 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address`
- 8.441.2.5 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress`
- 8.441.2.6 `struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address`
- 8.441.2.7 `struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress`
- 8.441.2.8 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address`
- 8.441.2.9 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`
- 8.441.2.10 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`
- 8.441.2.11 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`
- 8.441.2.12 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`
- 8.441.2.13 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`
- 8.441.2.14 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`
- 8.441.2.15 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

8.442 UnPackGetProfileSettingOut Struct Reference

Data Fields

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

8.442.1 Field Documentation

- 8.442.1.1 `UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile`
- 8.442.1.2 `uint16_t* UnPackGetProfileSettingOut::pExtErrCode`

8.443 unpackWdsProfileParam Union Reference

Data Fields

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

8.443.1 Field Documentation

- 8.443.1.1 `LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP`
- 8.443.1.2 `LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2`

8.444 wds_currNetworkInfo Struct Reference

Data Fields

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

8.444.1 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
---------------	--

8.444.2 Field Documentation

8.444.2.1 `uint8_t wds_currNetworkInfo::NetworkType`

8.444.2.2 `uint32_t wds_currNetworkInfo::RATMask`

8.444.2.3 `uint32_t wds_currNetworkInfo::SOMask`

8.445 wds_Domain Struct Reference

Data Fields

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

8.445.1 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)

8.445.2 Field Documentation

8.445.2.1 `uint16_t wds_Domain::domainLen`

8.445.2.2 `uint8_t wds_Domain::domainName[256]`

8.446 wds_DomainNameList Struct Reference

Data Fields

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

8.446.1 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)

8.446.2 Field Documentation

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

8.446.2.2 `uint8_t wds_DomainNameList::numInstances`

8.447 wds_GPRSQoS Struct Reference

Data Fields

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

8.447.1 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

8.447.2 Field Documentation

8.447.2.1 `uint32_t wds_GPRSQoS::delayClass`

8.447.2.2 `uint32_t wds_GPRSQoS::meanThroughputClass`

8.447.2.3 `uint32_t wds_GPRSQoS::peakThroughputClass`

8.447.2.4 `uint32_t wds_GPRSQoS::precedenceClass`

8.447.2.5 `uint32_t wds_GPRSQoS::reliabilityClass`

8.448 wds_IPV6AddressInfo Struct Reference

Data Fields

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

8.448.1 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.

8.448.2 Field Documentation

8.448.2.1 `uint16_t wds_IPV6AddressInfo::IPAddressV6[8]`

8.448.2.2 `uint8_t wds_IPV6AddressInfo::IPv6PrefixLen`

8.449 wds_IPV6GWAddressInfo Struct Reference

Data Fields

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

8.449.1 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.

8.449.2 Field Documentation

8.449.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

8.449.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

8.450 wds_PCSCFFQDNAddress Struct Reference

Data Fields

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

8.450.1 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)

8.450.2 Field Documentation

8.450.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

8.450.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

8.451 wds_PCSCFFQDNAddressList Struct Reference

Data Fields

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

8.451.1 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)

8.451.2 Field Documentation

8.451.2.1 `uint8_t wds_PCSCFFQDNAddressList::numInstances`

8.451.2.2 `struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]`

8.452 wds_PCSCFIPv4ServerAddressList Struct Reference

Data Fields

- `uint8_t numInstances`
- `uint32_t pscsfIPv4Addr [64]`

8.452.1 Detailed Description

This structure contains the PCSCFIPv4ServerAddressList Information

Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> • number of address following
<i>pscsfIPv4Addr</i>	<ul style="list-style-type: none"> • P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)

8.452.2 Field Documentation

8.452.2.1 `uint8_t wds_PCSCFIPv4ServerAddressList::numInstances`

8.452.2.2 `uint32_t wds_PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]`

8.453 wds_ProfileIdentifier Struct Reference

Data Fields

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

8.453.1 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

8.453.2 Field Documentation

8.453.2.1 uint8_t wds_ProfileIdentifier::profileIndex

8.453.2.2 uint8_t wds_ProfileIdentifier::profileType

8.454 wds_profileInfo Union Reference**Data Fields**

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

8.454.1 Detailed Description

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

8.454.2 Field Documentation

8.454.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

8.454.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

8.455 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](#)

8.455.1 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.

8.455.2 Field Documentation

8.455.2.1 `uint8_t wds_UMTSMinQoS::deliveryErrSDU`

8.455.2.2 `uint32_t wds_UMTSMinQoS::grntDownlinkBitrate`

8.455.2.3 `uint32_t wds_UMTSMinQoS::grntUplinkBitrate`

8.455.2.4 `uint32_t wds_UMTSMinQoS::maxDownlinkBitrate`

8.455.2.5 `uint32_t wds_UMTSMinQoS::maxSDUSize`

8.455.2.6 `uint32_t wds_UMTSMinQoS::maxUplinkBitrate`

8.455.2.7 `uint8_t wds_UMTSMinQoS::qosDeliveryOrder`

8.455.2.8 `uint8_t wds_UMTSMInQoS::resBerRatio`

8.455.2.9 `uint8_t wds_UMTSMInQoS::sduErrorRatio`

8.455.2.10 `uint8_t wds_UMTSMInQoS::trafficClass`

8.455.2.11 `uint32_t wds_UMTSMInQoS::trafficPriority`

8.455.2.12 `uint32_t wds_UMTSMInQoS::transferDelay`

8.456 wdsDhcpv4HwConfig Struct Reference

Data Fields

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

8.456.1 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

8.456.2 Field Documentation

8.456.2.1 `uint8_t wdsDhcpv4HwConfig::chaddr`[16]

8.456.2.2 `uint8_t wdsDhcpv4HwConfig::chaddrLen`

8.456.2.3 `uint8_t wdsDhcpv4HwConfig::hwType`

8.457 wdsDhcpv4Option Struct Reference

Data Fields

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

8.457.1 Detailed Description

Parameters

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

8.457.2 Field Documentation

8.457.2.1 `uint8_t wdsDhcpv4Option::optCode`8.457.2.2 `uint8_t wdsDhcpv4Option::optVal[255]`8.457.2.3 `uint8_t wdsDhcpv4Option::optValLen`

8.458 wdsDhcpv4OptionList Struct Reference

Data Fields

- `uint8_t numOpt`
- `wdsDhcpv4Option * pOptList`

8.458.1 Detailed Description

Parameters

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

8.458.2 Field Documentation

8.458.2.1 `uint8_t wdsDhcpv4OptionList::numOpt`8.458.2.2 `wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList`

8.459 wdsDhcpv4ProfileId Struct Reference

Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

8.459.1 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)
------------------	---

8.459.2 Field Documentation

8.459.2.1 `uint8_t wdsDhcpv4ProfileId::profileId`

8.459.2.2 `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](#)
- struct [unpack_dms_GetManufacturer_t](#)
- struct [unpack_dms_GetOfflineReason_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)
- int [pack_dms_GetManufacturer](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_GetManufacturer](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetManufacturer_t](#) *pOutput)
- int [pack_dms_GetOfflineReason](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_dms_GetOfflineReason](#) (uint8_t *pResp, uint16_t respLen, [unpack_dms_GetOfflineReason_t](#) *pOutput)

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>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

9.3.2.3 `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>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

9.3.2.4 `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_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
in	<i>reqArg</i>	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_GetManufacturer (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

To get device manufacturer information.pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.3.2.18 `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

9.3.2.19 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

9.3.2.20 int pack_dms_GetOfflineReason (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

To get operating mode offline reason pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.3.2.21 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

9.3.2.22 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

9.3.2.23 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

9.3.2.24 `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

9.3.2.25 `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

9.3.2.26 `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>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

9.3.2.27 `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>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_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>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_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

9.3.2.30 `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

9.3.2.31 `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

9.3.2.32 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

9.3.2.33 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

9.3.2.34 `int pack_dms_SLQSDmsSwiIndicationRegister (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSDmsSwiIndicationRegister_t * reqArg)`

Set the registration state for different indication pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

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

9.3.2.35 `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

9.3.2.36 `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

9.3.2.37 `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

9.3.2.38 `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

9.3.2.39 `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

9.3.2.40 `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

9.3.2.41 `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

9.3.2.42 `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

9.3.2.43 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

9.3.2.44 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

9.3.2.45 `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

9.3.2.46 `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

9.3.2.47 `int unpack_dms_GetCustFeaturesV2 (uint8_t * pResp, uint16_t respLen, unpack_dms_GetCustFeaturesV2_t * pOutput)`

9.3.2.48 `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

9.3.2.49 `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

9.3.2.50 `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

9.3.2.51 `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

9.3.2.52 `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

9.3.2.53 `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

9.3.2.54 `int unpack_dms_GetFirmwareRevision (uint8_t * pResp, uint16_t respLen, unpack_dms_GetFirmwareRevision_t * pOutput)`

get Firmware Revision unpack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.3.2.55 `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

9.3.2.56 `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

9.3.2.57 `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

9.3.2.58 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

9.3.2.59 int unpack_dms_GetManufacturer (uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetManufacturer_t * *pOutput*)

To get device manufacturer information unpack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.3.2.60 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

9.3.2.61 `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

9.3.2.62 `int unpack_dms_GetOfflineReason (uint8_t * pResp, uint16_t respLen, unpack_dms_GetOfflineReason_t * pOutput)`

To get operating mode offline reason unpack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.3.2.63 `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

9.3.2.64 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

9.3.2.65 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

9.3.2.66 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

9.3.2.67 `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

9.3.2.68 `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

9.3.2.69 `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

9.3.2.70 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

9.3.2.71 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

9.3.2.72 `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

9.3.2.73 `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

9.3.2.74 `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

9.3.2.75 `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

9.3.2.76 `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

9.3.2.77 `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

9.3.2.78 `int unpack_dms_SLQSDmsSwiIndicationRegister (uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiIndicationRegister_t * pOutput)`

Set the registration state for different indication unpack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

Note

support EM/MC74xx onwards

See Also

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

9.3.2.79 `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

9.3.2.80 `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

9.3.2.81 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

9.3.2.82 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

9.3.2.83 `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

9.3.2.84 `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

9.3.2.85 `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

9.3.2.86 `int unpack_dms_UIMGetICCID (uint8_t * pResp, uint16_t respLen, unpack_dms_UIMGetICCID_t * pOutput)`

Unpacks the UIMGetICCID response message to a user-provided response structure.

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.4 fms.h File Reference

Data Structures

- struct [CarrierImage_t](#)
- struct [pack_fms_GetImagesPreference_t](#)
- struct [FMSImageElement](#)
- struct [FMSPrefImageList](#)
- struct [unpack_fms_GetImagesPreference_t](#)
- struct [pack_fms_GetStoredImages_t](#)
- struct [FMSImageIdElement](#)
- struct [FMSImageIdEntries](#)
- struct [FMSImageList](#)
- struct [unpack_fms_GetStoredImages_t](#)
- struct [pack_fms_SetImagesPreference_t](#)
- struct [unpack_fms_SetImagesPreference_t](#)

Macros

- `#define FMS_GOBI_MBN_IMG_ID_STR_LEN 16`
- `#define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100`
- `#define FMS_GOBI_LISTENTRIES_MAX 2`
- `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`
- `#define FMS_MAX_IMAGE_ID_ELEMENT 50`
- `#define FMS_IMAGE_ID_MAX_ENTRIES 2`
- `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`
- `#define FMS_IMAGE_ID_IMG_ID_LEN 16`
- `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`
- `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

Functions

- int [pack_fms_GetImagesPreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_fms_GetImagesPreference_t](#) *reqArg)
- int [unpack_fms_GetImagesPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_fms_GetImagesPreference_t](#) *pOutput)
- int [pack_fms_GetStoredImages](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_fms_GetStoredImages_t](#) *reqArg)
- int [unpack_fms_GetStoredImages](#) (uint8_t *pResp, uint16_t respLen, [unpack_fms_GetStoredImages_t](#) *pOutput)
- int [pack_fms_SetImagesPreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_fms_SetImagesPreference_t](#) *reqArg)
- int [unpack_fms_SetImagesPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_fms_SetImagesPreference_t](#) *pOutput)
- uint32_t [GetValidFwPriCombinations](#) ([FMSImageList](#) *pStoredImageList, uint32_t *pValidCombinationSize, [CarrierImage_t](#) *pValidCombinations)

9.4.1 Macro Definition Documentation

9.4.1.1 `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`

9.4.1.2 `#define FMS_GOBI_LISTENTRIES_MAX 2`

9.4.1.3 `#define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100`

9.4.1.4 `#define FMS_GOBI_MBN_IMG_ID_STR_LEN 16`

9.4.1.5 `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`

9.4.1.6 `#define FMS_IMAGE_ID_IMG_ID_LEN 16`

9.4.1.7 `#define FMS_IMAGE_ID_MAX_ENTRIES 2`

9.4.1.8 `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

9.4.1.9 `#define FMS_MAX_IMAGE_ID_ELEMENT 50`

9.4.1.10 `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`

9.4.2 Function Documentation

9.4.2.1 `uint32_t GetValidFwPriCombinations (FMSImageList * pStoredImageList, uint32_t * pValidCombinationSize, CarrierImage_t * pValidCombinations)`

This API distills valid Firmware/PRI combinations from `GetStoredImages` result

Parameters

in	<i>pStoredImageList</i>	<ul style="list-style-type: none"> • 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 libsdp.h File Reference

Data Structures

- struct [_libSDP_FirmwareInfo_](#)

Macros

- #define [IMG_MASK_MDM](#) (1<<0)
- #define [IMG_MASK_PRI](#) (1<<1)
- #define [IMG_MASK_CLEAR](#) (0x000)
- #define [IMG_MASK_GENERIC](#) (0xFFFF)
- #define [LIBSDP_CARRIER_PACKAGE_SKU](#) "9999999"
- #define [LIBSDP_SKU_STRING_LENGTH](#) 7
- #define [FIRMWARE_INFO_STRING_SIZE](#) 20

Typedefs

- typedef struct [_libSDP_FirmwareInfo_](#) [libSDP_FirmwareInfo](#)

- typedef void(* [libsdh_logger](#))(uint8_t lvl, const char *buff)

Enumerations

- enum [libSDP_fwdwl_error_codes](#) {
[eSDP_FWDWL_SUCCESS](#) = 0,
[eSDP_FWDWL_ERR_GENERAL](#) = 100,
[eSDP_FWDWL_ERR_SDK](#),
[eSDP_FWDWL_ERR_SET_CBK](#),
[eSDP_FWDWL_ERR_PATH_TOO_LONG](#),
[eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED](#),
[eSDP_FWDWL_ERR_FW_UPGRADE](#),
[eSDP_FWDWL_ERR_INVALID_DEV](#),
[eSDP_FWDWL_ERR_INVALID_PATH](#),
[eSDP_FWDWL_ERR_TIMEOUT](#),
[eSDP_FWDWL_ERR_FAIL](#),
[eSDP_FWDWL_ERR_PRI_FAIL](#),
[eSDP_FWDWL_ERR_FW_VERSION_FAIL](#),
[eSDP_FWDWL_ERR_END](#) }
- enum [libSDP_Fw_Type](#) {
[eFW_TYPE_MBN_GOBI](#),
[eFW_TYPE_MBN](#),
[eFW_TYPE_CWE](#),
[eFW_TYPE_NVU](#),
[eFW_TYPE_SPK](#),
[eFW_TYPE_INVALID](#),
[eFW_TYPE_CWE_NVU](#) }
- enum [libSDP_Models](#) {
[eModel_Unknown](#) = -1,
[eModel_9X15](#) = 0,
[eModel_9X30](#) = 1 }

Functions

- int [libSDP_ExtractFirmwareParametersByPath](#) (char *pImagePath, [libSDP_FirmwareInfo](#) *info)
- int [libSDP_BuildImagesPreferenceRequest](#) ([libSDP_FirmwareInfo](#) info, [pack_fms_SetImagesPreference_t](#) *pack)
- int [libSDP_CalculateImageMask](#) ([unpack_fms_SetImagesPreference_t](#) SetPrefRspFromModem)
- int [libSDP_getFileType](#) (char *szPath)
- unsigned int [libSDP_DownloadFW](#) (char *pImagePath, char *szTTYPath, int iFWImageType, int image_mask, int iModelFamily)
- int [libSDP_GetModelFamily](#) (char *pModelString)
- int [libSDP_CheckValidFirmwareInfo](#) ([libSDP_FirmwareInfo](#) info)
- char * [libSDP_GetVersion](#) ()
- int [libsdh_set_log_func](#) ([libsdh_logger](#) func)

9.5.1 Detailed Description

Filename: [libsdh.h](#)

Purpose: Global definitions used inside the SDK

Copyright: © 2016 Sierra Wireless Inc., all rights reserved

9.5.2 Macro Definition Documentation

9.5.2.1 `#define FIRMWARE_INFO_STRING_SIZE 20`

9.5.2.2 `#define IMG_MASK_CLEAR (0x000)`

9.5.2.3 `#define IMG_MASK_GENERIC (0xFFFF)`

9.5.2.4 `#define IMG_MASK_MDM (1<<0)`

9.5.2.5 `#define IMG_MASK_PRI (1<<1)`

9.5.2.6 `#define LIBSDP_CARRIER_PACKAGE_SKU "9999999"`

9.5.2.7 `#define LIBSDP_SKU_STRING_LENGTH 7`

9.5.3 Typedef Documentation

9.5.3.1 `typedef struct _libSDP_FirmwareInfo libSDP_FirmwareInfo`

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

Parameters

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

9.5.3.2 `typedef void(* libsdplogger)(uint8_t lvl, const char *buff)`

This Custom Log prototype.

Parameters

<i>lvl</i> [IN]	<ul style="list-style-type: none"> Log level.
<i>buff</i> [IN]	<ul style="list-style-type: none"> Log String.

Returns

none

9.5.4 Enumeration Type Documentation

9.5.4.1 enum libSDP_Fw_Type

Enumerator

eFW_TYPE_MBN_GOBI

eFW_TYPE_MBN

eFW_TYPE_CWE

eFW_TYPE_NVU

eFW_TYPE_SPK

eFW_TYPE_INVALID

eFW_TYPE_CWE_NVU

9.5.4.2 enum libSDP_fwdwl_error_codes

Enumerator

eSDP_FWDWL_SUCCESS

eSDP_FWDWL_ERR_GENERAL

eSDP_FWDWL_ERR_SDK

eSDP_FWDWL_ERR_SET_CBK

eSDP_FWDWL_ERR_PATH_TOO_LONG

eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED

eSDP_FWDWL_ERR_FW_UPGRADE

eSDP_FWDWL_ERR_INVALID_DEV

eSDP_FWDWL_ERR_INVALID_PATH

eSDP_FWDWL_ERR_TIMEOUT

eSDP_FWDWL_ERR_FAIL

eSDP_FWDWL_ERR_PRI_FAIL

eSDP_FWDWL_ERR_FW_VERSION_FAIL

eSDP_FWDWL_ERR_END

9.5.4.3 enum libSDP_Models

Enumerator

eModel_Unknown

eModel_9X15

eModel_9X30

9.5.5 Function Documentation

9.5.5.1 `int libSDP_BuildImagesPreferenceRequest (libSDP_FirmwareInfo info, pack_fms_SetImagesPreference_t * pack)`

This API Build Image Preference Request Using Firmware Information.

Parameters

<i>info</i> [IN]	<ul style="list-style-type: none"> See libSDP_FirmwareInfo for more information.
<i>pack</i> [OUT]	<ul style="list-style-type: none"> See pack_fms_SetImagesPreference_t for more information.

Returns

0 on success, libSDP_fwdwl_error_codes error value otherwise

See Also

See [libSDP_fwdwl_error_codes](#) for error values

9.5.5.2 `int libSDP_CalculateImageMask (unpack_fms_SetImagesPreference_t SetPrefRspFromModem)`

This API Calculate Image Mask for Firmware Download.

Parameters

<i>SetPrefRsp-FromModem</i> [IN]	<ul style="list-style-type: none"> See unpack_fms_SetImagesPreference_t for more information.
----------------------------------	--

Returns

Image Mask

- IMG_MASK_MDM | IMG_MASK_PRI

9.5.5.3 `int libSDP_CheckValidFirmwareInfo (libSDP_FirmwareInfo info)`

This API Check Valid Firmware Information to build Image Preference Request.

Parameters

<i>info</i> [IN]	<ul style="list-style-type: none"> See libSDP_FirmwareInfo for more information.
------------------	---

Returns

0 on success, -1 error value otherwise

9.5.5.4 unsigned int libSDP_DownloadFW (char * *plImagePath*, char * *szTTYPath*, int *iFWImageType*, int *image_mask*, int *iModelFamily*)

This API Download Firmware.

Parameters

<i>plImagePath</i> [IN]	<ul style="list-style-type: none"> Firmware Folder Path.
<i>szTTYPath</i> [IN]	<ul style="list-style-type: none"> QDL Device Path.
<i>iFWImageType</i> [IN]	<ul style="list-style-type: none"> Firmware Type. See libSDP_Fw_Type
<i>image_mask</i> [IN]	<ul style="list-style-type: none"> Image Mask. <ul style="list-style-type: none"> – IMG_MASK_MDM IMG_MASK_PRI
<i>iModelFamily</i> [IN]	<ul style="list-style-type: none"> Modem Family. See libSDP_Models

Returns

0 on success, libSDP_fwdwl_error_codes error value otherwise

See Also

See [libSDP_fwdwl_error_codes](#) for error values

9.5.5.5 int libSDP_ExtractFirmwareParametersByPath (char * *plImagePath*, libSDP_FirmwareInfo * *info*)

This API Extrac Firmwre Parameters From Path.

Parameters

<i>plImagePath</i> [IN]	<ul style="list-style-type: none"> Firmware Folder Path.
<i>libSDP_FirmwareInfo</i> [OUT]	<ul style="list-style-type: none"> See libSDP_FirmwareInfo for more information.

Returns

0 on success, libSDP_fwdwl_error_codes error value otherwise

See Also

See [libSDP_fwdwl_error_codes](#) for error values

9.5.5.6 int libSDP_getFileType (char * *szPath*)

This API Get File Type By Path

Parameters

<i>szPath</i> [IN]	<ul style="list-style-type: none">• See libSDP_FirmwareInfo for more information.
--------------------	---

Returns

eFW_TYPE_INVALID on error, libSDP_Fw_Type value otherwise

See Also

See [libSDP_Fw_Type](#) for values

9.5.5.7 int libSDP_GetModelFamily (char * *pModelString*)

This API Get Model Family from a model string.

Parameters

<i>pModelString</i> [IN]	<ul style="list-style-type: none">• Model String.
--------------------------	---

Returns

eModel_Unknown on Error, libSDP_Models value otherwise

See Also

See [libSDP_Models](#) for values

9.5.5.8 char* libSDP_GetVersion ()

This API Get Lib SDP Version.

Parameters

<i>No;</i>	
------------	--

Returns

Version String

9.5.5.9 int libsdp_set_log_func (libsdplogger *func*)

This API Set Custom Log function.

Parameters

<i>func</i> [IN]	<ul style="list-style-type: none"> • See libsdplogger for more information.
------------------	--

Returns

none

9.6 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 LOCEVENTMASKSETSPSTREAMINGREPORT 0x00001000`
- `#define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000`
- `#define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000`
- `#define LOCEVENTMASKGEOFENCEGENALERT 0x00008000`
- `#define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000`
- `#define LOCEVENTMASKPEDOMETERCONTROL 0x00020000`
- `#define LOCEVENTMASKMOTIONDATACONTROL 0x00040000`
- `#define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000`
- `#define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000`
- `#define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000`
- `#define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000`
- `#define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000`
- `#define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000`
- `#define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF`

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)

9.6.1 Macro Definition Documentation

9.6.1.1 `#define LOC_UINT8_MAX_STRING_SZ 255`

9.6.1.2 `#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.

9.6.1.3 `#define LOCEVENTMASKENGINESTATE 0x00000080`

The control point must enable this mask to receive engine state report event indications.

9.6.1.4 `#define LOCEVENTMASKFIXSESSIONSTATE 0x00000100`

The control point must enable this mask to receive fix session status report event indications.

9.6.1.5 `#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.

9.6.1.6 `#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.

9.6.1.7 `#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.

9.6.1.8 `#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).

9.6.1.9 `#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.

9.6.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040

The control point must enable this mask to receive position injection request event indications.

9.6.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020

The control point must enable this mask to receive predicted orbits request event indications.

9.6.1.12 #define LOCEVENTMASKINJECTTIMERREQ 0x00000010

The control point must enable this mask to receive time injection request event indications.

9.6.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000

The control point must enable this mask to receive Wi-Fi Access Point (AP) data inject request event indications.

9.6.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF

Invalid Event Mask

9.6.1.15 #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.

9.6.1.16 #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.

9.6.1.17 #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.

9.6.1.18 #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.

9.6.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008

The control point must enable this mask to receive NI Notify/Verify request event indications.

9.6.1.20 #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.

9.6.1.21 #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.

9.6.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001

The control point must enable this mask to receive position report event indications.

9.6.1.23 #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.).

9.6.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000

The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications.

9.6.1.25 #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.

9.6.1.26 #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.).

9.6.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200

The control point must enable this mask to receive Wi-Fi position request event indications.

9.6.2 Enumeration Type Documentation**9.6.2.1 anonymous enum**

Enumerator

```
eQMI_LOC_SESS_STATUS_SUCCESS
eQMI_LOC_SESS_STATUS_IN_PROGRESS
eQMI_LOC_SESS_STATUS_FAILURE
eQMI_LOC_SESS_STATUS_TIMEOUT
```

9.6.3 Function Documentation

```
9.6.3.1 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

9.6.3.2 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

9.6.3.3 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

9.6.3.4 `int pack_loc_SetOperationMode (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetOperationMode_t * reqArg)`

Set Operation Mode pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.6.3.5 `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>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.6.3.6 `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>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.6.3.7 `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

9.6.3.8 `int unpack_loc_BestAvailPos_Ind (uint8_t * pResp, uint16_t respLen, unpack_loc_BestAvailPos_Ind_t * pOutput)`

Loc Best Avial position Indication unpack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.6.3.9 `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

9.6.3.10 `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

9.6.3.11 `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

9.6.3.12 `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

9.6.3.13 `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

9.6.3.14 `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

9.6.3.15 `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

9.6.3.16 `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

9.6.3.17 `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

9.6.3.18 `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

9.7 nas.h File Reference

Data Structures

- struct [unpack_nas_GetSignalStrengths_t](#)
- struct [unpack_nas_SLQSGetSysSelectionPref_t](#)
- struct [nas_netSelectionPref](#)
- struct [nas_acqOrderPref](#)
- struct [nas_CSGID](#)
- struct [pack_nas_SLQSSetSysSelectionPref_t](#)
- struct [pack_nas_SLQSNasIndicationRegisterExt_t](#)
- struct [RFBandInfoElements](#)
- struct [unpack_nas_GetRFInfo_t](#)
- struct [cdmaSSInfo](#)
- struct [hdrSSInfo](#)
- struct [lteSSInfo](#)
- struct [tdscdmaSigInfoExt](#)
- struct [unpack_nas_SLQSNasGetSigInfo_t](#)
- struct [unpack_nas_SLQSNasSigInfoCallback_ind_t](#)
- struct [unpack_nas_GetHomeNetwork_t](#)
- struct [nas_SrvStatusInfo](#)
- struct [nas_GSMSrvStatusInfo](#)
- struct [nas_sysInfoCommon](#)
- struct [nas_CDMASysInfo](#)
- struct [nas_HDRSysInfo](#)
- struct [nas_GSMSysInfo](#)
- struct [nas_WCDMASysInfo](#)
- struct [nas_LTESysInfo](#)
- struct [nas_AddCDMASysInfo](#)
- struct [nas_AddSysInfo](#)
- struct [nas_CallBarringSysInfo](#)
- struct [unpack_nas_SLQSGetSysInfo_t](#)
- struct [unpack_nas_SLQSSysInfoCallback_ind_t](#)
- struct [unpack_nas_GetServingNetwork_t](#)
- struct [unpack_nas_GetServingNetworkCapabilities_t](#)
- struct [nas_QmiNas3GppNetworkInfo](#)
- struct [nas_QmiNas3GppNetworkRAT](#)
- struct [nas_QmisNasPcsDigit](#)
- struct [unpack_nas_PerformNetworkScan_t](#)
- struct [unpack_nas_SLQSSwiGetLteCQI_t](#)
- struct [nas_CommInfo](#)
- struct [nas_LTEInfo](#)
- struct [unpack_nas_SLQSNasSwiModemStatus_t](#)
- struct [nas_servSystem](#)
- struct [nas_dataSrvCapabilities](#)
- struct [nas_currentPLMN](#)

- struct [nas_roamIndList](#)
- struct [nas_qaQmi3Gpp2TimeZone](#)
- struct [nas_detailSvcInfo](#)
- struct [nas_CDMASysInfoExt](#)
- struct [nas_callBarStatus](#)
- struct [unpack_nas_SLQSGetServingSystem_t](#)
- struct [nas_rxSignalStrengthListElement](#)
- struct [nas_ecioListElement](#)
- struct [nas_errorRateListElement](#)
- struct [nas_rsrqInformation](#)
- struct [nas_lteSnrinformation](#)
- struct [nas_lteRsrpinformation](#)
- struct [unpack_nas_SLQSGetSignalStrength_t](#)
- struct [nas_SLQSSignalStrengthsIndReq](#)
- struct [pack_nas_SLQSSetSignalStrengthsCallback_t](#)
- struct [nas_SLQSSignalStrengthsInformation](#)
- struct [nas_RejectReasonTlv](#)
- struct [nas_SignalStrengthTlv](#)
- struct [nas_RFInfoTlv](#)
- struct [nas_SLQSSignalStrengthsTlv](#)
- struct [unpack_nas_SetEventReportInd_t](#)
- struct [unpack_nas_GetCDMANetworkParameters_t](#)
- struct [pack_nas_SetACCOLC_t](#)
- struct [nas_CDMARSSIThresh](#)
- struct [nas_CDMAECIOThresh](#)
- struct [nas_HDRRSSIThresh](#)
- struct [nas_HDRECIOThresh](#)
- struct [nas_HDRSINRThreshold](#)
- struct [nas_HDRIOThresh](#)
- struct [nas_GSMRSSIThresh](#)
- struct [nas_WCDMARSSIThresh](#)
- struct [nas_WCDMAECIOThresh](#)
- struct [nas_LTERSSIThresh](#)
- struct [nas_LTESNRThreshold](#)
- struct [nas_LTERSRQThresh](#)
- struct [nas_LTERSRPThresh](#)
- struct [nas_LTESigRptConfig](#)
- struct [nas_TDSCDMARSCPTthresh](#)
- struct [nas_TDSCDMARSSIThresh](#)
- struct [nas_TDSCDMAECIOThresh](#)
- struct [nas_TDSCDMASINRThresh](#)
- struct [pack_nas_SLQSNasConfigSigInfo2_t](#)
- struct [unpack_nas_SetDataCapabilitiesCallback_ind_t](#)
- struct [unpack_nas_GetNetworkPreference_t](#)
- struct [pack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetNetworkPreference_t](#)
- struct [unpack_nas_SetRoamingIndicatorCallback_ind_t](#)
- struct [NAServingSystemInfo](#)
- struct [unpack_nas_SetServingSystemCallback_ind_t](#)
- struct [NASPhyCaAggScellIndType](#)
- struct [NASPhyCaAggScellIDBw](#)
- struct [NASPhyCaAggScellInfo](#)
- struct [NASPhyCaAggPcellInfo](#)
- struct [NASPhyCaAggScellIndex](#)
- struct [NasGetLTECphyCaInfo](#)

- struct [unpack_nas_SlqsGetLTECphyCAInfo_t](#)
- struct [NASEmergencyModeTlv](#)
- struct [NASModePreferenceTlv](#)
- struct [NASBandPreferenceTlv](#)
- struct [NASPRLPreferenceTlv](#)
- struct [NASRoamPreferenceTlv](#)
- struct [NASLTEBandPreferenceTlv](#)
- struct [NASNetSelPreferenceTlv](#)
- struct [NASServDomainPrefTlv](#)
- struct [NASGWAcqOrderPrefTlv](#)
- struct [NASQmiCbkNasSystemSelPrefInd](#)
- struct [unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t](#)
- struct [NASOTAMessageTlv](#)
- struct [NASLteNasReleaseInfoTlv](#)
- struct [NASTimeInfoTlv](#)
- struct [NASQmiCbkNasSwiOTAMessageInd](#)
- struct [unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t](#)
- struct [nas_MNRInfo](#)
- struct [pack_nas_SLQSInitiateNetworkRegistration_t](#)
- struct [pack_nas_SLQSNasSwiOTAMessageCallback_t](#)
- struct [pack_nas_SLQSGetPLMNName_t](#)
- struct [unpack_nas_SLQSGetPLMNName_t](#)
- struct [nas_nmrCellInfo](#)
- struct [nas_GERANInfo](#)
- struct [nas_geranInstInfo](#)
- struct [nas_UMTSinstInfo](#)
- struct [nas_UMTSInfo](#)
- struct [nas_CDMAInfo](#)
- struct [nas_cellParams](#)
- struct [nas_LTEInfoIntraFreq](#)
- struct [nas_infoInterFreq](#)
- struct [nas_LTEInfoInterFreq](#)
- struct [nas_gsmCellInfo](#)
- struct [nas_lteGsmCellInfo](#)
- struct [nas_LTEInfoNeighboringGSM](#)
- struct [nas_wcdmaCellInfo](#)
- struct [nas_lteWcdmaCellInfo](#)
- struct [nas_LTEInfoNeighboringWCDMA](#)
- struct [nas_umtsLTENbrCell](#)
- struct [nas_WCDMAInfoLTENeighborCell](#)
- struct [unpack_nas_SLQSNasGetCellLocationInfo_t](#)
- struct [nas_timeInfo](#)
- struct [unpack_nas_SLQSGetNetworkTime_t](#)
- struct [nas_UniversalTime](#)
- struct [unpack_nas_SLQSNasNetworkTimeCallBack_ind_t](#)
- struct [nas_PhyCaAggScellIndType](#)
- struct [nas_PhyCaAggScellIDIBw](#)
- struct [nas_PhyCaAggScellInfo](#)
- struct [nas_PhyCaAggPcellInfo](#)
- struct [nas_PhyCaAggScellIndex](#)
- struct [unpack_nas_SetNasLTECphyCaIndCallback_ind_t](#)
- struct [nas_RxSigInfo](#)
- struct [nas_SccRxInfo](#)
- struct [unpack_nas_SLQSSwiGetLteSccRxInfo_t](#)

Macros

- `#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048`
- `#define NAS_MAX_NUM_NETWORKS 30`
- `#define NAS_MAX_DESCRIPTION_LENGTH 255`
- `#define NAS_PLMN_LENGTH 3`
- `#define NAS_MAX_SCC_RX_INFO_INSTANCES 255`
- `#define NAS_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_SLQSSetSysSelectionPref (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_nas_SLQSSetSysSelectionPref (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPref_t *pOutput)`
- `int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSysSelectionPref_t *pReqParam)`
- `int unpack_nas_SLQSSetSysSelectionPref (uint8_t *pResp, uint16_t respLen)`
- `int pack_nas_SLQSSetBandPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint64_t bandPref)`
- `int unpack_nas_SLQSSetBandPreference (uint8_t *pResp, uint16_t respLen)`
- `int pack_nas_SLQSNasIndicationRegisterExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasIndicationRegisterExt_t *pReqParam)`
- `int unpack_nas_SLQSNasIndicationRegisterExt (uint8_t *pResp, uint16_t respLen)`
- `int pack_nas_GetRFInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_nas_GetRFInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_GetRFInfo_t *pOutput)`
- `int pack_nas_SLQSNasGetSigInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_nas_SLQSNasGetSigInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetSigInfo_t *pOutput)`

- int [unpack_nas_SLQSNasSigInfoCallback_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_SetLURRejectCallback](#) (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int [unpack_nas_SetLURRejectCallback](#) (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_GetCDMANetworkParameters_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)
- int [pack_nas_SLQSSwiGetLteSccRxInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReq, uint16_t *pLen)
- int [unpack_nas_SLQSSwiGetLteSccRxInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_nas_SLQSSwiGetLteSccRxInfo_t](#) *pOutput)

9.7.1 Macro Definition Documentation

9.7.1.1 `#define NAS_MAX_DESCRIPTION_LENGTH 255`

9.7.1.2 `#define NAS_MAX_NUM_NETWORKS 30`

9.7.1.3 `#define NAS_MAX_SCC_RX_INFO_INSTANCES 255`

9.7.1.4 `#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048`

9.7.1.5 `#define NAS_PLMN_LENGTH 3`

9.7.1.6 #define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255

9.7.2 Enumeration Type Documentation

9.7.2.1 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

9.7.2.2 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

9.7.2.3 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

9.7.2.4 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

9.7.3 Function Documentation

9.7.3.1 int pack_nas_GetACCOLC (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.7.3.2 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

9.7.3.3 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

9.7.3.4 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

9.7.3.5 int pack_nas_GetNetworkPreference (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in, out	pCtx	qmi request context
out	pReq	qmi request buffer
out	pLen	qmi request length

9.7.3.6 int pack_nas_GetRFInfo (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

get rf info pack

Parameters

in, out	pCtx	qmi request context
out	pReq	qmi request buffer
out	pLen	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_XXX error value otherwise

See Also

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

9.7.3.7 int pack_nas_GetServingNetwork (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in, out	pCtx	qmi request context
out	pReq	qmi request buffer
out	pLen	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_XXX error value otherwise

See Also

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

9.7.3.8 int pack_nas_GetServingNetworkCapabilities (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)

Parameters

in, out	pCtx	qmi request context
out	pReq	qmi request buffer
out	pLen	qmi request length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.7.3.9 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

9.7.3.10 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

9.7.3.11 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

9.7.3.12 `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

9.7.3.13 `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

9.7.3.14 `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

9.7.3.15 `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

9.7.3.16 `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

9.7.3.17 `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

9.7.3.18 `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

9.7.3.19 `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

9.7.3.20 `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

9.7.3.21 `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

9.7.3.22 `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

9.7.3.23 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

9.7.3.24 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

9.7.3.25 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

9.7.3.26 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

9.7.3.27 int pack_nas_SLQSNasSwiModemStatus (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

9.7.3.28 int pack_nas_SLQSNasSwiOTAMessageCallback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasSwiOTAMessageCallback_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

9.7.3.29 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

9.7.3.30 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

9.7.3.31 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

9.7.3.32 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

9.7.3.33 int pack_nas_SLQSSwiGetLteSccRxInfo (pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen*)

get LTE Secondary carrier Rx signal level information pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.7.3.34 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

9.7.3.35 int unpack_nas_GetANAAAAuthenticationStatus (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

9.7.3.36 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

9.7.3.37 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

9.7.3.38 int unpack_nas_GetNetworkPreference (uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetNetworkPreference_t * *pOutput*)

9.7.3.39 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

9.7.3.40 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

9.7.3.41 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

9.7.3.42 `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

9.7.3.43 `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

9.7.3.44 `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

9.7.3.45 `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

9.7.3.46 `int unpack_nas_SetEventReportInd (uint8_t * pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *
pOutput)`

9.7.3.47 `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

9.7.3.48 `int unpack_nas_SetNasLTECphyCalndCallback_ind (uint8_t * pResp, uint16_t respLen,
unpack_nas_SetNasLTECphyCalndCallback_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

9.7.3.49 `int unpack_nas_SetNetworkPreference (uint8_t * pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t * pOutput)`

9.7.3.50 `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

9.7.3.51 `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

9.7.3.52 `int unpack_nas_SetServingSystemCallback_ind (uint8_t * pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t * pOutput)`

9.7.3.53 `int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t * pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t * pOutput)`

9.7.3.54 `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

9.7.3.55 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

9.7.3.56 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

9.7.3.57 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

9.7.3.58 `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

9.7.3.59 `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

9.7.3.60 `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

9.7.3.61 `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

9.7.3.62 `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

9.7.3.63 `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

9.7.3.64 `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

9.7.3.65 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

9.7.3.66 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

9.7.3.67 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

9.7.3.68 int unpack_nas_SLQSNasSwiOTAMessageCallback (uint8_t * *pResp*, uint16_t *respLen*)

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.7.3.69 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

9.7.3.70 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

9.7.3.71 `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

9.7.3.72 `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

9.7.3.73 `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

9.7.3.74 `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

9.7.3.75 `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

9.7.3.76 `int unpack_nas_SLQSSwiGetLteScCRxInfo (uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteScCRxInfo_t * pOutput)`

get LTE Secondary carrier Rx signal level information unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.8 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

9.8.1 Detailed Description

Network Access Service API Band Classes table.

9.8.2 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

9.8.2.1 LTE Bands

- 28 - LTE Band Class 28
- 39 - LTE Band Class 39
- 40 - LTE Band Class 40
- 41 - LTE Band Class 41
- 120 - FDD UL:1920-1980; DL:2110-2170; E-UTRA Operating Band 1
- 121 - FDD UL:1850-1910; DL:1930-1990; E-UTRA Operating Band 2
- 122 - FDD UL:1710-1785; DL:1805-1880; E-UTRA Operating Band 3
- 123 - FDD UL:1710-1755; DL:2110-2155; E-UTRA Operating Band 4
- 124 - FDD UL: 824- 849; DL: 869- 894; E-UTRA Operating Band 5
- 125 - FDD UL: 830- 840; DL: 875- 885; E-UTRA Operating Band 6
- 126 - FDD UL:2500-2570; DL:2620-2690; E-UTRA Operating Band 7
- 127 - FDD UL: 880- 915; DL: 925- 960; E-UTRA Operating Band 8
- 128 - FDD UL:1749.9-1784.9; DL:1844.9-1879.9; E-UTRA Operating Band 9
- 129 - FDD UL:1710-1770; DL:2110-2170; E-UTRA Operating Band 10
- 130 - FDD UL:1427.9-1452.9; DL:1475.9-1500.9; E-UTRA Operating Band 11
- 131 - FDD UL:698-716; DL:728-746; E-UTRA Operating Band 12
- 132 - FDD UL: 777- 787; DL: 746-756; E-UTRA Operating Band 13
- 133 - FDD UL: 788- 798; DL: 758-768; E-UTRA Operating Band 14

- 134 - FDD UL: 704-716; DL: 734-746; E-UTRA Operating Band 17
- 135 - TDD LTE UL: 1900-1920; DL: 1900-1920; E-UTRA Operating Band 33
- 136 - TDD LTE UL: 2010-2025; DL: 2010-2025; E-UTRA Operating Band 34
- 137 - TDD LTE UL: 1850-1910; DL: 1850-1910; E-UTRA Operating Band 35
- 138 - TDD LTE UL: 1930-1990; DL: 1930-1990; E-UTRA Operating Band 36
- 139 - TDD LTE UL: 1910-1930; DL: 1910-1930; E-UTRA Operating Band 37
- 140 - TDD LTE UL: 2570-2620; DL: 2570-2620; E-UTRA Operating Band 38
- 141 - TDD LTE UL: 1880-1920; DL: 1880-1920; E-UTRA Operating Band 39
- 142 - TDD LTE UL: 2300-2400; DL: 2300-2400; E-UTRA Operating Band 40
- 143 - FDD LTE UL: 815-830; DL: 860-875; E-UTRA Operating Band 18
- 144 - FDD LTE UL: 830-845; DL: 875-890; E-UTRA Operating Band 19
- 145 - FDD LTE UL: 832-862; DL: 791-821; E-UTRA Operating Band 20
- 146 - FDD LTE UL: 1447.9-1462.9; DL: 1495.9-1510.9; E-UTRA Operating Band 21
- 147 - FDD LTE UL: 1626.5-1660.5; DL: 1525-1559; E-UTRA Operating Band 24
- 148 - FDD LTE UL: 1850-1919.5; DL: 1930-1995; E-UTRA Operating Band 25
- 149 - TDD LTE UL: 2496-2690; DL: 2496-2690; E-UTRA Operating Band 41
- 150 - TDD LTE UL: 3400-3600; DL: 3400-3600; E-UTRA Operating Band 42
- 151 - TDD LTE UL: 3600-3800; DL: 3600-3800; E-UTRA Operating Band 43
- 200 - TD-SCDMA Band A
- 201 - TD-SCDMA Band B
- 202 - TD-SCDMA Band C
- 203 - TD-SCDMA Band D
- 204 - TD-SCDMA Band E
- 205 - TD-SCDMA Band F

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

9.9 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

9.9.1 Detailed Description

Call Control Return Reasons table.

9.9.2 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_OUTGOINGINTEXTTOHOME - 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

9.10 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

9.10.1 Detailed Description

Wireless Data Service Call End Reasons.

9.10.2 Call end reason codes (Code - Reason)

9.10.2.1 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

9.10.2.2 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

9.10.2.3 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

9.10.2.4 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

9.10.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

9.10.2.6 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

9.10.2.7 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

9.10.2.8 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

9.10.2.9 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

9.10.2.10 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

9.10.2.11 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

9.10.2.12 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

9.11 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

9.11.1 Detailed Description

Carrier Codes table.

9.11.2 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

9.12 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

Macros

- `#define __GOBI_API_CODING_SCHEME_H__`

9.12.1 Detailed Description

Data Coding Scheme.

9.12.2 Call Control Result Reasons (Value - Name - Description)

9.12.2.1 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

9.12.3 Coding Group Bits 7..4(0001)

9.12.3.1 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.

9.12.4 Coding Group Bits 7..4(0010)

9.12.4.1 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

9.12.5 Coding Group Bits 7..4(0011)

9.12.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

9.12.6 Coding Group Bits 7..4(01xx)

9.12.6.1 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

9.12.7 Coding Group Bits 7..4(1001)

9.12.7.1 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

9.12.8 Coding Group Bits 7..4(1010..1101)

9.12.8.1 Reserved coding groups

9.12.9 Coding Group Bits 7..4(1110)

9.12.9.1 Defined by the WAP Forum

9.12.10 Coding Group Bits 7..4 (1111)

9.12.10.1 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

9.12.11 Macro Definition Documentation

9.12.11.1 #define __GOBI_API_CODING_SCHEME_H__

9.13 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

9.13.1 Detailed Description

Position Determination Service API GPS Capability Codes.

9.13.2 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

9.14 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

9.14.1 Detailed Description

Device Management Service API Power Modes table.

9.14.2 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

9.15 qaGobiApiTableRadioInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

9.15.1 Detailed Description

Network Access Service API Radio Interfaces table.

9.15.2 Radio interface

9.15.2.1 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

9.16 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

9.16.1 Detailed Description

Region Codes table.

9.16.2 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

9.17 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

9.17.1 Detailed Description

Voice Service Options.

9.17.2 Service Option codes (Code - Reason)

9.17.2.1 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

9.18 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

9.18.1 Detailed Description

Voice Supplementary Service Information Classes.

9.18.2 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

9.19 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

9.19.1 Detailed Description

Swi Audio related tables.

9.19.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

9.19.3 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

9.20 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

9.20.1 Detailed Description

Update Complete Status table.

9.20.2 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

9.21 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

9.21.1 Detailed Description

Voice Service Call and supplementary services end reasons.

9.21.2 Voice Call and supplementary services end reason codes (Code - Reason)

9.21.2.1 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

9.21.2.2 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

9.21.2.3 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

9.21.2.4 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

9.21.2.5 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

9.21.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

9.21.2.7 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

9.21.2.8 reject causes

- 237 - Service option is not supported

9.21.2.9 IP end reasons

- 300 - Received SIP 400 bad request;waiting for INVITE response
- 301 - Received SIP 400 bad request;waiting for INVITE response
- 302 - Received SIP 404 not found; call failed; called party does not exist
- 303 - Received SIP 415 unsupported media type; call failed; called party does not support media
- 304 - Received SIP 480 temporarily unavailable; call failed; called party is not in the LTE area
- 305 - No network response; call failed
- 306 - No network response; unable to put call on hold
- 307 - Moved to eHRPD; call failed or dropped; not in the LTE area
- 308 - Upgrade/downgrade rejected (200 OK with the current call SDP)
- 309 - Received 403 call forbidden; waiting for INVITE response
- 310 - Generic timeout; did not receive a response from the server or other end
- 311 - Reported on the MO side for generic internal software errors; user can try again if the call still exists
- 312 - Reported on the MT side if the upgrade timer has been cancelled or cannot complete the request for some reason after notifying the user of a re-invite request
- 313 - Call origination is rejected due to a Service-Specific Access Control (SSAC) barring
- 314 - Phone was put in thermal emergency
- 315 - 1XCSFB call ended because of a soft failure
- 316 - 1XCSFB call ended because of a hard failure

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

9.22 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 }

```

9.22.1 Enumeration Type Documentation

9.22.1.1 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

9.22.1.2 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**

9.23 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)

9.23.1 Macro Definition Documentation

9.23.1.1 `#define LIBPACK_MAX_QOS_FILTERS 25`

9.23.1.2 `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`

9.23.1.3 `#define LIBPACK_MAX_QOS_FLOWS 8`

9.23.2 Function Documentation

9.23.2.1 `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

9.23.2.2 `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

9.23.2.3 `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

9.23.2.4 `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

9.23.2.5 `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

9.23.2.6 `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

9.23.2.7 `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

9.23.2.8 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

9.23.2.9 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

9.23.2.10 `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

9.23.2.11 `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

9.23.2.12 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

9.24 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)

9.24.1 Macro Definition Documentation

9.24.1.1 `#define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255`

9.24.1.2 `#define MAX_MS_TRANSFER_ROUTE_MSG 256`

9.24.1.3 `#define MAX_MSC_ADDRESS_SIZE 256`

9.24.1.4 `#define MAX_MSE_TWS_MSG 1254`

9.24.1.5 `#define MAX_SMS_LIST_SIZE 255`

9.24.1.6 `#define MAX_SMS_MESSAGE_SIZE 2048`

9.24.2 Typedef Documentation

9.24.2.1 `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

9.24.2.2 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

9.24.2.3 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

9.24.2.4 typedef struct **sSMSMessageMode** **sSMSMessageModeInfo**

Parameters

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

9.24.2.5 typedef struct **sMSMTMessage** **sMSMTMessageInfo**

Parameters

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

9.24.2.6 typedef struct **sMSOnIMS** **sMSOnIMSInfo**

Parameters

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

9.24.2.7 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

9.24.3 Enumeration Type Documentation

9.24.3.1 enum eqmiCbkJSetStatus

Enumerator

LIBPACK_QMI_CBK_PARAM_RESET
LIBPACK_QMI_CBK_PARAM_SET
LIBPACK_QMI_CBK_PARAM_NOCHANGE

9.24.4 Function Documentation

9.24.4.1 `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

9.24.4.2 `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

9.24.4.3 int pack_sms_SLQSDeleteSMS (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_sms_SLQSDeleteSMS_t * *reqParam*)

delete sms pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.24.4.4 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

9.24.4.5 `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

9.24.4.6 `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

9.24.4.7 `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

9.24.4.8 `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

9.24.4.9 `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

9.24.4.10 `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

9.24.4.11 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

9.24.4.12 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

9.24.4.13 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

9.24.4.14 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

9.25 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](#)

9.25.1 Detailed Description

SWI data types.

9.25.2 Macro Definition Documentation

9.25.2.1 `#define QMI_NO_LTE_FW_SUPPORT 0`

9.25.2.2 `#define QMI_TLV_PLACEHOLDER 0x8F`

9.25.2.3 `#define SWI_API`

9.25.2.4 `#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.

9.25.3 Typedef Documentation

9.25.3.1 `typedef int BOOL`

9.25.3.2 `typedef unsigned char BYTE`

9.25.3.3 `typedef char CHAR`

9.25.3.4 `typedef float FLOAT`

9.25.3.5 `typedef signed int INT32`

9.25.3.6 `typedef signed char INT8`

9.25.3.7 `typedef const char* LPCSTR`

9.25.3.8 `typedef signed short SHORT`

9.25.3.9 `typedef unsigned long ULONG`

9.25.3.10 `typedef unsigned long long ULONGLONG`

9.25.3.11 `typedef unsigned short USHORT`

9.25.3.12 `typedef unsigned short WORD`

9.26 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)

9.26.1 Function Documentation

9.26.1.1 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

9.26.1.2 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

9.26.1.3 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

9.26.1.4 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

9.27 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)

9.27.1 Macro Definition Documentation

9.27.1.1 `#define LIBPACK_MAX_SWIOMA_STR_LEN 255`

9.27.2 Function Documentation

9.27.2.1 `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

9.27.2.2 `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

9.27.2.3 `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

9.27.2.4 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

9.27.2.5 `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

9.27.2.6 `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

9.27.2.7 `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

9.27.2.8 `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

9.27.2.9 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

9.27.2.10 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

9.27.2.11 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

9.27.2.12 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

9.27.2.13 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

9.27.2.14 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

9.27.2.15 `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

9.28 SWIWWANCMAPI.h File Reference

9.29 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](#)
- struct [pack_uim_SLQSUIMPowerUp_t](#)
- struct [pack_uim_SLQSUIMPowerDown_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_SLQSUIEventRegister](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIEventRegister_t](#) *pOutput)
- int [unpack_uim_SLQSUISetStatusChangeCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUISetStatusChangeCallBack_ind_t](#) *pOutput)
- int [pack_uim_SLQSUIGetSlotsStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_uim_SLQSUIGetSlotsStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SLQSUIGetSlotsStatus_t](#) *pOutput)
- int [pack_uim_SLQSUISwitchSlot](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUISwitchSlot_t](#) *reqArg)
- int [unpack_uim_SLQSUISwitchSlot](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_uim_SetUimSlotStatusChangeCallBack_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_uim_SetUimSlotStatusChangeCallBack_ind_t](#) *pOutput)
- int [pack_uim_SLQSUIPowerUp](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIPowerUp_t](#) *reqArg)
- int [unpack_uim_SLQSUIPowerUp](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_uim_SLQSUIPowerDown](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_uim_SLQSUIPowerDown_t](#) *reqArg)
- int [unpack_uim_SLQSUIPowerDown](#) (uint8_t *pResp, uint16_t respLen)

9.29.1 Macro Definition Documentation

9.29.1.1 `#define MAX_DESCRIPTION_LENGTH 255`

9.29.1.2 `#define MAX_ICCID_LENGTH 255`

9.29.1.3 `#define MAX_NO_OF_APPLICATIONS 10`

9.29.1.4 `#define MAX_NO_OF_SLOTS 5`

9.29.1.5 `#define MAX_SLOTS_STATUS 255`

9.29.1.6 `#define UIM_MAX_DESCRIPTION_LENGTH 255`

9.29.1.7 `#define UIM_MAX_NO_OF_APPLICATIONS 10`

9.29.1.8 `#define UIM_MAX_NO_OF_SLOTS 5`

9.29.1.9 `#define UIM_UINT8_MAX_STRING_SZ 255`

9.29.2 Function Documentation

9.29.2.1 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

9.29.2.2 `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

9.29.2.3 `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

9.29.2.4 `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

9.29.2.5 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

9.29.2.6 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

9.29.2.7 int pack_uim_SLQSUIPowerDown (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIPowerDown_t * reqArg)

Powers down the card pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.29.2.8 int pack_uim_SLQSUIMPowerUp (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_uim_SLQSUIMPowerUp_t * *reqArg*)

Powers up the card pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.29.2.9 int pack_uim_SLQSUIMSwitchSlot (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_uim_SLQSUIMSwitchSlot_t * *reqArg*)

switch slot pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.29.2.10 `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

9.29.2.11 `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

9.29.2.12 `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

9.29.2.13 `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

9.29.2.14 `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

9.29.2.15 `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

9.29.2.16 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

9.29.2.17 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

9.29.2.18 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

9.29.2.19 int unpack_uim_SLQSUIMPowerDown (uint8_t * *pResp*, uint16_t *respLen*)

Powers down the card 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

9.29.2.20 int unpack_uim_SLQSUIMPowerUp (uint8_t * *pResp*, uint16_t *respLen*)

Powers up the card 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

9.29.2.21 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_SLQSUIEventRegister to subscribe

9.29.2.22 int unpack_uim_SLQSUISwitchSlot (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

9.29.2.23 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

9.29.2.24 `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

9.30 wds.h File Reference

Data Structures

- struct [LibPackQosClassID](#)
- struct [LibPackTFTIDParams](#)
- struct [LibPackGPRSRequestedQoS](#)
- struct [LibPackUMTSQoS](#)
- struct [LibPackUMTSReqQoSSigInd](#)
- struct [pack_wds_SLQSStartDataSession_t](#)
- struct [unpack_wds_SLQSStartDataSession_t](#)
- struct [unpack_wds_SLQSSetPacketSrvStatusCallback_t](#)
- struct [pack_wds_SLQSStopDataSession_t](#)
- struct [wds_ProfileIdentifier](#)
- struct [wds_GPRSQoS](#)
- struct [wds_PCSCFIPv4ServerAddressList](#)
- struct [wds_PCSCFFQDNAddress](#)
- struct [wds_PCSCFFQDNAddressList](#)
- struct [wds_Domain](#)
- struct [wds_DomainNameList](#)
- struct [wds_IPV6AddressInfo](#)
- struct [wds_IPV6GWAddressInfo](#)
- struct [unpack_wds_SLQSGetRuntimeSettings_t](#)
- struct [wds_currNetworkInfo](#)
- struct [unpack_wds_SLQSSetWdsEventCallback_ind_t](#)
- struct [pack_wds_SLQSSetWdsEventCallback_t](#)
- struct [pack_wds_SLQSGetRuntimeSettings_t](#)
- struct [wds_UMTSMinQoS](#)
- struct [LibPackprofile_3GPP](#)
- struct [LibPackprofile_3GPP2](#)
- union [wds_profileInfo](#)
- struct [pack_wds_SLQSCreateProfile_t](#)
- struct [PackCreateProfileOut](#)
- struct [unpack_wds_SLQSCreateProfile_t](#)
- struct [pack_wds_SLQSModifyProfile_t](#)
- struct [unpack_wds_SLQSModifyProfile_t](#)

- struct [pack_wds_SLQSGetProfileSettings_t](#)
- struct [LibpackProfile3GPP](#)
- struct [LibpackProfile3GPP2](#)
- union [unpackWdsProfileParam](#)
- struct [UnPackGetProfileSettingOut](#)
- struct [unpack_wds_SLQSGetProfileSettings_t](#)
- struct [unpack_wds_GetSessionState_t](#)
- struct [pack_wds_GetDefaultProfile_t](#)
- struct [unpack_wds_GetDefaultProfile_t](#)
- struct [unpack_wds_GetConnectionRate_t](#)
- struct [pack_wds_GetPacketStatus_t](#)
- struct [unpack_wds_GetPacketStatus_t](#)
- struct [unpack_wds_GetSessionDuration_t](#)
- struct [pack_wds_GetSessionDuration_t](#)
- struct [unpack_wds_GetDormancyState_t](#)
- struct [pack_wds_GetDormancyState_t](#)
- struct [pack_wds_SLQSDeleteProfile_t](#)
- struct [unpack_wds_SLQSDeleteProfile_t](#)
- struct [pack_wds_SetDefaultProfile_t](#)
- struct [unpack_wds_SLQSGet3GPPConfigItem_t](#)
- struct [pack_wds_SLQSSet3GPPConfigItem_t](#)
- struct [unpack_wds_GetMobileIP_t](#)
- struct [pack_wds_GetMobileIP_t](#)
- struct [pack_wds_GetMobileIPProfile_t](#)
- struct [unpack_wds_GetMobileIPProfile_t](#)
- struct [currNetworkInfo](#)
- struct [unpack_wds_SLQSGetCurrDataSystemStat_t](#)
- struct [pack_wds_SLQSGetCurrDataSystemStat_t](#)
- struct [unpack_wds_GetLastMobileIPError_t](#)
- struct [pack_wds_GetLastMobileIPError_t](#)
- struct [rmTrasnferStaticsReq](#)
- struct [pack_wds_RMSetTransferStatistics_t](#)
- struct [unpack_wds_RMSetTransferStatistics_t](#)
- struct [pack_wds_SetMobileIPProfile_t](#)
- struct [unpack_wds_SetMobileIPProfile_t](#)
- struct [pack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#)
- struct [ipv6AddressInfo](#)
- struct [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#)
- struct [transferStatInd](#)
- struct [pack_wds_SLQSGetDUNCallInfo_t](#)
- struct [connectionStatus](#)
- struct [dunchannelRate](#)
- struct [unpack_wds_SLQSGetDUNCallInfo_t](#)
- struct [qmiWDSDataBearerTechnology](#)
- struct [unpack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSGetDataBearerTechnology_t](#)
- struct [pack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [unpack_wds_SLQSSetIPFamilyPreference_t](#)
- struct [pack_wds_SetDefaultProfileNum_t](#)
- struct [pack_wds_GetDefaultProfileNum_t](#)
- struct [unpack_wds_GetDefaultProfileNum_t](#)
- struct [wdsDhcpv4ProfileId](#)
- struct [wdsDhcpv4HwConfig](#)
- struct [wdsDhcpv4Option](#)
- struct [wdsDhcpv4OptionList](#)

- struct [pack_wds_SLQSSGetDHCPv4ClientConfig_t](#)
- struct [unpack_wds_SLQSSGetDHCPv4ClientConfig_t](#)
- struct [pack_wds_GetPacketStatistics_t](#)
- struct [unpack_wds_GetPacketStatistics_t](#)
- struct [unpack_wds_GetByteTotals_t](#)
- struct [unpack_wds_SLQSGetCurrentChannelRate_t](#)
- struct [unpack_wds_SLQSSGetLoopback_t](#)
- struct [pack_wds_SLQSSSetLoopback_t](#)

Macros

- `#define IPV6_ADDRESS_ARRAY_SIZE 8`
- `#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24`
- `#define PACK_WDS_IPV4 4`
- `#define PACK_WDS_IPV6 6`
- `#define BYT_STAT_STAT_MASK 0X000000C0`

Typedefs

- typedef union [unpackWdsProfileParam](#) [UnpackQmiProfileInfo](#)

Functions

- int [pack_wds_SLQSStartDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSStartDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSStartDataSession](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSStartDataSession_t](#) *pOutput)
- int [unpack_wds_SLQSSetPacketSrvStatusCallback](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetPacketSrvStatusCallback_t](#) *pOutput)
- int [pack_wds_SLQSStopDataSession](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSStopDataSession_t](#) *reqArg)
- int [unpack_wds_SLQSStopDataSession](#) (uint8_t *pResp, uint16_t respLen)
- int [unpack_wds_SLQSGetRuntimeSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetRuntimeSettings_t](#) *pOutput)
- int [unpack_wds_SLQSSetWdsEventCallback_ind](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetWdsEventCallback_ind_t](#) *pOutput)
- int [unpack_wds_SLQSSetWdsEventCallback](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_SLQSSetWdsEventCallback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSetWdsEventCallback_t](#) *reqArg)
- int [pack_wds_SLQSGetRuntimeSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetRuntimeSettings_t](#) *reqArg)
- int [pack_wds_SLQSCreateProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSCreateProfile_t](#) *reqArg)
- int [unpack_wds_SLQSCreateProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSCreateProfile_t](#) *pOutput)
- int [pack_wds_SLQSModifyProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSModifyProfile_t](#) *reqArg)
- int [unpack_wds_SLQSModifyProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSModifyProfile_t](#) *pOutput)
- int [pack_wds_SLQSGetProfileSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetProfileSettings_t](#) *reqArg)
- int [unpack_wds_SLQSGetProfileSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetProfileSettings_t](#) *pOutput)
- int [pack_wds_GetSessionState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int [unpack_wds_GetSessionState](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetSessionState_t](#) *pOutput)
- int [pack_wds_GetDefaultProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDefaultProfile_t](#) *reqParam)
- int [unpack_wds_GetDefaultProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDefaultProfile_t](#) *pOutput)
- int [pack_wds_GetConnectionRate](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_wds_GetConnectionRate](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetConnectionRate_t](#) *pOutput)
- int [pack_wds_GetPacketStatus](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetPacketStatus_t](#) *reqParam)
- int [unpack_wds_GetPacketStatus](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetPacketStatus_t](#) *pOutput)
- int [pack_wds_GetSessionDuration](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetSessionDuration_t](#) *reqParam)
- int [unpack_wds_GetSessionDuration](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetSessionDuration_t](#) *pOutput)
- int [pack_wds_GetDormancyState](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDormancyState_t](#) *reqParam)
- int [unpack_wds_GetDormancyState](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDormancyState_t](#) *pOutput)
- int [pack_wds_SLQSDeleteProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSDeleteProfile_t](#) *reqParam)
- int [unpack_wds_SLQSDeleteProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSDeleteProfile_t](#) *pOutput)
- int [pack_wds_SetDefaultProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetDefaultProfile_t](#) *reqParam)
- int [unpack_wds_SetDefaultProfile](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_SLQSGet3GPPConfigItem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_wds_SLQSGet3GPPConfigItem](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGet3GPPConfigItem_t](#) *pOutput)
- int [pack_wds_SLQSSet3GPPConfigItem](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSet3GPPConfigItem_t](#) *reqParam)
- int [unpack_wds_SLQSSet3GPPConfigItem](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_GetMobileIP](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetMobileIP_t](#) *pReqParam)
- int [unpack_wds_GetMobileIP](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetMobileIP_t](#) *pOutput)
- int [pack_wds_GetMobileIPProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetMobileIPProfile_t](#) *reqParam)
- int [unpack_wds_GetMobileIPProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetMobileIPProfile_t](#) *pOutput)
- int [pack_wds_SLQSGetCurrDataSystemStat](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetCurrDataSystemStat_t](#) *pReqParam)
- int [unpack_wds_SLQSGetCurrDataSystemStat](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetCurrDataSystemStat_t](#) *pOutput)
- int [pack_wds_GetLastMobileIPError](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetLastMobileIPError_t](#) *pReqParam)
- int [unpack_wds_GetLastMobileIPError](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetLastMobileIPError_t](#) *pOutput)
- int [pack_wds_RMSetTransferStatistics](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_RMSetTransferStatistics_t](#) *reqParam)
- int [unpack_wds_RMSetTransferStatistics](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_RMSetTransferStatistics_t](#) *pOutput)
- int [pack_wds_SetMobileIPProfile](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetMobileIPProfile_t](#) *reqParam)

- int [unpack_wds_SetMobileIPProfile](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SetMobileIPProfile_t](#) *pOutput)
- int [pack_wds_SLQSWdsSwiPDPRuntimeSettings](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#) *reqParam)
- int [unpack_wds_SLQSWdsSwiPDPRuntimeSettings](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t](#) *pOutput)
- int [pack_wds_SLQSGetDUNCallInfo](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetDUNCallInfo_t](#) *reqParam)
- int [unpack_wds_SLQSGetDUNCallInfo](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetDUNCallInfo_t](#) *pOutput)
- int [pack_wds_SLQSGetDataBearerTechnology](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSGetDataBearerTechnology_t](#) *pReqParam)
- int [unpack_wds_SLQSGetDataBearerTechnology](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetDataBearerTechnology_t](#) *pOutput)
- int [pack_wds_SLQSSetIPFamilyPreference](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSetIPFamilyPreference_t](#) *pReqParam)
- int [unpack_wds_SLQSSetIPFamilyPreference](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSetIPFamilyPreference_t](#) *pOutput)
- int [pack_wds_SetDefaultProfileNum](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SetDefaultProfileNum_t](#) *pReqParam)
- int [unpack_wds_SetDefaultProfileNum](#) (uint8_t *pResp, uint16_t respLen)
- int [pack_wds_GetDefaultProfileNum](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetDefaultProfileNum_t](#) *pReqParam)
- int [unpack_wds_GetDefaultProfileNum](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetDefaultProfileNum_t](#) *pOutput)
- int [pack_wds_SLQSSGetDHCPv4ClientConfig](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSGetDHCPv4ClientConfig_t](#) *pReq)
- int [unpack_wds_SLQSSGetDHCPv4ClientConfig](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSGetDHCPv4ClientConfig_t](#) *pOutput)
- int [pack_wds_GetPacketStatistics](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_GetPacketStatistics_t](#) *pReq)
- int [unpack_wds_GetPacketStatistics](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetPacketStatistics_t](#) *pOutput)
- int [pack_wds_GetByteTotals](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_wds_GetByteTotals](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_GetByteTotals_t](#) *pOutput)
- int [pack_wds_SLQSGetCurrentChannelRate](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_wds_SLQSGetCurrentChannelRate](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSGetCurrentChannelRate_t](#) *pOutput)
- int [pack_wds_SLQSSGetLoopback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int [unpack_wds_SLQSSGetLoopback](#) (uint8_t *pResp, uint16_t respLen, [unpack_wds_SLQSSGetLoopback_t](#) *pOutput)
- int [pack_wds_SLQSSSetLoopback](#) ([pack_qmi_t](#) *pCtx, uint8_t *pReqBuf, uint16_t *pLen, [pack_wds_SLQSSSetLoopback_t](#) *reqArg)
- int [unpack_wds_SLQSSSetLoopback](#) (uint8_t *pResp, uint16_t respLen)

9.30.1 Macro Definition Documentation

9.30.1.1 `#define BYT_STAT_STAT_MASK 0X000000C0`

9.30.1.2 `#define IPV6_ADDRESS_ARRAY_SIZE 8`

9.30.1.3 `#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24`

9.30.1.4 `#define PACK_WDS_IPV4 4`

9.30.1.5 `#define PACK_WDS_IPV6 6`

9.30.2 Typedef Documentation

9.30.2.1 `typedef union unpackWdsProfileParam UnpackQmiProfileInfo`

9.30.3 Function Documentation

9.30.3.1 `int pack_wds_GetByteTotals (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get Rx/Tx byte counts since the start of the last packet data session pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.2 `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

9.30.3.3 `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

9.30.3.4 `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

9.30.3.5 `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

9.30.3.6 `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

9.30.3.7 `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

9.30.3.8 `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

9.30.3.9 `int pack_wds_GetPacketStatistics (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetPacketStatistics_t * pReq)`

gets current packet transfer counter values pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.10 `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

9.30.3.11 `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

9.30.3.12 `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

```
9.30.3.13 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

```
9.30.3.14 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

```
9.30.3.15 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

9.30.3.16 int pack_wds_SetMobileIPProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SetMobileIPProfile_t * reqParam)

set mobile ip profile pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

Note

PDN Specific: Yes

9.30.3.17 int pack_wds_SLQSCreateProfile (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSCreateProfile_t * reqArg)

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

Note

PDN Specific: Yes

9.30.3.18 `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

9.30.3.19 `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

9.30.3.20 `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

9.30.3.21 `int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen)`

get current Tx/Rx channel bitrate of the current packet data pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.22 `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

9.30.3.23 `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

9.30.3.24 `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

9.30.3.25 `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

9.30.3.26 `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

9.30.3.27 `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

9.30.3.28 `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

9.30.3.29 `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

9.30.3.30 `int pack_wds_SLQSSetDHCPv4ClientConfig (pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSetDHCPv4ClientConfig_t * pReq)`

get DHCPv4 Client Config pack

Parameters

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

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.31 int pack_wds_SLQSSGetLoopback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*)

get the value of loopback mode and multiplier pack

Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.32 int pack_wds_SLQSSSetLoopback (pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*,
pack_wds_SLQSSSetLoopback_t * *reqArg*)

Enable/disable Data Loopback Mode and set the value of loopback multiplier pack

Parameters

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

9.30.3.33 `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

9.30.3.34 `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

9.30.3.35 `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

9.30.3.36 `int unpack_wds_GetByteTotals (uint8_t * pResp, uint16_t respLen, unpack_wds_GetByteTotals_t * pOutput)`

get Rx/Tx byte counts since the start of the last packet data session unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.37 `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

9.30.3.38 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

9.30.3.39 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

9.30.3.40 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

9.30.3.41 `int unpack_wds_GetLastMobileIPError (uint8_t * pResp, uint16_t respLen, unpack_wds_GetLastMobileIP-Error_t * pOutput)`

get current data system unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.42 `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

9.30.3.43 `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

9.30.3.44 int unpack_wds_GetPacketStatistics (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatistics_t * *pOutput*)

gets current packet transfer counter values unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.45 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

9.30.3.46 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

9.30.3.47 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

9.30.3.48 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

9.30.3.49 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

9.30.3.50 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

9.30.3.51 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

9.30.3.52 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

9.30.3.53 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

9.30.3.54 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

9.30.3.55 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

9.30.3.56 `int unpack_wds_SLQSGetCurrentChannelRate (uint8_t * pResp, uint16_t respLen,
unpack_wds_SLQSGetCurrentChannelRate_t * pOutput)`

get current Tx/Rx channel bitrate of the current packet data unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.57 `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

9.30.3.58 `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

9.30.3.59 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

9.30.3.60 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

9.30.3.61 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

9.30.3.62 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

9.30.3.63 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

9.30.3.64 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

9.30.3.65 `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

9.30.3.66 `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

9.30.3.67 `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

9.30.3.68 `int unpack_wds_SLQSSGetLoopback (uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSSGetLoopback_t * pOutput)`

get the value of loopback mode and multiplier unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.69 `int unpack_wds_SLQSSSetLoopback (uint8_t * pResp, uint16_t respLen)`

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.70 `int unpack_wds_SLQSStartDataSession (uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSStartDataSession_t * pOutput)`

start data session unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

9.30.3.71 int unpack_wds_SLQSSStopDataSession (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

9.30.3.72 int unpack_wds_SLQSWdsSwiPDPRuntimeSettings (uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t * *pOutput*)

get current data system unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

See Also

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

Index

- [_3gppRelease](#)
 - [unpack_wds_SLQSGet3GPPConfigItem_t, 386](#)
 - [_libSDP_FirmwareInfo_, 23](#)
 - [szCarrier_str, 23](#)
 - [szCarrierPriversion_str, 24](#)
 - [szFwversion_str, 24](#)
 - [szModelid_str, 24](#)
 - [szPackageid_str, 24](#)
 - [szSku_str, 24](#)
- [AAASPI](#)
 - [unpack_wds_GetMobileIPProfile_t, 380](#)
- [AAASState](#)
 - [unpack_wds_GetMobileIPProfile_t, 380](#)
- [AMSSString](#)
 - [unpack_dms_GetFirmwareRevision_t, 269](#)
 - [unpack_dms_GetFirmwareRevisions_t, 269](#)
- [APNName](#)
 - [unpack_wds_SLQSGetRuntimeSettings_t, 391](#)
- [accolc](#)
 - [pack_nas_SetACCOLC_t, 187](#)
- [ackIndicator](#)
 - [sMSTransferRouteMTMessage, 246](#)
- [acqOrdeLen](#)
 - [nas_acqOrderPref, 79](#)
- [activationStatus](#)
 - [dms_ActivationStatusTlv, 30](#)
- [ActivationStatusTlv](#)
 - [unpack_dms_SetEventReport_ind_t, 276](#)
- [activeBandClass](#)
 - [nas_RFInfoTlv, 136](#)
 - [RFBandInfoElements, 238](#)
- [activeChannel](#)
 - [nas_RFInfoTlv, 136](#)
 - [RFBandInfoElements, 238](#)
- [ActiveTechPref](#)
 - [unpack_nas_GetNetworkPreference_t, 308](#)
- [addr](#)
 - [unpack_qos_IPv4Addr_t, 340](#)
 - [unpack_qos_IPv6Addr_t, 340](#)
- [address](#)
 - [unpack_wds_GetMobileIPProfile_t, 380](#)
- [aid](#)
 - [uim_sessionInformation, 258](#)
 - [uim_UIMSessionInformation, 261](#)
- [aidLength](#)
 - [appStats, 26](#)
 - [uim_appStatus, 250](#)
 - [uim_sessionInformation, 258](#)
 - [uim_UIMSessionInformation, 261](#)
- [aidVal](#)
 - [appStats, 26](#)
 - [uim_appStatus, 250](#)
- [aidingIndicatorMask](#)
 - [loc_sensorDataUsage, 75](#)
- [alertmsg](#)
 - [unpack_omaDmConfigTlv_t, 336](#)
- [alertmsglength](#)
 - [unpack_omaDmConfigTlv_t, 336](#)
- [ambr_dl](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 345](#)
- [ambr_dl_ext](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 345](#)
- [ambr_dl_ext2](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 346](#)
- [ambr_ul](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 346](#)
- [ambr_ul_ext](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 346](#)
- [ambr_ul_ext2](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 346](#)
- [amssSize](#)
 - [unpack_dms_GetFirmwareRevision_t, 269](#)
 - [unpack_dms_GetFirmwareRevisions_t, 269](#)
- [apdoxypages.c, 411](#)
- [apnId](#)
 - [pack_qos_SLQSQosSwiReadApnExtraParams_t, 204](#)
 - [pack_qos_SLQSQosSwiReadDataStats_t, 204](#)
 - [unpack_qos_SLQSQosSwiReadApnExtraParams_t, 346](#)
 - [unpack_qos_SLQSQosSwiReadDataStats_t, 347](#)
- [apnName](#)
 - [unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 397](#)
- [apnname](#)
 - [unpack_wds_GetDefaultProfile_t, 378](#)
- [apnsize](#)
 - [unpack_wds_GetDefaultProfile_t, 378](#)
- [appNameLength](#)
 - [loc_LocApplicationInfo, 74](#)
- [appProviderLength](#)
 - [loc_LocApplicationInfo, 74](#)

- appState
 - appStats, 26
 - uim_appStatus, 250
- appStats, 24
 - aidLength, 26
 - aidVal, 26
 - appState, 26
 - appType, 26
 - persoFeature, 26
 - persoRetries, 26
 - persoState, 26
 - persoUnblockRetries, 26
 - pin1Retries, 27
 - pin1State, 27
 - pin2Retries, 27
 - pin2State, 27
 - puk1Retries, 27
 - puk2Retries, 27
 - univPin, 27
- AppStatus
 - slotInf, 241
 - uim_slotInfo, 260
- appType
 - appStats, 26
 - uim_appStatus, 250
- appVersionLength
 - loc_LocApplicationInfo, 74
- appVersionValid
 - loc_LocApplicationInfo, 74
- Application
 - unpack_nas_GetCDMANetworkParameters_t, 306
- appversion_str
 - unpack_dms_GetFirmwareInfo_t, 268
- arfcn
 - nas_GERANInfo, 98
 - nas_gsmCellInfo, 100
- auth
 - unpack_wds_GetDefaultProfile_t, 378
- Authentication
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- authentication
 - pack_wds_SetDefaultProfile_t, 224
- Autosdm
 - unpack_swima_SLQSOMADMGetSettings_t, 369
- avgPeriod
 - nas_LTESigRptConfig, 120
- bEnable
 - pack_nas_SLQSSetSignalStrengthsCallback_t, 198
- bForceDownload
 - pack_fms_SetImagesPreference_t, 180
- bICCID
 - slot_t, 239
- bICCIDLength
 - slot_t, 239
- bLogicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, 219
 - slot_t, 239
- bNumberOfPhySlots
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 373
- BOOL
 - SwiDataTypes.h, 563
- BPTlv
 - NASQmiCbkNasSystemSelPrefInd, 171
- bResetStatistics
 - rmTrasferStaticsReq, 238
- BYT_STAT_STAT_MASK
 - wds.h, 589
- BYTE
 - SwiDataTypes.h, 563
- band
 - nas_LTEInfo, 113
- band1900
 - nas_gsmCellInfo, 100
- band_pref
 - NASBandPreferenceTlv, 163
- BandCapability
 - unpack_dms_GetBandCapability_t, 263
- bandCapability
 - unpack_dms_SLQSGetBandCapability_t, 283
- bandwidth
 - nas_LTEInfo, 113
- baseId
 - nas_CDMAInfo, 83
 - nas_CDMA SysInfo, 87
- baseLat
 - nas_CDMAInfo, 83
 - nas_CDMA SysInfo, 87
- baseLong
 - nas_CDMAInfo, 83
 - nas_CDMA SysInfo, 87
- BasestationID
 - unpack_nas_SLQSGetServingSystem_t, 317
- BasestationLatitude
 - unpack_nas_SLQSGetServingSystem_t, 318
- BasestationLongitude
 - unpack_nas_SLQSGetServingSystem_t, 318
- BearerID
 - unpack_qos_QosFlowInfo_t, 343
- bearerID
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 393
- bearerId
 - unpack_QosFlowStat_t, 359
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 397
- bootSize
 - unpack_dms_GetFirmwareRevisions_t, 269
- BootString
 - unpack_dms_GetFirmwareRevisions_t, 269
- bootversion_str
 - unpack_dms_GetFirmwareInfo_t, 268
- Broadcast
 - unpack_nas_GetCDMANetworkParameters_t, 306
- bsInfoValid

- nas_CDMA SysInfo, 87
- bsPRev
 - nas_CDMA SysInfo, 87
- bsPRevValid
 - nas_CDMA SysInfo, 87
- bsic
 - nas_GERANInfo, 98
- bsicId
 - nas_gsmCellInfo, 100
- bucketSz
 - unpack_qos_tokenBucket_t, 358
- buildID
 - FMSImageIdElement, 36
 - image_info_t, 38
- buildIDLen
 - image_info_t, 38
- buildIDLength
 - FMSImageIdElement, 36
- buildId
 - FMSImageElement, 35
- buildIdLength
 - FMSImageElement, 35
- ByteLoopbackMode
 - unpack_wds_SLQSSGetLoopback_t, 396
- ByteLoopbackMultiplier
 - unpack_wds_SLQSSGetLoopback_t, 396
- CDMA_P_Rev
 - unpack_nas_SLQSSGetServingSystem_t, 318
- CDMAECIOThreshListLen
 - nas_CDMAECIOThresh, 82
- CDMARSSIThreshListLen
 - nas_CDMARSSIThresh, 84
- CDMA SSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 328
- CDMA SystemInfoExt
 - unpack_nas_SLQSSGetServingSystem_t, 318
- CHAR
 - SwiDataTypes.h, 563
- CQIValueCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, 332
- CQIValueCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, 332
- CSDomain
 - unpack_nas_GetServingNetwork_t, 309
- CallBarStatus
 - unpack_nas_SLQSSGetServingSystem_t, 318
- callDuration
 - unpack_wds_GetSessionDuration_t, 384
- callEndReason
 - unpack_wds_SLQSSGetDUNCallInfo_t, 389
- cardState
 - slotInf, 241
 - uim_slotInfo, 260
- carrier
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 285
- carrier_str
 - unpack_dms_GetFirmwareInfo_t, 268
- CarrierImage_t, 27
- m_FwBuildId, 28
- m_FwImageId, 28
- m_PriBuildId, 28
- m_PriImageId, 28
- m_nCarrierId, 28
- m_nFolderId, 28
- m_nStorage, 28
- ccsSupported
 - nas_CDMA SysInfo, 87
- ccsSupportedValid
 - nas_CDMA SysInfo, 87
- cdmaSSInfo, 28
 - ecio, 28
 - rssI, 28
- cdmaSysIdValid
 - nas_CDMA SysInfo, 87
- cell_resel_priority
 - nas_infoInterFreq, 110
- cellBroadcastCap
 - nas_AddSysInfo, 80
- CellID
 - unpack_nas_SLQSSGetServingSystem_t, 318
- cellID
 - nas_GERANInfo, 98
 - nas_UMTSInfo, 154
- cellId
 - nas_GSM SysInfo, 104
 - nas_LTE SysInfo, 123
 - nas_WCDMA SysInfo, 163
- cellIdValid
 - nas_gsmCellInfo, 100
 - nas_GSM SysInfo, 104
 - nas_LTE SysInfo, 123
 - nas_WCDMA SysInfo, 163
- cellInterFreqParams
 - nas_infoInterFreq, 110
- cellsTDD
 - nas_umtsLTENbrCell, 156
- CellParams
 - nas_LTEInfoIntraFreq, 116
- cellReselPriority
 - nas_lteGsmCellInfo, 111
 - nas_LTEInfoIntraFreq, 116
 - nas_lteWcdmaCellInfo, 125
- cells_len
 - nas_infoInterFreq, 110
 - nas_lteGsmCellInfo, 111
- cellsLen
 - nas_LTEInfoIntraFreq, 116
 - nas_lteWcdmaCellInfo, 125
- chaddr
 - wdsDhcpv4HwConfig, 408
- chaddrLen
 - wdsDhcpv4HwConfig, 408
- changePIN
 - pack_uim_ChangePin_t, 214
- channelRate
 - unpack_wds_SLQSSGetDUNCallInfo_t, 389

- common.h
 - eCTL, [413](#)
 - eDMS, [413](#)
 - eIND, [414](#)
 - eLOC, [413](#)
 - eLOG_DEBUG, [413](#)
 - eLOG_FATAL, [413](#)
 - eLOG_INFO, [413](#)
 - eLOG_WARN, [413](#)
 - eNAS, [413](#)
 - eQOS, [413](#)
 - eREQ, [414](#)
 - eRSP, [414](#)
 - eSMS, [413](#)
 - eSWILOC, [414](#)
 - eSWIOMA, [414](#)
 - eTIMEOUT_10_S, [414](#)
 - eTIMEOUT_20_S, [414](#)
 - eTIMEOUT_2_S, [414](#)
 - eTIMEOUT_300_S, [414](#)
 - eTIMEOUT_30_S, [414](#)
 - eTIMEOUT_5_S, [414](#)
 - eTIMEOUT_60_S, [414](#)
 - eTIMEOUT_8_S, [414](#)
 - eTIMEOUT_DEFAULT, [414](#)
 - eTMD, [414](#)
 - eUIM, [413](#)
 - eWDS, [413](#)
- common.h, [411](#)
 - eLOG_LEVEL, [413](#)
 - eQMI_SVC, [413](#)
 - eTimeout, [414](#)
 - fill_pack_ctx, [414](#)
 - fill_sdu_hdr, [414](#)
 - get_version, [414](#)
 - glog, [415](#)
 - gloglvl, [415](#)
 - helper_get_resp_ctx, [414](#)
 - helper_get_xid, [415](#)
 - helper_set_log_func, [415](#)
 - helper_set_log_lvl, [415](#)
 - libpack_GetVersion, [415](#)
 - libpack_log, [415](#)
 - logger, [413](#)
 - MINREQBKLEN, [413](#)
 - MSGID_AND_LEN, [413](#)
 - MSGID_DONT_CARE, [413](#)
 - msgtype, [414](#)
 - SDU_HDR_LEN, [413](#)
 - UNUSEDPARAM, [413](#)
 - unpack_result_code_only, [415](#)
- commonInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, [330](#)
- ConcSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, [318](#)
- conn_status
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [393](#)
- connectionStatus, [28](#)
 - MDMCallDuration, [29](#)
 - MDMConnStatus, [29](#)
 - unpack_wds_GetSessionState_t, [384](#)
 - unpack_wds_SLQSGetDUNCallInfo_t, [389](#)
- content
 - uim_readResult, [256](#)
- contentLen
 - uim_readResult, [256](#)
- contextId
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [235](#)
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- contextType
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [235](#)
- cpich_ecno
 - nas_wcdmaCellInfo, [158](#)
- cpich_rscp
 - nas_wcdmaCellInfo, [158](#)
- crashAction
 - pack_dms_SetCrashAction_t, [175](#)
- csAttachState
 - nas_servSystem, [142](#)
 - NASServingSystemInfo, [173](#)
- csBarStatus
 - nas_CallBarringSysInfo, [81](#)
 - nas_callBarStatus, [82](#)
- cur_carr_name
 - unpack_dms_GetFirmwareInfo_t, [268](#)
- cur_carr_rev
 - unpack_dms_GetFirmwareInfo_t, [268](#)
- curDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, [388](#)
- curProfile
 - pack_wds_SLQSModifyProfile_t, [230](#)
 - UnPackGetProfileSettingOut, [398](#)
- CurrChanRxRate
 - dunchannelRate, [34](#)
- CurrChanTxRate
 - dunchannelRate, [34](#)
- currDBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [394](#)
- currNWInfo
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [394](#)
- currNetworkInfo, [29](#)
 - NetworkType, [29](#)
 - RATMask, [29](#)
 - SOMask, [29](#)
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [387](#)
- current_channel_rx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, [388](#)

- current_channel_tx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 388
- currentChannelRXRate
 - unpack_wds_GetConnectionRate_t, 377
- currentChannelTXRate
 - unpack_wds_GetConnectionRate_t, 377
- currentDataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, 233
- currentNetwork
 - qmiWSDDataBearerTechnology, 238
- CurrentPLMN
 - unpack_nas_SLQSGetServingSystem_t, 318
- cust_attr
 - DMScustSettingInfo, 31
- cust_id
 - DMScustSettingInfo, 31
 - DMSgetCustomInput, 33
 - pack_dms_GetCustFeaturesV2_t, 174
 - pack_dms_SetCustFeaturesV2_t, 176
- cust_value
 - DMScustSettingInfo, 31
 - pack_dms_SetCustFeaturesV2_t, 176
- custSetting
 - DMScustSettingList, 32
- CustomSCP
 - unpack_nas_GetCDMANetworkParameters_t, 306
- dBTechAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- dBTechnology
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- DHCPRelayEnabled
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- DMS_IMGDETAILS_LEN
 - dms.h, 420
- DMS_PM_FACTORY
 - dms.h, 420
- DMS_PM_LOW
 - dms.h, 420
- DMS_PM_OFFLINE
 - dms.h, 420
- DMS_PM_ONLINE
 - dms.h, 420
- DMS_PM_RESET
 - dms.h, 420
- DMS_PM_SHUT_DOWN
 - dms.h, 420
- DMScustSettingInfo, 31
 - cust_attr, 31
 - cust_id, 31
 - cust_value, 31
 - id_length, 31
 - value_length, 31
- DMScustSettingList, 31
 - custSetting, 32
- list_type, 32
- num_instances, 32
- DMSgetCustomFeatureV2, 32
 - pCustSettingInfo, 33
 - pCustSettingList, 33
 - pGetCustomInput, 33
- DMSgetCustomInput, 33
 - cust_id, 33
 - list_type, 33
- DTMInd
 - unpack_nas_SLQSGetServingSystem_t, 318
- data
 - sMSCAddress, 242
 - sMSEtwSMessage, 243
 - sMSTransferRouteMTMessage, 246
- data_buf
 - NASOTAMessageTlv, 166
- data_len
 - NASOTAMessageTlv, 166
- dataBearer
 - pack_wds_SLQSSetWdsEventCallback_t, 233
- dataBearerMask
 - unpack_wds_SLQSGetDataBearerTechnology_t, 388
- dataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- dataCapabilities
 - nas_dataSrvCapabilities, 93
- dataCapabilitiesLen
 - nas_dataSrvCapabilities, 93
- DataCaps
 - unpack_nas_GetServingNetwork_t, 309
 - unpack_nas_GetServingNetworkCapabilities_t, 310
- dataCaps
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 311
- DataCapsLen
 - unpack_nas_GetServingNetwork_t, 309
 - unpack_nas_GetServingNetworkCapabilities_t, 310
- dataCapsSize
 - unpack_nas_SetDataCapabilitiesCallback_ind_t, 311
- DataRate
 - unpack_qos_swiQosFlow_t, 356
- dataRateMax
 - unpack_qos_dataRate_t, 339
- dataServiceCaCapability
 - unpack_dms_GetDeviceCapabilities_t, 266
- DataServiceCapability
 - unpack_dms_GetDeviceCap_t, 265
- DataSrvCapabilities
 - unpack_nas_SLQSGetServingSystem_t, 318
- dataSysStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- dataSystemStatus

- pack_wds_SLQSSetWdsEventCallback_t, 233
- Date
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 367
- DateLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 367
- day
 - nas_timeInfo, 152
 - nas_UniversalTime, 157
- dayLtSavingAdj
 - nas_timeInfo, 152
- dayOfWeek
 - nas_timeInfo, 152
 - nas_UniversalTime, 157
- daylightSavings
 - nas_qaQmi3Gpp2TimeZone, 133
- defaultPDNEnabled
 - unpack_wds_SLQSGet3GPPConfigItem_t, 386
- DefaultRoamInd
 - unpack_nas_SLQSGetServingSystem_t, 318
- delayClass
 - LibPackGPRSRequestedQoS, 40
 - wds_GPRSQoS, 402
- deliveryErrSDU
 - LibPackUMTSQoS, 67
 - wds_UMTSMInQoS, 407
- description
 - unpack_omaDmFotaTlv_t, 338
- descriptionlength
 - unpack_omaDmFotaTlv_t, 338
- Desription
 - nas_QmiNas3GppNetworkInfo, 134
- destPortRangeEnd
 - LibPackTFTIDParams, 64
- destPortRangeStart
 - LibPackTFTIDParams, 64
- DetailedSvcInfo
 - unpack_nas_SLQSGetServingSystem_t, 318
- DevCrashState
 - unpack_dms_GetCrashAction_t, 264
- DisableIMSI
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- dl_bw_value
 - nas_PhyCaAggPcellInfo, 128
 - nas_PhyCaAggScellIDIBw, 129
 - nas_PhyCaAggScellInfo, 133
 - NASPhyCaAggPcellInfo, 167
 - NASPhyCaAggScellIDIBw, 167
 - NASPhyCaAggScellInfo, 169
- dms.h, 415
 - DMS_IMGDETAILS_LEN, 420
 - DMS_PM_FACTORY, 420
 - DMS_PM_LOW, 420
 - DMS_PM_OFFLINE, 420
 - DMS_PM_ONLINE, 420
 - DMS_PM_RESET, 420
 - DMS_PM_SHUT_DOWN, 420
 - MAX_BUILD_ID_LEN, 420
 - pack_dms_GetActivationState, 421
 - pack_dms_GetBandCapability, 421
 - pack_dms_GetCrashAction, 421
 - pack_dms_GetCustFeature, 422
 - pack_dms_GetCustFeaturesV2, 422
 - pack_dms_GetDeviceCap, 422
 - pack_dms_GetDeviceCapabilities, 422
 - pack_dms_GetDeviceHardwareRev, 423
 - pack_dms_GetDeviceMfr, 423
 - pack_dms_GetDeviceSerialNumbers, 423
 - pack_dms_GetFSN, 425
 - pack_dms_GetFirmwareInfo, 424
 - pack_dms_GetFirmwareRevision, 424
 - pack_dms_GetFirmwareRevisions, 425
 - pack_dms_GetHardwareRevision, 425
 - pack_dms_GetIMSI, 426
 - pack_dms_GetManufacturer, 426
 - pack_dms_GetModelID, 426
 - pack_dms_GetNetworkTime, 427
 - pack_dms_GetOfflineReason, 427
 - pack_dms_GetPRLVersion, 428
 - pack_dms_GetPower, 427
 - pack_dms_GetSerialNumbers, 428
 - pack_dms_GetUSBComp, 428
 - pack_dms_GetVoiceNumber, 429
 - pack_dms_SLQSDmsSwiGetResetInfo, 432
 - pack_dms_SLQSDmsSwiIndicationRegister, 432
 - pack_dms_SLQSGetBandCapability, 433
 - pack_dms_SLQSSwiClearDyingGaspStatistics, 433
 - pack_dms_SLQSSwiGetDyingGaspCfg, 434
 - pack_dms_SLQSSwiGetDyingGaspStatistics, 434
 - pack_dms_SLQSSwiGetFirmwareCurr, 434
 - pack_dms_SLQSSwiGetFwUpdateStatus, 435
 - pack_dms_SLQSSwiSetDyingGaspCfg, 435
 - pack_dms_SetCrashAction, 429
 - pack_dms_SetCustFeature, 430
 - pack_dms_SetCustFeaturesV2, 430
 - pack_dms_SetEventReport, 431
 - pack_dms_SetFirmwarePreference, 431
 - pack_dms_SetPower, 431
 - pack_dms_SetUSBComp, 432
 - pack_dms_UIMGetICCID, 435
 - UNIQUE_ID_LEN, 421
 - unpack_dms_GetActivationState, 436
 - unpack_dms_GetBandCapability, 436
 - unpack_dms_GetCrashAction, 436
 - unpack_dms_GetCustFeature, 437
 - unpack_dms_GetCustFeaturesV2, 437
 - unpack_dms_GetDeviceCap, 437
 - unpack_dms_GetDeviceCapabilities, 438
 - unpack_dms_GetDeviceHardwareRev, 438
 - unpack_dms_GetDeviceMfr, 438
 - unpack_dms_GetDeviceSerialNumbers, 439
 - unpack_dms_GetFSN, 440
 - unpack_dms_GetFirmwareInfo, 439

- unpack_dms_GetFirmwareRevision, 439
- unpack_dms_GetFirmwareRevisions, 440
- unpack_dms_GetHardwareRevision, 440
- unpack_dms_GetIMSI, 441
- unpack_dms_GetManufacturer, 441
- unpack_dms_GetModelID, 441
- unpack_dms_GetNetworkTime, 442
- unpack_dms_GetOfflineReason, 442
- unpack_dms_GetPRLVersion, 443
- unpack_dms_GetPower, 442
- unpack_dms_GetSerialNumbers, 443
- unpack_dms_GetUSBComp, 443
- unpack_dms_GetVoiceNumber, 444
- unpack_dms_SLQSDmsSwiGetResetInfo, 447
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 447
- unpack_dms_SLQSDmsSwiIndicationRegister, 448
- unpack_dms_SLQSGetBandCapability, 448
- unpack_dms_SLQSSwiClearDyingGaspStatistics, 448
- unpack_dms_SLQSSwiGetDyingGaspCfg, 449
- unpack_dms_SLQSSwiGetDyingGaspStatistics, 449
- unpack_dms_SLQSSwiGetFirmwareCurr, 449
- unpack_dms_SLQSSwiGetFwUpdateStatus, 450
- unpack_dms_SLQSSwiSetDyingGaspCfg, 450
- unpack_dms_SetCrashAction, 444
- unpack_dms_SetCustFeature, 444
- unpack_dms_SetCustFeaturesV2, 445
- unpack_dms_SetEventReport, 445
- unpack_dms_SetEventReport_ind, 445
- unpack_dms_SetFirmwarePreference, 446
- unpack_dms_SetPower, 446
- unpack_dms_SetUSBComp, 446
- unpack_dms_UIMGetICCID, 451
- dms_ActivationStatusTlv, 29
 - activationStatus, 30
 - TlvPresent, 30
- dms_OperatingModeTlv, 30
 - operatingMode, 30
 - TlvPresent, 30
- domain
 - wds_DomainNameList, 401
- domainLen
 - wds_Domain, 401
- DomainList
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- domainName
 - wds_Domain, 401
- dormancyStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- dormancyState
 - unpack_wds_GetDormancyState_t, 379
- dormancyStatus
 - pack_wds_SLQSSetWdsEventCallback_t, 233
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- dtmSupp
 - nas_GSMSysInfo, 104
- dtmSuppValid
 - nas_GSMSysInfo, 104
- dunchannelRate, 33
 - CurrChanRxRate, 34
 - CurrChanTxRate, 34
 - MaxChanRxRate, 34
 - MaxChanTxRate, 34
- Duration
 - pack_nas_SetNetworkPreference_t, 187
 - unpack_nas_GetNetworkPreference_t, 308
- eCTL
 - common.h, 413
- eDMS
 - common.h, 413
- eFW_TYPE_CWE
 - libsdh.h, 457
- eFW_TYPE_CWE_NVU
 - libsdh.h, 457
- eFW_TYPE_INVALID
 - libsdh.h, 457
- eFW_TYPE_MBN
 - libsdh.h, 457
- eFW_TYPE_MBN_GOBI
 - libsdh.h, 457
- eFW_TYPE_NVU
 - libsdh.h, 457
- eFW_TYPE_SPK
 - libsdh.h, 457
- eIND
 - common.h, 414
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
 - nas.h, 478
- eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
 - nas.h, 478
- eLOC
 - common.h, 413
- eLOG_DEBUG

- common.h, [413](#)
- eLOG_FATAL
 - common.h, [413](#)
- eLOG_INFO
 - common.h, [413](#)
- eLOG_WARN
 - common.h, [413](#)
- eModel_9X15
 - libsdh.h, [457](#)
- eModel_9X30
 - libsdh.h, [457](#)
- eModel_Unknown
 - libsdh.h, [457](#)
- eNAS
 - common.h, [413](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
 - nas.h, [478](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
 - nas.h, [478](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
 - nas.h, [478](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
 - nas.h, [478](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
 - nas.h, [478](#)
- eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
 - nas.h, [478](#)
- eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
 - nas.h, [478](#)
- eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
 - nas.h, [478](#)
- eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
 - nas.h, [478](#)
- eQCWWAN_ERR_API_MUTEX_TIMEOUT
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_BUFFER_SZ
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_CANCEL_OP
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_DRIVER
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_ENUM_BEGIN
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_ENUM_END
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_FILE_COPY
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_FILE_OPEN
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_GENERAL
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_INTERNAL
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_INVALID_ARG
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_INVALID_DEVID
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_INVALID_FILE
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_INVALID_QMI_RSP
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_INVALID_XID
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_MALFORMED_QMI_RSP
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_MEMORY
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_MULTIPLE_DEVICES
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_NO_CANCELABLE_OP
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_NO_CONNECTION
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_NO_DEVICE
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_NO_SIGNAL
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_NONE
 - qmerrno.h, [540](#)
- eQCWWAN_ERR_NULL_TLV
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_OFFLINE
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_PDU_GENERATION
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_ABORTED
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_ACCESS_DENIED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_ACK_NOT_SENT
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_ARG_TOO_LONG
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
 - qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_CALL_FAILED
 - qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_CARD_BUSY_RSP
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
 - qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_CAT_END
 - qmerrno.h, [544](#)
- eQCWWAN_ERR_QMI_CAT_START
 - qmerrno.h, [544](#)

eQCWWAN_ERR_QMI_CAUSE_CODE
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_CONNECT
 qmerrno.h, 540
 eQCWWAN_ERR_QMI_DEVICE_IN_USE
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_DEVICE_NOT_READY
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_DISABLED
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_ENCODING
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE
 qmerrno.h, 544
 eQCWWAN_ERR_QMI_EVENT_REG_FAILED
 qmerrno.h, 544
 eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_FDN_RESTRICT
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_FLOW_SUSPENDED
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_GENERAL
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_IFACE
 qmerrno.h, 540
 eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INCORRECT_PIN
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_INJECT_TIMEOUT
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCE-
 S
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INTERNAL
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_ARG
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
 qmerrno.h, 544
 eQCWWAN_ERR_QMI_INVALID_HANDLE
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_ID
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_INDEX
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_OPERATION
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_PINID
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_PROFILE
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTI-
 ON
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_QMI_CMD
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_INVALID_QOS_ID
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTIO-
 N
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_TECH_PREF
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
 qmerrno.h, 544
 eQCWWAN_ERR_QMI_INVALID_TRANSITION
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_INVALID_TX_ID
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_MALFORMED_MSG
 qmerrno.h, 541
 eQCWWAN_ERR_QMI_MAX
 qmerrno.h, 543
 eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_I-
 N_USE
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_-
 USE
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAIL-
 URE
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
 qmerrno.h, 542
 eQCWWAN_ERR_QMI_MISSING_ARG
 qmerrno.h, 541

- eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_REQ
qmerrno.h, [540](#)
- eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, [540](#)
- eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, [540](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_RSP
qmerrno.h, [540](#)
- eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, [540](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, [543](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, [541](#)
- eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, [542](#)
- eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, [544](#)
- eQCWWAN_ERR_RESET
qmerrno.h, [541](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_END
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, [543](#)

- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_START
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, [543](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIDCS_END
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIDCS_START
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_END
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWIIM_START
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWISM_END
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, [544](#)
- eQCWWAN_ERR_SWISMS_START
qmerrno.h, [544](#)
- eQMI_LOC_SESS_STATUS_FAILURE
loc.h, [465](#)
- eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, [465](#)
- eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, [465](#)
- eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, [465](#)
- eQOS
common.h, [413](#)
- eREQ
common.h, [414](#)
- eRSP
common.h, [414](#)
- eSDP_FWDWL_ERR_END
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_FAIL
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_FW_UPGRADE
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_FW_VERSION_FAIL
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_GENERAL
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_INVALID_DEV
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_INVALID_PATH
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_PATH_TOO_LONG
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_PRI_FAIL
libsdh.h, [457](#)
- eSDP_FWDWL_ERR_SDK
libsdh.h, [457](#)

- eSDP_FWDWL_ERR_SET_CBK
 - libsdp.h, [457](#)
- eSDP_FWDWL_ERR_TIMEOUT
 - libsdp.h, [457](#)
- eSDP_FWDWL_SUCCESS
 - libsdp.h, [457](#)
- eSMS
 - common.h, [413](#)
- eSWILOC
 - common.h, [414](#)
- eSWIOMA
 - common.h, [414](#)
- eTIMEOUT_10_S
 - common.h, [414](#)
- eTIMEOUT_20_S
 - common.h, [414](#)
- eTIMEOUT_2_S
 - common.h, [414](#)
- eTIMEOUT_300_S
 - common.h, [414](#)
- eTIMEOUT_30_S
 - common.h, [414](#)
- eTIMEOUT_5_S
 - common.h, [414](#)
- eTIMEOUT_60_S
 - common.h, [414](#)
- eTIMEOUT_8_S
 - common.h, [414](#)
- eTIMEOUT_DEFAULT
 - common.h, [414](#)
- eTMD
 - common.h, [414](#)
- eUIM
 - common.h, [413](#)
- eWDS
 - common.h, [413](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID-
 - _IDENT_FOR_PROFILE
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT-
 - _DEFINED
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_-
 - PROFILES
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFIL-
 - E_FAMILY
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_F-
 - LAG_SET
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_N-
 - OT_SET
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_END
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMI-
 - LY
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_-
 - HNDL
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_-
 - IDENT
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_-
 - OP
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_-
 - PROFILE_NUM
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_-
 - PROFILE_TYPE
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_-
 - SUBS_ID
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_IN-
 - VALID
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NO-
 - T_INITED
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL
 - qmerrno.h, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END
 - qmerrno.h, [545](#)
- eLOG_LEVEL
 - common.h, [413](#)
- EMTlv
 - NASQmiCbkNasSystemSelPrefInd, [171](#)
- eQCWWANError
 - qmerrno.h, [540](#)
- eQMI_SVC
 - common.h, [413](#)
- ESNString
 - unpack_dms_GetDeviceSerialNumbers_t, [267](#)
- ETWSPLMNInfo
 - eTWSPMLNInfoTlv, [34](#)
- eTWSPMLNInfoTlv, [34](#)
 - ETWSPLMNInfo, [34](#)
 - TlvPresent, [34](#)
- ETWSPLMNTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [360](#)
- ETWSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [360](#)
- eTimeout
 - common.h, [414](#)
- eValid
 - LibPackTFTIDParams, [64](#)
- earfcn
 - nas_infoInterFreq, [110](#)

- nas_LTEInfoIntraFreq, 116
- nas_umtsLTENbrCell, 156
- ecio
 - cdmaSSInfo, 28
 - hdrSSInfo, 38
 - nas_ecioListElement, 95
 - nas_UMTSInfo, 154
 - tdscdmaSigInfoExt, 247
- ecioDelta
 - nas_SLQSSignalStrengthsIndReq, 144
- ecioInfo
 - nas_SLQSSignalStrengthsInformation, 145
- ecioList
 - unpack_nas_SLQSGetSignalStrength_t, 319
- ecioListLen
 - unpack_nas_SLQSGetSignalStrength_t, 319
- ecioThresholdList
 - nas_SLQSSignalStrengthsIndReq, 144
- ecioThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, 144
- egprsSupp
 - nas_GSMSSysInfo, 104
- egprsSuppValid
 - nas_GSMSSysInfo, 104
- EmerMode
 - NASEmergencyModeTlv, 164
- emmConnState
 - nas_LTEInfo, 113
- emmState
 - nas_LTEInfo, 113
- emmSubState
 - nas_LTEInfo, 113
- enable
 - pack_qos_SLQSSetQosEventCallback_t, 205
- enabled
 - unpack_wds_GetMobileIPProfile_t, 380
- EncryptedPIN1
 - pack_uim_ChangePin_t, 214
 - pack_uim_SetPinProtection_t, 216
 - pack_uim_UnblockPin_t, 220
- engineState
 - unpack_loc_EngineState_Ind_t, 296
- eqmiCbkSetStatus
 - sms.h, 557
- error
 - unpack_wds_GetLastMobileIPError_t, 379
- errorRate
 - nas_errorRateListElement, 96
- errorRateInfo
 - nas_SLQSSignalStrengthsInformation, 145
- errorRateList
 - unpack_nas_SLQSGetSignalStrength_t, 319
- errorRateListLen
 - unpack_nas_SLQSGetSignalStrength_t, 319
- errorState
 - slotInf, 241
 - uim_slotInfo, 260
- esn
 - unpack_dms_GetSerialNumbers_t, 274
- esnSize
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- EspSpi
 - unpack_qos_swiQosFilter_t, 352
- EtwsMessageInfo
 - sMSEtwsMessageTlv, 243
- event
 - unpack_qos_SLQSSetQosPriEventCallback_ind_t, 348
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 350
- eventMask
 - pack_uim_SLQSUIEventRegister_t, 217
 - unpack_uim_SLQSUIEventRegister_t, 374
- eventRegister
 - pack_loc_EventRegister_t, 183
- eventType
 - unpack_swioma_SLQSOMADMAAlertCallback_ind_t, 365
- executingImage
 - FMSImageIDEntries, 36
- exponent
 - unpack_qos_pktErrRate_t, 341
- ExtErrorCode
 - PackCreateProfileOut, 236
- extPowerState
 - pack_loc_SetExtPowerState_t, 183
- extendedErrorCode
 - unpack_wds_SLQSDeleteProfile_t, 385
- FLOAT
 - SwiDataTypes.h, 563
- FMSImageElement, 34
 - buildId, 35
 - buildIdLength, 35
 - imageId, 35
 - imageType, 35
- FMSImageIDEntries, 36
 - executingImage, 36
 - imageIDElement, 36
 - imageIDSize, 36
 - imageType, 37
 - maxImages, 37
- FMSImageIDElement, 35
 - buildID, 36
 - buildIDLength, 36
 - failureCount, 36
 - imageID, 36
 - storageIndex, 36
- FMSImageList, 37
 - imageIDEntries, 37
 - listSize, 37
- FMSPrefImageList, 37
 - listEntries, 38
 - listSize, 38
- FOTAUpdate
 - pack_swioma_SLQSOMADMSetSettings_t, 213
 - unpack_swioma_SLQSOMADMGetSettings_t, 369

- FOTAdownload
 - pack_swima_SLQSOMADMSetSettings_t, 213
 - unpack_swima_SLQSOMADMGetSettings_t, 369
- failureCount
 - FMSImageIdElement, 36
- family
 - pack_wds_GetDefaultProfileNum_t, 222
 - pack_wds_SetDefaultProfileNum_t, 225
- fileID
 - uim_fileInfo, 255
- fileIndex
 - pack_uim_ReadTransparent_t, 215
- fill_pack_ctx
 - common.h, 414
- fill_sdu_hdr
 - common.h, 414
- filterId
 - LibPackTFTIDParams, 64
- fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, 210
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- fix_rate_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, 210
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- fix_type_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- flowLabel
 - LibPackTFTIDParams, 64
- fms.h, 451
 - GetValidFwPriCombinations, 452
 - pack_fms_GetImagesPreference, 453
 - pack_fms_GetStoredImages, 453
 - pack_fms_SetImagesPreference, 453
 - unpack_fms_GetImagesPreference, 453
 - unpack_fms_GetStoredImages, 454
 - unpack_fms_SetImagesPreference, 454
- Forbidden
 - nas_QmiNas3GppNetworkInfo, 134
- ForceRev0
 - unpack_nas_GetCDMANetworkParameters_t, 306
- format
 - sMSTransferRouteMTMessage, 246
- fqnAddr
 - wds_PCSCFFQDNAddress, 403
- fqnLen
 - wds_PCSCFFQDNAddress, 403
- freq
 - nas_PhyCaAggPcellInfo, 128
 - nas_PhyCaAggScellIndType, 130
 - nas_PhyCaAggScellInfo, 133
 - NASPhyCaAggPcellInfo, 167
 - NASPhyCaAggScellIndType, 168
 - NASPhyCaAggScellInfo, 169
- freqsLen
 - nas_LTEInfoInterfreq, 114
 - nas_LTEInfoNeighboringGSM, 117
 - nas_LTEInfoNeighboringWCDMA, 118
- function
 - pack_swiloc_SwiLocSetAutoStart_t, 210
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- function_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- FwAutoCheck
 - unpack_swima_SLQSOMADMGetSettings_t, 369
- FwAvailability
 - unpack_swima_SLQSOMADMStartSession_t, 370
- fwloadsize
 - unpack_omaDmFotaTlv_t, 338
- fwloadComplete
 - unpack_omaDmFotaTlv_t, 338
- fwvers
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 285
- gDIBitRate
 - LibPackQosClassID, 63
- GPRSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- GPSPMP
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- GPSSel
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- GSMRSSIThreshListLen
 - nas_GSMRSSIThresh, 101
- GSMSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 328
- gUIBitRate
 - LibPackQosClassID, 63
- GWAOPTlv
 - NASQmiCbkNasSystemSelPrefInd, 171
- GWAcqOrderPref
 - NASGWAcqOrderPrefTlv, 164
- GWAddressV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- geoSysIdx
 - nas_AddCDMASysInfo, 80
 - nas_AddSysInfo, 80
- geranArfcn
 - nas_geranInstInfo, 99
- geranBsicBcc
 - nas_geranInstInfo, 99
- geranBsicNcc
 - nas_geranInstInfo, 99
- geranInst
 - nas_UMTSInfo, 154
- GeranInstInfo
 - nas_UMTSInfo, 154
- geranRssi
 - nas_geranInstInfo, 99
- get_version
 - common.h, 414
- GetCustomFeatureV2
 - unpack_dms_GetCustFeaturesV2_t, 265

- GetValidFwPriCombinations
 - fms.h, [452](#)
- globalCellId
 - nas_LTEInfoIntrafreq, [116](#)
- glog
 - common.h, [415](#)
- gloglvl
 - common.h, [415](#)
- gnssSvUsedList
 - loc_svUsedforFix, [78](#)
- gnssSvUsedList_len
 - loc_svUsedforFix, [78](#)
- Gpp2TimeZone
 - unpack_nas_SLQSGetServingSystem_t, [318](#)
- GppNetworkDSTAdjustment
 - unpack_nas_SLQSGetServingSystem_t, [318](#)
- GppTimeZone
 - unpack_nas_SLQSGetServingSystem_t, [318](#)
- GpsEnable
 - pack_dms_SetCustFeature_t, [175](#)
 - unpack_dms_GetCustFeature_t, [264](#)
- gpsTimeOfWeekMs
 - loc_gpsTime, [73](#)
- gpsWeek
 - loc_gpsTime, [73](#)
- grntDownlinkBitrate
 - LibPackUMTSQoS, [67](#)
 - wds_UMTSMInQoS, [407](#)
- grntUplinkBitrate
 - LibPackUMTSQoS, [67](#)
 - wds_UMTSMInQoS, [407](#)
- GsmCellInfo
 - nas_lteGsmCellInfo, [111](#)
- gsmUmtsDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [197](#)
- gsmUmtsUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, [198](#)
- guaranteedRate
 - unpack_qos_dataRate_t, [339](#)
- gwAddressV6
 - wds_IPV6GWAddressInfo, [403](#)
- gwV6PrefixLen
 - wds_IPV6GWAddressInfo, [403](#)
- HASPI
 - unpack_wds_GetMobileIPProfile_t, [380](#)
- HASState
 - unpack_wds_GetMobileIPProfile_t, [380](#)
- HDOP
 - loc_precisionDilution, [75](#)
- HDRECIOTreshListLen
 - nas_HDRECIOTresh, [105](#)
- HDRIOTreshListLen
 - nas_HDRIOTresh, [106](#)
- HDRRSSITreshListLen
 - nas_HDRRSSITresh, [106](#)
- HDRSINRThreshListLen
 - nas_HDRSINRThreshold, [107](#)
- HDRSSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, [328](#)
- HardwareControlledMode
 - unpack_dms_GetPower_t, [273](#)
- hdrActiveProt
 - nas_HDRSysInfo, [109](#)
- hdrActiveProtValid
 - nas_HDRSysInfo, [109](#)
- hdrHybrid
 - nas_detailSvcInfo, [95](#)
- HdrPersonality
 - unpack_nas_SLQSGetServingSystem_t, [318](#)
- hdrPersonality
 - nas_HDRSysInfo, [109](#)
 - NASServingSystemInfo, [173](#)
- hdrPersonalityValid
 - nas_HDRSysInfo, [109](#)
- hdrSSInfo, [38](#)
 - ecio, [38](#)
 - io, [38](#)
 - rsi, [38](#)
 - sinr, [38](#)
- hdrSrvStatus
 - nas_detailSvcInfo, [95](#)
- helper_get_resp_ctx
 - common.h, [414](#)
- helper_get_xid
 - common.h, [415](#)
- helper_set_log_func
 - common.h, [415](#)
- helper_set_log_lvl
 - common.h, [415](#)
- hotSwap
 - uim_hotSwapStatus, [255](#)
- hotSwapLength
 - uim_hotSwapStatus, [255](#)
- hour
 - nas_timeInfo, [152](#)
 - nas_UniversalTime, [157](#)
- hsCallStatus
 - nas_WCDMASysInfo, [163](#)
- hsCallStatusValid
 - nas_WCDMASysInfo, [163](#)
- hsInd
 - nas_WCDMASysInfo, [163](#)
- hsIndValid
 - nas_WCDMASysInfo, [163](#)
- hwType
 - wdsDhcpv4HwConfig, [408](#)
- hwVer
 - unpack_dms_GetHardwareRevision_t, [270](#)
- iLTEbandValue
 - nas_PhyCaAggPcellInfo, [128](#)
 - nas_PhyCaAggScellInfo, [133](#)
 - NASPhyCaAggPcellInfo, [167](#)
 - NASPhyCaAggScellInfo, [170](#)
- IMCNflag

- unpack_wds_SLQSGetRuntimeSettings_t, 391
- IMEIString
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- IMG_MASK_CLEAR
 - libsdh.h, 456
- IMG_MASK_GENERIC
 - libsdh.h, 456
- IMG_MASK_MDM
 - libsdh.h, 456
- IMG_MASK_PRI
 - libsdh.h, 456
- IMSInfo
 - sMSOnIMSTlv, 245
- IMSTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 360
- INT32
 - SwiDataTypes.h, 563
- INT8
 - SwiDataTypes.h, 563
- IPAddressV6
 - ipv6AddressInfo, 39
 - wds_IPV6AddressInfo, 402
- IPFamSupport
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- IPFamilyPreference
 - pack_wds_SLQSSetIPFamilyPreference_t, 232
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- IPSECSPi
 - LibPackTFTIDParams, 64
- IPv6AddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- IPv6GWAddrInfo
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- IPv6PrefixLen
 - ipv6AddressInfo, 39
 - wds_IPV6AddressInfo, 402
- IPv4
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- IPv4DstAddr
 - unpack_qos_swiQosFilter_t, 352
- IPv4SrcAddr
 - unpack_qos_swiQosFilter_t, 352
- IPv4Tos
 - unpack_qos_swiQosFilter_t, 352
- IPv6DstAddr
 - unpack_qos_swiQosFilter_t, 352
- IPv6Label
 - unpack_qos_swiQosFilter_t, 352
- IPv6SrcAddr
 - unpack_qos_swiQosFilter_t, 352
- IPv6TrafCls
 - unpack_qos_swiQosFilter_t, 352
- Id
 - unpack_qos_swiQosFilter_t, 352
- id
 - loc_BdsSV, 68
 - loc_SV, 76
- nas_CSGID, 91
- unpack_qos_QosFlowInfoState_t, 344
- unpack_qos_SLQSSetQosStatusCallback_ind_t, 350
- id_length
 - DMScustSettingInfo, 31
- image_info_t, 38
 - buildID, 38
 - buildIDLen, 38
 - imageType, 38
 - uniqueID, 38
- imageID
 - FMSImageIdElement, 36
- imageIDElement
 - FMSImageIDEntries, 36
- imageIDEntries
 - FMSImageList, 37
- imageIDSize
 - FMSImageIDEntries, 36
- imageId
 - FMSImageElement, 35
- imageList
 - unpack_fms_GetStoredImages_t, 289
- ImageListSize
 - unpack_fms_GetImagesPreference_t, 288
- imageListSize
 - pack_fms_SetImagesPreference_t, 180
- imageType
 - FMSImageElement, 35
 - FMSImageIDEntries, 37
 - image_info_t, 38
- ImageTypes
 - unpack_fms_SetImagesPreference_t, 289
- ImageTypesSize
 - unpack_fms_SetImagesPreference_t, 289
- imagelistSize
 - unpack_fms_GetStoredImages_t, 289
- imei_no
 - unpack_dms_GetSerialNumbers_t, 274
- imeiSize
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- imeiSvnSize
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- ImeiSvnString
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- imeisv_svn
 - unpack_dms_GetSerialNumbers_t, 274
- imgType
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 286
- imsRegState
 - nas_CommInfo, 91
- imsi
 - unpack_dms_GetIMSI_t, 270
- imsi_11_12
 - nas_CDMASysInfoExt, 88
- InUse
 - nas_QmiNas3GppNetworkInfo, 134
- includes_pcs_digit

- nas_QmisNasPcsDigit, [135](#)
- index
 - pack_wds_GetMobileIPProfile_t, [222](#)
 - pack_wds_SetDefaultProfileNum_t, [225](#)
 - pack_wds_SetMobileIPProfile_t, [225](#)
 - unpack_qos_swiQosFilter_t, [352](#)
 - unpack_qos_swiQosFlow_t, [356](#)
 - unpack_wds_GetDefaultProfileNum_t, [379](#)
- index1xPri
 - uim_cardStatus, [252](#)
- index1xSec
 - uim_cardStatus, [252](#)
- indexGwPri
 - uim_cardStatus, [252](#)
- indexGwSec
 - uim_cardStatus, [252](#)
- Info
 - unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, [331](#)
 - unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, [331](#)
- InfoInterfreq
 - nas_LTEInfoInterfreq, [114](#)
- insNmrCellInfo
 - nas_GERANInfo, [98](#)
- instancesSize
 - unpack_nas_GetRFInfo_t, [308](#)
- interval
 - pack_wds_SLQSSetWdsEventCallback_t, [233](#)
- Io
 - unpack_nas_SLQSGetSignalStrength_t, [319](#)
- io
 - hdrSSInfo, [38](#)
 - nas_SLQSSignalStrengthsInformation, [145](#)
- ioDelta
 - nas_SLQSSignalStrengthsIndReq, [144](#)
- ipAddress
 - pack_wds_SetDefaultProfile_t, [224](#)
- ipFamily
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [393](#)
- ipVersion
 - LibPackTFTIDParams, [64](#)
- ipaddr
 - unpack_wds_GetDefaultProfile_t, [378](#)
- ipaddrv6
 - unpack_wds_GetDefaultProfile_t, [378](#)
- ipv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- ipv4GWAddress
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- ipv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- ipv6AddressInfo, [39](#)
- IPAddressV6, [39](#)
- IPv6PrefixLen, [39](#)
- ipv6GWAddress
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- is856SysId
 - nas_HDRSysInfo, [109](#)
- is856SysIdValid
 - nas_HDRSysInfo, [109](#)
- is_DataRate_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_EspSpi_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv4DstAddr_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv4SrcAddr_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv4Tos_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv6DstAddr_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv6Label_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv6SrcAddr_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_IPv6TrafCls_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_Id_Available
 - unpack_qos_swiQosFilter_t, [352](#)
- is_Jitter_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_Latency_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_LteBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, [283](#)
- is_LteQci_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_MaxAllowedPktSz_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_MinPolicedPktSz_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_NxtHdrProto_Available
 - unpack_qos_swiQosFilter_t, [353](#)
- is_PktErrRate_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_Precedence_Available
 - unpack_qos_swiQosFilter_t, [353](#)
- is_ProfileId3GPP2_Available
 - unpack_qos_swiQosFlow_t, [356](#)
- is_RxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, [343](#)
- is_TCPDstPort_Available
 - unpack_qos_swiQosFilter_t, [353](#)
- is_TCPSrcPort_Available
 - unpack_qos_swiQosFilter_t, [353](#)
- is_TdsBandCapability_Available
 - unpack_dms_SLQSGetBandCapability_t, [283](#)
- is-TokenBucket_Available
 - unpack_qos_swiQosFlow_t, [356](#)

- is_TrafficClass_Available
 - unpack_qos_swiQosFlow_t, 356
- is_TranDstPort_Available
 - unpack_qos_swiQosFilter_t, 353
- is_TranSrcPort_Available
 - unpack_qos_swiQosFilter_t, 353
- is_TxQFlowGranted_Available
 - unpack_qos_QosFlowInfo_t, 343
- is_UDPDstPort_Available
 - unpack_qos_swiQosFilter_t, 353
- is_UDPSrcPort_Available
 - unpack_qos_swiQosFilter_t, 353
- is_val_3GPP2Pri_Available
 - unpack_qos_swiQosFlow_t, 356
- is_val_3GPPImCn_Available
 - unpack_qos_swiQosFlow_t, 356
- is_val_3GPPResResidualBER_Available
 - unpack_qos_swiQosFlow_t, 356
- is_val_3GPPSigInd_Available
 - unpack_qos_swiQosFlow_t, 357
- is_val_3GPPTraHdlPri_Available
 - unpack_qos_swiQosFlow_t, 357
- isNewFlow
 - unpack_qos_QosFlowInfoState_t, 344
- isPrefDataPath
 - nas_GSMSrvStatusInfo, 102
 - nas_SrvStatusInfo, 146
- isRadioTuned
 - nas_RxSigInfo, 139
- isSysForbidden
 - nas_detailSvcInfo, 95
 - nas_sysInfoCommon, 148
- isSysForbiddenValid
 - nas_sysInfoCommon, 148
- isSysPriMatch
 - nas_CDMASysInfo, 87
 - nas_HDRSysInfo, 109
- isSysPriMatchValid
 - nas_CDMASysInfo, 87
 - nas_HDRSysInfo, 109
- IsVoiceEnabled
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- Jitter
 - unpack_qos_swiQosFlow_t, 357
- LIBPACK_QMI_CBK_PARAM_NOCHANGE
 - sms.h, 557
- LIBPACK_QMI_CBK_PARAM_RESET
 - sms.h, 557
- LIBPACK_QMI_CBK_PARAM_SET
 - sms.h, 557
- LBPTiv
 - NASQmiCbkNasSystemSelPrefInd, 171
- LOCEVENTMASKGNSSSVINFO
 - loc.h, 463
- LOCEVENTMASKNMEA
 - loc.h, 464
- LOCEVENTMASKWIFIREQ
 - loc.h, 465
- LPCSTR
 - SwiDataTypes.h, 563
- LTEAttachProfile
 - unpack_wds_SLQSGet3GPPConfigItem_t, 386
- LTEAttachProfileList
 - unpack_wds_SLQSGet3GPPConfigItem_t, 386
- LTEAttachProfileListLen
 - pack_wds_SLQSSet3GPPConfigItem_t, 231
 - unpack_wds_SLQSGet3GPPConfigItem_t, 386
- LTETBandPref
 - NASLTETBandPreferenceTlv, 165
- LTETCphyCAInfo
 - unpack_nas_SlqsGetLTETCphyCAInfo_t, 315
- LTERSRPThreshListLen
 - nas_LTERSRPThresh, 118
- LTERSRQThreshListLen
 - nas_LTERSRQThresh, 119
- LTERSSIThreshListLen
 - nas_LTERSSIThresh, 120
- LTESNRThreshListLen
 - nas_LTESNRThreshold, 121
- LTESInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 328
- LTEWCDMACellInfo
 - nas_LTEInfoNeighboringWCDMA, 118
- Lac
 - unpack_nas_SLQSGetServingSystem_t, 318
- lac
 - nas_GERANInfo, 98
 - nas_GSMSysInfo, 104
 - nas_LTESysInfo, 123
 - nas_UMTSInfo, 154
 - nas_WCDMASysInfo, 163
- lacValid
 - nas_GSMSysInfo, 104
 - nas_LTESysInfo, 123
 - nas_WCDMASysInfo, 163
- lastCallDataBearerTech
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- lastCallDataBearerTechnology
 - unpack_wds_SLQSGetDataBearerTechnology_t, 388
- lastCallRXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- lastCallTXOKBytesCnt
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- Latency
 - unpack_qos_swiQosFlow_t, 357
- leapSeconds
 - nas_qaQmi3Gpp2TimeZone, 133
- len
 - loc_BdsSVInfo, 69
 - loc_SVInfo, 77
 - unpack_nas_GetSignalStrengths_t, 310
- length
 - sMSCAddress, 242

- sMSEtwMessage, 243
- sMSTransferRouteMTMessage, 246
- uim_readTransparentInfo, 256
- LibPackGPRSRequestedQoS, 39
 - delayClass, 40
 - meanThroughputClass, 40
 - peakThroughputClass, 40
 - precedenceClass, 40
 - reliabilityClass, 40
- LibPackQosClassID, 62
 - gDIBitRate, 63
 - gUIBitRate, 63
 - maxDIBitRate, 63
 - maxUIBitRate, 63
 - QCI, 63
- LibPackTFTIDParams, 63
 - destPortRangeEnd, 64
 - destPortRangeStart, 64
 - eValid, 64
 - filterId, 64
 - flowLabel, 64
 - IPSECSPi, 64
 - ipVersion, 64
 - nextHeader, 65
 - pSourceIP, 65
 - sourceIPMask, 65
 - srcPortRangeEnd, 65
 - srcPortRangeStart, 65
 - tosMask, 65
- LibPackUMTSQoS, 65
 - deliveryErrSDU, 67
 - grntDownlinkBitrate, 67
 - grntUplinkBitrate, 67
 - maxDownlinkBitrate, 67
 - maxSDUSize, 67
 - maxUplinkBitrate, 67
 - qosDeliveryOrder, 67
 - resBerRatio, 67
 - sduErrorRatio, 67
 - trafficClass, 67
 - trafficPriority, 67
 - transferDelay, 67
- LibPackUMTSReqQoSsigInd, 67
 - SigInd, 68
 - UMTSReqQoS, 68
- LibPackprofile_3GPP, 51
 - pAPNClass, 55
 - pAPNDisabledFlag, 55
 - pAPNName, 55
 - pAPNnameSize, 55
 - pAddrAllocPref, 55
 - pAuthenticationPref, 56
 - pGPRSMinimumQoS, 56
 - pGPRSRequestedQoS, 56
 - pIPv4AddrPref, 56
 - pIPv6AddPref, 56
 - pImCnFlag, 56
 - pPDNInactivTimeout, 56
 - pPDType, 56
 - pPassword, 56
 - pPasswordSize, 56
 - pPcscfAddrUsingDhcp, 56
 - pPcscfAddrUsingPCO, 56
 - pPdpAccessConFlag, 56
 - pPdpContext, 56
 - pPdpDataCompType, 56
 - pPdpHdrCompType, 56
 - pPriDNSIPv4AddPref, 56
 - pPriDNSIPv6addpref, 56
 - pPrimaryID, 56
 - pProfilename, 56
 - pProfilenameSize, 56
 - pQosClassID, 56
 - pSecDNSIPv4AddPref, 56
 - pSecDNSIPv6addpref, 56
 - pSecondaryFlag, 56
 - pTFTID1Params, 56
 - pTFTID2Params, 56
 - pUMTSMInQoS, 56
 - pUMTSMInQoSsigInd, 57
 - pUMTSReqQoS, 57
 - pUMTSReqQoSsigInd, 57
 - pUsername, 57
 - pUsernameSize, 57
- LibPackprofile_3GPP2, 57
 - pAPNClass3GPP2, 61
 - pAPNEnabled3GPP2, 61
 - pAllowLinger, 61
 - pApnString, 61
 - pApnStringSize, 61
 - pAppPriority, 61
 - pAppType, 61
 - pAuthPassword, 61
 - pAuthPassword_tSize, 61
 - pAuthProtocol, 61
 - pAuthRetryCount, 61
 - pAuthTimeout, 61
 - pDataMode, 61
 - pDataRate, 61
 - plpcpAckTimeout, 61
 - plpcpCreqRetryCount, 61
 - plsPcscfAddressNedded, 61
 - pLcpAckTimeout, 61
 - pLcpCreqRetryCount, 61
 - pNegoDnsSrvrPref, 62
 - pPDNInactivTimeout3GPP2, 62
 - pPdnType, 62
 - pPppSessCloseTimer1x, 62
 - pPppSessCloseTimerDO, 62
 - pPriV6DnsAddress, 62
 - pPrimaryV4DnsAddress, 62
 - pRATType, 62
 - pSecV6DnsAddress, 62
 - pSecondaryV4DnsAddress, 62
 - pUserId, 62
 - pUserIdSize, 62

- libSDP_BuildImagesPreferenceRequest
 - libsdp.h, [458](#)
- libSDP_CalculateImageMask
 - libsdp.h, [458](#)
- libSDP_CheckValidFirmwareInfo
 - libsdp.h, [458](#)
- libSDP_DownloadFW
 - libsdp.h, [458](#)
- libSDP_ExtractFirmwareParametersByPath
 - libsdp.h, [459](#)
- libSDP_FirmwareInfo
 - libsdp.h, [456](#)
- libSDP_Fw_Type
 - libsdp.h, [457](#)
- libSDP_GetModelFamily
 - libsdp.h, [460](#)
- libSDP_GetVersion
 - libsdp.h, [460](#)
- libSDP_Models
 - libsdp.h, [457](#)
- libSDP_fwdwl_error_codes
 - libsdp.h, [457](#)
- libSDP_getFileType
 - libsdp.h, [459](#)
- libpack_GetVersion
 - common.h, [415](#)
- libpack_log
 - common.h, [415](#)
- LibpackProfile3GPP, [40](#)
 - pAPNClass, [44](#)
 - pAPNDisabledFlag, [44](#)
 - pAPNName, [44](#)
 - pAPNnameSize, [44](#)
 - pAddrAllocPref, [44](#)
 - pAuthenticationPref, [44](#)
 - pGPRSMinimumQoS, [44](#)
 - pGPRSRequestedQos, [44](#)
 - pIPv4AddrPref, [45](#)
 - pIPv6AddPref, [45](#)
 - pImCnFlag, [44](#)
 - pPDNInactivTimeout, [45](#)
 - pPDpType, [45](#)
 - pPassword, [45](#)
 - pPasswordSize, [45](#)
 - pPcscfAddrUsingDhcp, [45](#)
 - pPcscfAddrUsingPCO, [45](#)
 - pPdpAccessConFlag, [45](#)
 - pPdpContext, [45](#)
 - pPdpDataCompType, [45](#)
 - pPdpHdrCompType, [45](#)
 - pPriDNSIPv4AddPref, [45](#)
 - pPriDNSIPv6addpref, [45](#)
 - pPrimaryID, [45](#)
 - pProfileName, [45](#)
 - pProfileNameSize, [45](#)
 - pQosClassID, [45](#)
 - pSecDNSIPv4AddPref, [45](#)
 - pSecDNSIPv6addpref, [45](#)
 - pSecondaryFlag, [45](#)
 - pTFTID1Params, [45](#)
 - pTFTID2Params, [45](#)
 - pUMTSMinQoS, [45](#)
 - pUMTSMinQosSigInd, [45](#)
 - pUMTSReqQoS, [45](#)
 - pUMTSReqQoSSigInd, [45](#)
 - pUsername, [45](#)
 - pUsernameSize, [46](#)
- LibpackProfile3GPP2, [46](#)
 - pAPNClass3GPP2, [50](#)
 - pAPNEnabled3GPP2, [50](#)
 - pAllowLinger, [50](#)
 - pApnString, [50](#)
 - pApnStringSize, [50](#)
 - pAppPriority, [50](#)
 - pAppType, [50](#)
 - pAuthPassword, [50](#)
 - pAuthPasswordSize, [50](#)
 - pAuthProtocol, [50](#)
 - pAuthRetryCount, [50](#)
 - pAuthTimeout, [50](#)
 - pDataMode, [50](#)
 - pDataRate, [50](#)
 - pIpcpAckTimeout, [50](#)
 - pIpcpCreqRetryCount, [50](#)
 - pIscfAddressNedded, [50](#)
 - pLcpAckTimeout, [50](#)
 - pLcpCreqRetryCount, [50](#)
 - pNegoDnsSrvrPref, [51](#)
 - pPDNInactivTimeout3GPP2, [51](#)
 - pPdnType, [51](#)
 - pPppSessCloseTimer1x, [51](#)
 - pPppSessCloseTimerDO, [51](#)
 - pPriV6DnsAddress, [51](#)
 - pPrimaryV4DnsAddress, [51](#)
 - pRATType, [51](#)
 - pSecV6DnsAddress, [51](#)
 - pSecondaryV4DnsAddress, [51](#)
 - pUserId, [51](#)
 - pUserIdSize, [51](#)
- libsdp.h
 - eFW_TYPE_CWE, [457](#)
 - eFW_TYPE_CWE_NVU, [457](#)
 - eFW_TYPE_INVALID, [457](#)
 - eFW_TYPE_MBN, [457](#)
 - eFW_TYPE_MBN_GOBI, [457](#)
 - eFW_TYPE_NVU, [457](#)
 - eFW_TYPE_SPK, [457](#)
 - eModel_9X15, [457](#)
 - eModel_9X30, [457](#)
 - eModel_Unknown, [457](#)
 - eSDP_FWDWL_ERR_END, [457](#)
 - eSDP_FWDWL_ERR_FAIL, [457](#)
 - eSDP_FWDWL_ERR_FW_UPGRADE, [457](#)
 - eSDP_FWDWL_ERR_FW_VERSION_FAIL, [457](#)
 - eSDP_FWDWL_ERR_GENERAL, [457](#)
 - eSDP_FWDWL_ERR_INVALID_DEV, [457](#)

- eSDP_FWDWL_ERR_INVALID_PATH, 457
- eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED, 457
- eSDP_FWDWL_ERR_PATH_TOO_LONG, 457
- eSDP_FWDWL_ERR_PRI_FAIL, 457
- eSDP_FWDWL_ERR_SDK, 457
- eSDP_FWDWL_ERR_SET_CBK, 457
- eSDP_FWDWL_ERR_TIMEOUT, 457
- eSDP_FWDWL_SUCCESS, 457
- libsdp.h, 454
 - IMG_MASK_CLEAR, 456
 - IMG_MASK_GENERIC, 456
 - IMG_MASK_MDM, 456
 - IMG_MASK_PRI, 456
 - libSDP_BuildImagesPreferenceRequest, 458
 - libSDP_CalculateImageMask, 458
 - libSDP_CheckValidFirmwareInfo, 458
 - libSDP_DownloadFW, 458
 - libSDP_ExtractFirmwareParametersByPath, 459
 - libSDP_FirmwareInfo, 456
 - libSDP_Fw_Type, 457
 - libSDP_GetModelFamily, 460
 - libSDP_GetVersion, 460
 - libSDP_Models, 457
 - libSDP_fwdwl_error_codes, 457
 - libSDP_getFileType, 459
 - libsdp_set_log_func, 460
 - libsdplogger, 456
- libsdp_set_log_func
 - libsdp.h, 460
- libsdplogger
 - libsdp.h, 456
- list_type
 - DMScustSettingList, 32
 - DMSgetCustomInput, 33
 - pack_dms_GetCustFeaturesV2_t, 174
- listEntries
 - FMSPrefImageList, 38
- listSize
 - FMSImageList, 37
 - FMSPrefImageList, 38
- loc.h
 - eQMI_LOC_SESS_STATUS_FAILURE, 465
 - eQMI_LOC_SESS_STATUS_IN_PROGRESS, 465
 - eQMI_LOC_SESS_STATUS_SUCCESS, 465
 - eQMI_LOC_SESS_STATUS_TIMEOUT, 465
- loc.h, 461
 - LOCEVENTMASKNMEA, 464
 - LOCEVENTMASKWIFIREQ, 465
 - pack_loc_DeleteAssistData, 465
 - pack_loc_EventRegister, 466
 - pack_loc_SLQSLOCGetBestAvailPos, 467
 - pack_loc_SetExtPowerState, 466
 - pack_loc_SetOperationMode, 466
 - pack_loc_Start, 467
 - pack_loc_Stop, 467
 - unpack_loc_BestAvailPos_Ind, 468
 - unpack_loc_DeleteAssistData, 468
 - unpack_loc_EngineState_Ind, 468
 - unpack_loc_EventRegister, 469
 - unpack_loc_PositionRpt_Ind, 469
 - unpack_loc_SLQSLOCGetBestAvailPos, 471
 - unpack_loc_SetExtPowerConfig_Ind, 469
 - unpack_loc_SetExtPowerState, 470
 - unpack_loc_SetOperationMode, 470
 - unpack_loc_Start, 471
 - unpack_loc_Stop, 471
- loc_BdsSV, 68
 - id, 68
 - mask, 68
- loc_BdsSVInfo, 68
 - len, 69
 - pSV, 69
- loc_CellDb, 69
 - mask, 69
- loc_ClkInfo, 70
 - mask, 71
- loc_GnssData, 71
 - mask, 72
- loc_LocApplicationInfo, 73
 - appNameLength, 74
 - appProviderLength, 74
 - appVersionLength, 74
 - appVersionValid, 74
 - pAppName, 74
 - pAppProvider, 74
 - pAppVersion, 74
- loc_SV, 76
 - id, 76
 - mask, 76
 - system, 76
- loc_SVInfo, 76
 - len, 77
 - pSV, 77
- loc_gpsTime, 72
 - gpsTimeOfWeekMs, 73
 - gpsWeek, 73
- loc_precisionDilution, 74
 - HDOP, 75
 - PDOP, 75
 - VDOP, 75
- loc_sensorDataUsage, 75
 - aidingIndicatorMask, 75
 - usageMask, 76
- loc_svUsedforFix, 77
 - gnssSvUsedList, 78
 - gnssSvUsedList_len, 78
- localTimeOffset
 - nas_qaQmi3Gpp2TimeZone, 133
- logString
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 286
- logger
 - common.h, 413
- longName
 - unpack_nas_SLQSGetPLMNName_t, 316

- longNameCi
 - unpack_nas_SLQSSetPLMNName_t, 316
- longNameEn
 - unpack_nas_SLQSSetPLMNName_t, 316
- longNameLen
 - unpack_nas_SLQSSetPLMNName_t, 316
- longNameSB
 - unpack_nas_SLQSSetPLMNName_t, 316
- loopbackMode
 - pack_wds_SLQSSetLoopback_t, 234
- loopbackMultiplier
 - pack_wds_SLQSSetLoopback_t, 234
- LteBandCapability
 - unpack_dms_SLQSSetBandCapability_t, 283
- lteEmmDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 198
- lteEmmUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 198
- lteEsmDI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 198
- lteEsmUI
 - pack_nas_SLQSNasSwiOTAMessageCallback_t, 198
- LteGsmCellInfo
 - nas_LTEInfoNeighboringGSM, 117
- LteQci
 - unpack_qos_swIQosFlow_t, 357
- lteRsrpDelta
 - nas_SLQSSignalStrengthsIndReq, 144
- lteRsrpinfo
 - nas_SLQSSignalStrengthsInformation, 145
- lteSSInfo, 78
 - rsrp, 78
 - rsrq, 78
 - rsqi, 78
 - snr, 78
- lteSnrDelta
 - nas_SLQSSignalStrengthsIndReq, 144
- lteSnrinfo
 - nas_SLQSSignalStrengthsInformation, 145
- ltersrp
 - unpack_nas_SLQSSetSignalStrength_t, 319
- ltesnr
 - unpack_nas_SLQSSetSignalStrength_t, 319
- m_FwBuildId
 - CarrierImage_t, 28
- m_FwImageld
 - CarrierImage_t, 28
- m_PriBuildId
 - CarrierImage_t, 28
- m_PrImageld
 - CarrierImage_t, 28
- m_nCarrierId
 - CarrierImage_t, 28
- m_nFolderId
 - CarrierImage_t, 28
- m_nStorage
 - CarrierImage_t, 28
- MAX_BUILD_ID_LEN
 - dms.h, 420
- MAX_ICCID_LENGTH
 - uim.h, 576
- MAX_MSE_TWS_MSG
 - sms.h, 555
- MAX_NO_OF_SLOTS
 - uim.h, 576
- MAX_SLOTS_STATUS
 - uim.h, 576
- MAX_SMS_LIST_SIZE
 - sms.h, 555
- MCC
 - nas_CDMASysInfo, 87
 - nas_CDMASysInfoExt, 88
 - nas_currentPLMN, 92
 - nas_GSMSysInfo, 104
 - nas_LTESysInfo, 123
 - nas_QmiNas3GppNetworkInfo, 134
 - nas_QmiNas3GppNetworkRAT, 135
 - nas_QmisNasPcsDigit, 136
 - nas_WCDMASysInfo, 163
 - unpack_nas_GetServingNetwork_t, 309
- MDMCallDuration
 - connectionStatus, 29
- MDMConnStatus
 - connectionStatus, 29
- MEIDString
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- MIN
 - unpack_dms_GetVoiceNumber_t, 275
- MINREQBKLEN
 - common.h, 413
- MMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 360
- MNC
 - nas_CDMASysInfo, 87
 - nas_currentPLMN, 92
 - nas_GSMSysInfo, 104
 - nas_LTESysInfo, 124
 - nas_QmiNas3GppNetworkInfo, 134
 - nas_QmiNas3GppNetworkRAT, 135
 - nas_QmisNasPcsDigit, 136
 - nas_WCDMASysInfo, 163
 - unpack_nas_GetServingNetwork_t, 309
- MPTlv
 - NASQmiCbkNasSystemSelPrefInd, 171
- MSGID_AND_LEN
 - common.h, 413
- MSGID_DONT_CARE
 - common.h, 413
- MTMessageInfo
 - newMTMessageTlv, 173
- manufacturer
 - unpack_dms_GetManufacturer_t, 271

- Mask
 - pack_wds_SLQSGetDUNCallInfo_t, 228
- mask
 - loc_BdsSV, 68
 - loc_CellDb, 69
 - loc_ClkInfo, 71
 - loc_GnssData, 72
 - loc_SV, 76
 - unpack_qos_IPv6TrafCls_t, 341
 - unpack_qos_Tos_t, 358
- max_channel_rx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 388
- max_channel_tx_rate
 - unpack_wds_SLQSGetCurrentChannelRate_t, 388
- max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, 210
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- max_dist_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- max_time
 - pack_swiloc_SwiLocSetAutoStart_t, 210
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- max_time_reported
 - unpack_swiloc_SwiLocGetAutoStart_t, 364
- MaxAllowedPktSz
 - unpack_qos_swiQosFlow_t, 357
- MaxChanRxRate
 - dunchannelRate, 34
- MaxChanTxRate
 - dunchannelRate, 34
- maxChannelRXRate
 - unpack_wds_GetConnectionRate_t, 377
- maxChannelTXRate
 - unpack_wds_GetConnectionRate_t, 377
- maxDIBitRate
 - LibPackQosClassID, 63
- maxDownlinkBitrate
 - LibPackUMTSQoS, 67
 - wds_UMTSMInQoS, 407
- maxImages
 - FMSImageIDEntries, 37
- MaxRXChannelRate
 - unpack_dms_GetDeviceCap_t, 265
- maxRxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, 266
- maxSDUSize
 - LibPackUMTSQoS, 67
 - wds_UMTSMInQoS, 407
- MaxTXChannelRate
 - unpack_dms_GetDeviceCap_t, 265
- maxTxChannelRate
 - unpack_dms_GetDeviceCapabilities_t, 266
- maxUIBitRate
 - LibPackQosClassID, 63
- maxUplinkBitrate
 - LibPackUMTSQoS, 67
- wds_UMTSMInQoS, 407
- mcc
 - nas_CSGID, 92
 - nas_MNRInfo, 125
 - nas_netSelectionPref, 126
 - pack_nas_SLQSGetPLMNName_t, 188
 - unpack_nas_GetHomeNetwork_t, 307
- mdmCallDurationActive
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- meanThroughputClass
 - LibPackGPRSRequestedQoS, 40
 - wds_GPRSQoS, 402
- meid
 - unpack_dms_GetSerialNumbers_t, 274
- meidSize
 - unpack_dms_GetDeviceSerialNumbers_t, 267
- message
 - unpack_sms_SLQSGetSMS_t, 361
- message_type
 - NASOTAMessageTlv, 166
- messageFailureCode
 - unpack_sms_SendSMS_t, 359
- messageFormat
 - pack_sms_SendSMS_t, 205
 - unpack_sms_SLQSGetSMS_t, 361
- messageID
 - unpack_sms_SendSMS_t, 359
- messageIndex
 - pack_sms_SLQSGetSMS_t, 207
 - pack_sms_SLQSMModifySMSStatus_t, 209
 - qmiSmsMessageList, 237
 - sMSMTMessage, 245
- messageList
 - unpack_sms_SLQSGetSMSList_t, 362
- messageListSize
 - unpack_sms_SLQSGetSMSList_t, 362
- messageMode
 - sMSMessageMode, 244
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 363
- MessageModelInfo
 - messageModeTlv, 79
- messageModeTlv, 78
 - MessageModelInfo, 79
 - TlvPresent, 79
- messageSize
 - pack_sms_SendSMS_t, 205
 - unpack_sms_SLQSGetSMS_t, 362
- messageTag
 - pack_sms_SLQSMModifySMSStatus_t, 209
 - qmiSmsMessageList, 237
 - unpack_sms_SLQSGetSMS_t, 362
- MinPolicedPktSz
 - unpack_qos_swiQosFlow_t, 357
- minSize
 - unpack_dms_GetVoiceNumber_t, 275
- minute
 - nas_timeInfo, 152

- nas_UniversalTime, 157
- mipMode
 - unpack_wds_GetMobileIP_t, 380
- mipStatus
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- mipstatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 394
- mnc
 - nas_CSGID, 92
 - nas_MNRInfo, 126
 - nas_netSelectionPref, 126
 - pack_nas_SLQSGetPLMNName_t, 188
 - unpack_nas_GetHomeNetwork_t, 307
- mncPcsDigits
 - nas_CSGID, 92
- mobileCountryCode
 - sMSEtwsPlmn, 244
- mobileIP
 - pack_wds_SLQSSetWdsEventCallback_t, 233
- mobileNetworkCode
 - sMSEtwsPlmn, 244
- mode
 - pack_dms_SetEventReport_t, 176
 - pack_dms_SetPower_t, 176
 - pack_loc_SetOperationMode_t, 184
- ModePref
 - NASModePreferenceTlv, 165
- modelid
 - unpack_dms_GetModelID_t, 271
- modelid_str
 - unpack_dms_GetFirmwareInfo_t, 268
- modemMode
 - nas_CommInfo, 91
- modemindex
 - pack_fms_SetImagesPreference_t, 180
- month
 - nas_timeInfo, 152
 - nas_UniversalTime, 157
- msgid
 - pack_qmi_t, 203
 - unpack_qmi_t, 339
- msgtype
 - common.h, 414
- Mtu
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- multiplier
 - unpack_qos_pktErrRate_t, 341
- NAI
 - unpack_wds_GetMobileIPProfile_t, 380
- NAS_PLMN_LENGTH
 - nas.h, 477
- NASBandPreferenceTlv, 163
 - band_pref, 163
 - TlvPresent, 163
- NASEmergencyModeTlv, 164
 - EmerMode, 164
- TlvPresent, 164
- NASGWAcqOrderPrefTlv, 164
 - GWAcqOrderPref, 164
 - TlvPresent, 164
- NASLTEBandPreferenceTlv, 165
 - LTEBandPref, 165
 - TlvPresent, 165
- NASLteNasReleaseInfoTlv, 165
 - nas_major, 165
 - nas_minor, 165
 - nas_release, 165
 - TlvPresent, 165
- NASModePreferenceTlv, 165
 - ModePref, 165
 - TlvPresent, 165
- NASNetSelPreferenceTlv, 165
 - NetSelPref, 166
 - TlvPresent, 166
- NASOTAMessageTlv, 166
 - data_buf, 166
 - data_len, 166
 - message_type, 166
 - TlvPresent, 166
- NASPRLPreferenceTlv, 170
 - PRLPref, 170
 - TlvPresent, 170
- NASPhyCaAggPcellInfo, 166
 - dl_bw_value, 167
 - freq, 167
 - iLTEbandValue, 167
 - pci, 167
 - TlvPresent, 167
- NASPhyCaAggScellIDIBw, 167
 - dl_bw_value, 167
 - TlvPresent, 167
- NASPhyCaAggScellIndType, 168
 - freq, 168
 - pci, 169
 - scell_state, 169
 - TlvPresent, 169
- NASPhyCaAggScellIndex, 167
 - scell_idx, 168
 - TlvPresent, 168
- NASPhyCaAggScellInfo, 169
 - dl_bw_value, 169
 - freq, 169
 - iLTEbandValue, 170
 - pci, 170
 - scell_state, 170
 - TlvPresent, 170
- NASQmiCbkNasSwtOTAMessageInd, 170
 - nasRelInfoTlv, 170
 - otaMsgTlv, 170
 - timeTlv, 170
- NASQmiCbkNasSystemSelPrefInd, 170
 - BPTlv, 171
 - EMTlv, 171
 - GWAOPTlv, 171

- LBPTlv, [171](#)
- MPTlv, [171](#)
- NSPTlv, [171](#)
- PRLPTlv, [171](#)
- RPTlv, [171](#)
- SDPTlv, [171](#)
- NASRoamPreferenceTlv, [171](#)
 - RoamPref, [171](#)
 - TlvPresent, [171](#)
- NASServDomainPrefTlv, [171](#)
 - SrvDomainPref, [171](#)
 - TlvPresent, [171](#)
- NASServingSystemInfo, [171](#)
 - csAttachState, [173](#)
 - hdrPersonality, [173](#)
 - psAttachState, [173](#)
 - radiInterfaceList, [173](#)
 - radiInterfaceNo, [173](#)
 - registrationState, [173](#)
 - selectedNetwork, [173](#)
- NASTimeInfoTlv, [173](#)
 - time, [173](#)
 - TlvPresent, [173](#)
- NSPTlv
 - NASQmiCbkNasSystemSelPrefInd, [171](#)
- NWQoSStatus
 - unpack_qos_SLQSQosGetNetworkStatus_t, [344](#)
- naiSize
 - unpack_wds_GetMobileIPProfile_t, [380](#)
- Name
 - unpack_nas_GetServingNetwork_t, [309](#)
- name
 - unpack_nas_GetHomeNetwork_t, [307](#)
 - unpack_wds_GetDefaultProfile_t, [378](#)
- nameSize
 - unpack_nas_GetServingNetwork_t, [309](#)
- namelength
 - unpack_omaDmFotaTlv_t, [338](#)
- namesize
 - unpack_wds_GetDefaultProfile_t, [378](#)
- nas.h
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_100, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_15, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_25, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_50, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_6, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_CA_BW_NRB_75, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE, [478](#)
 - eLIBPACK_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE, [478](#)
- nas.h, [472](#)
 - NAS_PLMN_LENGTH, [477](#)
 - pack_nas_GetACCOLC, [478](#)
 - pack_nas_GetANAAAAAuthenticationStatus, [479](#)
 - pack_nas_GetCDMANetworkParameters, [479](#)
 - pack_nas_GetHomeNetwork, [479](#)
 - pack_nas_GetNetworkPreference, [480](#)
 - pack_nas_GetRFInfo, [480](#)
 - pack_nas_GetServingNetwork, [480](#)
 - pack_nas_GetServingNetworkCapabilities, [480](#)
 - pack_nas_GetSignalStrengths, [481](#)
 - pack_nas_PerformNetworkScan, [481](#)
 - pack_nas_SLQSGetNetworkTime, [483](#)
 - pack_nas_SLQSGetPLMNName, [483](#)
 - pack_nas_SLQSGetServingSystem, [483](#)
 - pack_nas_SLQSGetSignalStrength, [483](#)
 - pack_nas_SLQSGetSysInfo, [484](#)
 - pack_nas_SLQSGetSysSelectionPref, [484](#)
 - pack_nas_SLQSInitiateNetworkRegistration, [484](#)
 - pack_nas_SLQSNasConfigSigInfo2, [485](#)
 - pack_nas_SLQSNasGetCellLocationInfo, [485](#)
 - pack_nas_SLQSNasGetSigInfo, [485](#)
 - pack_nas_SLQSNasIndicationRegisterExt, [486](#)
 - pack_nas_SLQSNasSwiModemStatus, [486](#)
 - pack_nas_SLQSNasSwiOTAMessageCallback, [486](#)
 - pack_nas_SLQSSetBandPreference, [487](#)
 - pack_nas_SLQSSetSignalStrengthsCallback, [487](#)
 - pack_nas_SLQSSetSysSelectionPref, [487](#)
 - pack_nas_SLQSSwiGetLteCQI, [488](#)
 - pack_nas_SLQSSwiGetLteSccRxInfo, [488](#)
 - pack_nas_SetACCOLC, [481](#)
 - pack_nas_SetLURRejectCallback, [482](#)
 - pack_nas_SetNetworkPreference, [482](#)
 - pack_nas_SetRFInfoCallback, [482](#)
 - pack_nas_SlqsGetLTECphyCAInfo, [482](#)
 - unpack_nas_GetACCOLC, [488](#)
 - unpack_nas_GetANAAAAAuthenticationStatus, [489](#)
 - unpack_nas_GetCDMANetworkParameters, [489](#)
 - unpack_nas_GetHomeNetwork, [489](#)
 - unpack_nas_GetNetworkPreference, [490](#)
 - unpack_nas_GetRFInfo, [490](#)
 - unpack_nas_GetServingNetwork, [490](#)
 - unpack_nas_GetServingNetworkCapabilities, [490](#)

- unpack_nas_GetSignalStrengths, 491
- unpack_nas_PerformNetworkScan, 491
- unpack_nas_SLQSGetNetworkTime, 493
- unpack_nas_SLQSGetPLMNName, 494
- unpack_nas_SLQSGetServingSystem, 494
- unpack_nas_SLQSGetSignalStrength, 494
- unpack_nas_SLQSGetSysInfo, 495
- unpack_nas_SLQSGetSysSelectionPref, 495
- unpack_nas_SLQSInitiateNetworkRegistration, 495
- unpack_nas_SLQSNasConfigSigInfo2, 495
- unpack_nas_SLQSNasGetCellLocationInfo, 496
- unpack_nas_SLQSNasGetSigInfo, 496
- unpack_nas_SLQSNasIndicationRegisterExt, 496
- unpack_nas_SLQSNasNetworkTimeCallback_ind, 497
- unpack_nas_SLQSNasSigInfoCallback_ind, 497
- unpack_nas_SLQSNasSwiModemStatus, 497
- unpack_nas_SLQSNasSwiOTAMessageCallback, 498
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind, 498
- unpack_nas_SLQSNasSysInfoCallback_ind, 498
- unpack_nas_SLQSSetBandPreference, 499
- unpack_nas_SLQSSetSignalStrengthsCallback, 499
- unpack_nas_SLQSSetSysSelectionPref, 499
- unpack_nas_SLQSSetSysSelectionPrefCallback_ind, 499
- unpack_nas_SLQSSwiGetLteCQI, 500
- unpack_nas_SLQSSwiGetLteScsRxInfo, 500
- unpack_nas_SetACCOLC, 491
- unpack_nas_SetDataCapabilitiesCallback_ind, 492
- unpack_nas_SetEventReportInd, 492
- unpack_nas_SetLURRejectCallback, 492
- unpack_nas_SetNasLTECphyCalIndCallback_ind, 492
- unpack_nas_SetNetworkPreference, 493
- unpack_nas_SetRFInfoCallback, 493
- unpack_nas_SetRoamingIndicatorCallback_ind, 493
- unpack_nas_SetServingSystemCallback_ind, 493
- unpack_nas_SlqsGetLTECphyCAInfo, 493
- nas_AddCDMASysInfo, 79
 - geoSysIdx, 80
 - regPrd, 80
- nas_AddSysInfo, 80
 - cellBroadcastCap, 80
 - geoSysIdx, 80
- nas_CDMAECIOThresh, 82
 - CDMAECIOThreshListLen, 82
 - pCDMAECIOThreshList, 82
- nas_CDMAInfo, 83
 - baseId, 83
 - baseLat, 83
 - baseLong, 83
 - nid, 83
 - refpn, 84
 - sid, 84
- nas_CDMARSSIThresh, 84
 - CDMARSSIThreshListLen, 84
 - pCDMARSSIThreshList, 84
- nas_CDMASysInfo, 84
 - baseId, 87
 - baseLat, 87
 - baseLong, 87
 - bsInfoValid, 87
 - bsPRev, 87
 - bsPRevValid, 87
 - ccsSupported, 87
 - ccsSupportedValid, 87
 - cdmaSysIdValid, 87
 - isSysPriMatch, 87
 - isSysPriMatchValid, 87
 - MCC, 87
 - MNC, 87
 - networkID, 87
 - networkIdValid, 87
 - pRevInUse, 88
 - pRevInUseValid, 88
 - packetZone, 88
 - packetZoneValid, 88
 - sysInfoCDMA, 88
 - systemID, 88
- nas_CDMASysInfoExt, 88
 - imsi_11_12, 88
 - MCC, 88
- nas_CSGID, 91
 - id, 91
 - mcc, 92
 - mnc, 92
 - mncPcsDigits, 92
 - rat, 92
- nas_CallBarringSysInfo, 80
 - csBarStatus, 81
 - psBarStatus, 81
- nas_CommInfo, 89
 - imsRegState, 91
 - modemMode, 91
 - psState, 91
 - systemMode, 91
 - temperature, 91
- nas_GERANInfo, 96
 - arfcn, 98
 - bsic, 98
 - cellID, 98
 - insNmrCellInfo, 98
 - lac, 98
 - nmrInst, 98
 - plmn, 98
 - rxLev, 98
 - timingAdvance, 98
- nas_GSMRSSIThresh, 100
 - GSMRSSIThreshListLen, 101
 - pGSMRSSIThreshList, 101

- nas_GSMSrvStatusInfo, 101
 - isPrefDataPath, 102
 - srvStatus, 102
 - trueSrvStatus, 102
- nas_GSMSysInfo, 102
 - cellId, 104
 - cellIdValid, 104
 - dtmSupp, 104
 - dtmSuppValid, 104
 - egprsSupp, 104
 - egprsSuppValid, 104
 - lac, 104
 - lacValid, 104
 - MCC, 104
 - MNC, 104
 - networkIdValid, 104
 - regRejectInfoValid, 104
 - rejCause, 104
 - rejectSrvDomain, 105
 - sysInfoGSM, 105
- nas_HDRECIOTresh, 105
 - HDRECIOTreshListLen, 105
 - pHDRECIOTreshList, 105
- nas_HDRIOTresh, 105
 - HDRIOTreshListLen, 106
 - pHDRIOTreshList, 106
- nas_HDRRSSITresh, 106
 - HDRRSSITreshListLen, 106
 - pHDRRSSITreshList, 106
- nas_HDRSINRThreshold, 106
 - HDRSINRThresholdListLen, 107
 - pHDRSINRThresholdList, 107
- nas_HDRSysInfo, 107
 - hdrActiveProt, 109
 - hdrActiveProtValid, 109
 - hdrPersonality, 109
 - hdrPersonalityValid, 109
 - is856SysId, 109
 - is856SysIdValid, 109
 - isSysPrIMatch, 109
 - isSysPrIMatchValid, 109
 - sysInfoHDR, 109
- nas_LTEInfo, 111
 - band, 113
 - bandwidth, 113
 - emmConnState, 113
 - emmState, 113
 - emmSubState, 113
 - RXChan, 113
 - TXChan, 113
- nas_LTEInfoInterfreq, 114
 - freqsLen, 114
 - InfoInterfreq, 114
 - ueInIdle, 114
- nas_LTEInfoIntrafreq, 114
 - CellParams, 116
 - cellReselPriority, 116
 - cellsLen, 116
 - earfcn, 116
 - globalCellId, 116
 - plmn, 116
 - sIntraSearch, 116
 - sNonIntraSearch, 116
 - servingCellId, 116
 - tac, 116
 - threshServingLow, 116
 - ueInIdle, 116
- nas_LTEInfoNeighboringGSM, 116
 - freqsLen, 117
 - LteGsmCellInfo, 117
 - ueInIdle, 117
- nas_LTEInfoNeighboringWCDMA, 117
 - freqsLen, 118
 - ueInIdle, 118
- nas_LTERSRPThresh, 118
 - LTERSRPThreshListLen, 118
 - pLTERSRPThreshList, 119
- nas_LTERSRQThresh, 119
 - LTERSRQThreshListLen, 119
 - pLTERSRQThreshList, 119
- nas_LTERSSITresh, 119
 - LTERSSITreshListLen, 120
 - pLTERSSITreshList, 120
- nas_LTESNRThreshold, 121
 - LTESNRThresholdListLen, 121
 - pLTESNRThresholdList, 121
- nas_LTESigRptConfig, 120
 - avgPeriod, 120
 - rptRate, 120
- nas_LTESysInfo, 121
 - cellId, 123
 - cellIdValid, 123
 - lac, 123
 - lacValid, 123
 - MCC, 123
 - MNC, 124
 - networkIdValid, 124
 - regRejectInfoValid, 124
 - rejCause, 124
 - rejectSrvDomain, 124
 - sysInfoLTE, 124
 - tac, 124
 - tacValid, 124
- nas_MNRInfo, 125
 - mcc, 125
 - mnc, 126
 - rat, 126
- nas_PhyCaAggPcellInfo, 128
 - dl_bw_value, 128
 - freq, 128
 - iLTEbandValue, 128
 - pci, 128
 - TlvPresent, 128
- nas_PhyCaAggScellIDBw, 128
 - dl_bw_value, 129
 - TlvPresent, 129

- nas_PhyCaAggScellIndType, 129
 - freq, 130
 - pci, 130
 - scell_state, 130
 - TlvPresent, 130
- nas_PhyCaAggScellIndex, 129
 - scell_idx, 129
 - TlvPresent, 129
- nas_PhyCaAggScellInfo, 130
 - dl_bw_value, 133
 - freq, 133
 - iLTEbandValue, 133
 - pci, 133
 - scell_state, 133
 - TlvPresent, 133
- nas_QmiNas3GppNetworkInfo, 134
 - Description, 134
 - Forbidden, 134
 - InUse, 134
 - MCC, 134
 - MNC, 134
 - Preferred, 134
 - Roaming, 134
- nas_QmiNas3GppNetworkRAT, 134
 - MCC, 135
 - MNC, 135
 - RAT, 135
- nas_QmisNasPcsDigit, 135
 - includes_pcs_digit, 135
 - MCC, 136
 - MNC, 136
- nas_RFInfoTlv, 136
 - activeBandClass, 136
 - activeChannel, 136
 - radioInterface, 137
 - radioInterfaceSize, 137
 - TlvPresent, 137
- nas_RejectReasonTlv, 136
 - rejectCause, 136
 - serviceDomain, 136
 - TlvPresent, 136
- nas_RxSigInfo, 138
 - isRadioTuned, 139
 - rsrp, 139
 - rxChainIndex, 139
 - rxPower, 139
- nas_SLQSSignalStrengthsIndReq, 143
 - ecioDelta, 144
 - ecioThresholdList, 144
 - ecioThresholdListLen, 144
 - ioDelta, 144
 - lteRsrpDelta, 144
 - lteSnrDelta, 144
 - rsrqDelta, 144
 - rxSignalStrengthDelta, 144
 - sinrDelta, 144
 - sinrThresholdList, 144
 - sinrThresholdListLen, 144
- nas_SLQSSignalStrengthsInformation, 144
 - ecioInfo, 145
 - errorRateInfo, 145
 - io, 145
 - lteRsrpinfo, 145
 - lteSnrinfo, 145
 - rsrqInfo, 145
 - rxSignalStrengthInfo, 145
 - sinr, 145
- nas_SLQSSignalStrengthsTlv, 145
 - sSLQSSignalStrengthsInfo, 145
 - TlvPresent, 145
- nas_SccRxInfo, 140
 - numInstances, 141
 - rsrq, 141
 - sigInfo, 141
 - snr, 141
 - TlvPresent, 141
- nas_SignalStrengthTlv, 143
 - radioInterface, 143
 - signalStrength, 143
 - TlvPresent, 143
- nas_SrvStatusInfo, 145
 - isPrefDataPath, 146
 - srvStatus, 146
- nas_TDSCDMAECIOThresh, 149
- nas_TDSCDMARSCPTthresh, 149
- nas_TDSCDMARSSIThresh, 150
- nas_TDSCDMASINRThresh, 150
- nas_UMTSInfo, 152
 - cellID, 154
 - ecio, 154
 - geranInst, 154
 - GeranInstInfo, 154
 - lac, 154
 - plmn, 154
 - psc, 154
 - rscp, 154
 - UMTSInstInfo, 154
 - uarfcn, 154
 - umtsInst, 154
- nas_UMTSinstInfo, 154
 - umtsEcio, 155
 - umtsPsc, 155
 - umtsRscp, 155
 - umtsUarfcn, 155
- nas_UniversalTime, 156
 - day, 157
 - dayOfWeek, 157
 - hour, 157
 - minute, 157
 - month, 157
 - second, 157
 - year, 157
- nas_WCDMAECIOThresh, 158
- nas_WCDMAInfoLTENeighborCell, 159
 - umtsLTENbrCellLen, 159
 - wcdmaRRState, 159

- nas_WCDMARSSIThresh, 159
- nas_WCDMASysInfo, 160
 - cellId, 163
 - cellIdValid, 163
 - hsCallStatus, 163
 - hsCallStatusValid, 163
 - hsInd, 163
 - hsIndValid, 163
 - lac, 163
 - lacValid, 163
 - MCC, 163
 - MNC, 163
 - networkIdValid, 163
 - psc, 163
 - pscValid, 163
 - regRejectInfoValid, 163
 - rejCause, 163
 - rejectSrvDomain, 163
 - sysInfoWCDMA, 163
- nas_acqOrderPref, 79
 - acqOrdeLen, 79
 - pAcqOrder, 79
- nas_callBarStatus, 81
 - csBarStatus, 82
 - psBarStatus, 82
- nas_cellParams, 88
 - pci, 89
 - rsrp, 89
 - rsrq, 89
 - rssi, 89
 - srxlev, 89
- nas_currentPLMN, 92
 - MCC, 92
 - MNC, 92
 - netDescr, 92
 - netDescrLength, 93
- nas_dataSrvCapabilities, 93
 - dataCapabilities, 93
 - dataCapabilitiesLen, 93
- nas_detailSvcInfo, 93
 - hdrHybrid, 95
 - hdrSrvStatus, 95
 - isSysForbidden, 95
 - srvCapability, 95
 - srvStatus, 95
- nas_ecioListElement, 95
 - ecio, 95
 - radioIf, 95
- nas_errorRateListElement, 96
 - errorRate, 96
 - radioIf, 96
- nas_geranInstInfo, 98
 - geranArfcn, 99
 - geranBsicBcc, 99
 - geranBsicNcc, 99
 - geranRssi, 99
- nas_gsmCellInfo, 99
 - arfcn, 100
 - band1900, 100
 - bsicId, 100
 - cellIdValid, 100
 - rssI, 100
 - srxlev, 100
- nas_infoInterFreq, 109
 - cell_resel_priority, 110
 - cellInterFreqParams, 110
 - cells_len, 110
 - earfcn, 110
 - threshXHigh, 110
 - threshXLow, 110
- nas_lteGsmCellInfo, 110
 - cellReselPriority, 111
 - cells_len, 111
 - GsmCellInfo, 111
 - nccPermitted, 111
 - threshGsmHigh, 111
 - threshGsmLow, 111
- nas_lteRsrpinformation, 118
 - rsrplevel, 118
- nas_lteSnrinformation, 120
 - snrlevel, 121
- nas_lteWcdmaCellInfo, 124
 - cellReselPriority, 125
 - cellsLen, 125
 - threshXhigh, 125
 - threshXlow, 125
 - uarfcn, 125
 - WCDMACellInfo, 125
- nas_major
 - NASLteNasReleaseInfoTlv, 165
- nas_minor
 - NASLteNasReleaseInfoTlv, 165
- nas_netSelectionPref, 126
 - mcc, 126
 - mnc, 126
 - netReg, 126
- nas_nmrCellInfo, 126
 - nmrArfcn, 127
 - nmrBsic, 127
 - nmrCellID, 127
 - nmrLac, 127
 - nmrPlmn, 127
 - nmrRxLev, 127
- nas_qaQmi3Gpp2TimeZone, 133
 - daylightSavings, 133
 - leapSeconds, 133
 - localTimeOffset, 133
- nas_release
 - NASLteNasReleaseInfoTlv, 165
- nas_roamIndList, 137
 - numInstances, 137
 - radioInterface, 137
 - roamIndicator, 137
- nas_rsrqInformation, 138
 - radioIf, 138
 - rsrq, 138

- nas_rxSignalStrengthListElement, 139
 - radioIcf, 140
 - rxSignalStrength, 140
- nas_servSystem, 141
 - csAttachState, 142
 - numRadioInterfaces, 142
 - psAttachState, 142
 - radioInterface, 142
 - regState, 142
 - selNetwork, 142
- nas_sysInfoCommon, 146
 - isSysForbidden, 148
 - isSysForbiddenValid, 148
 - roamStatus, 148
 - roamStatusValid, 148
 - srvCapability, 148
 - srvCapabilityValid, 148
 - srvDomain, 148
 - srvDomainValid, 148
- nas_timeInfo, 151
 - day, 152
 - dayLtSavingAdj, 152
 - dayOfWeek, 152
 - hour, 152
 - minute, 152
 - month, 152
 - radioInterface, 152
 - second, 152
 - timeZone, 152
 - TlvPresent, 152
 - year, 152
- nas_umtsLTENbrCell, 155
 - cellsIsTDD, 156
 - earfcn, 156
 - pci, 156
 - rsrp, 156
 - rsrq, 156
 - srxlev, 156
- nas_wcdmaCellInfo, 157
 - cpich_ecno, 158
 - cpich_rscp, 158
 - psc, 158
 - srxlev, 158
- NasGetLTECphyCaInfo, 164
 - PhyCaAggPcellInfo, 164
 - PhyCaAggScellIDBw, 164
 - PhyCaAggScellIndType, 164
 - PhyCaAggScellIndex, 164
 - PhyCaAggScellInfo, 164
- nasRelInfoTlv
 - NASQmiCbkNasSwiOTAMessageInd, 170
- nccPermitted
 - nas_lteGsmCellInfo, 111
- netDescr
 - nas_currentPLMN, 92
- netDescrLength
 - nas_currentPLMN, 93
- netInfoLen
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- netReg
 - nas_netSelectionPref, 126
- NetSelPref
 - NASNetSelPreferenceTlv, 166
- NetworkID
 - unpack_nas_SLQSGetServingSystem_t, 318
- networkID
 - nas_CDMASysInfo, 87
- networkIdValid
 - nas_CDMASysInfo, 87
 - nas_GSMsysInfo, 104
 - nas_LTESysInfo, 124
 - nas_WCDMASysInfo, 163
- networkInfoLen
 - unpack_wds_SLQSGetCurrDataSystemStat_t, 387
- NetworkType
 - currNetworkInfo, 29
 - wds_currNetworkInfo, 400
- NewMMTlv
 - unpack_sms_SetNewSMSCallback_ind_t, 360
- newMTMessageTlv, 173
 - MTMessageInfo, 173
 - TlvPresent, 174
- newPINLen
 - uim_unblockUIMPIN, 262
- newPINVal
 - uim_unblockUIMPIN, 262
- nextHeader
 - LibPackTFTIDParams, 65
- nid
 - nas_CDMAInfo, 83
 - unpack_nas_GetHomeNetwork_t, 307
- nmrArfcn
 - nas_nmrCellInfo, 127
- nmrBsic
 - nas_nmrCellInfo, 127
- nmrCellID
 - nas_nmrCellInfo, 127
- nmrInst
 - nas_GERANInfo, 98
- nmrLac
 - nas_nmrCellInfo, 127
- nmrPlmn
 - nas_nmrCellInfo, 127
- nmrRxLev
 - nas_nmrCellInfo, 127
- notification
 - unpack_omaDmNotificationsTlv_t, 339
- notificationType
 - sMSEtwsMessage, 243
- notused
 - unpack_dms_SetCrashAction_t, 275
- num_instances
 - DMScustSettingList, 32
- numApp

- slotInf, [241](#)
- uim_slotInfo, [260](#)
- numEntries
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [285](#)
- NumFlows
 - unpack_qos_SLQSSetQosEventCallback_ind_t, [347](#)
- numInstances
 - nas_roamIndList, [137](#)
 - nas_SccRxInfo, [141](#)
 - wds_DomainNameList, [401](#)
 - wds_PCSCFFQDNAddressList, [404](#)
 - wds_PCSCFIPv4ServerAddressList, [404](#)
- numOpt
 - wdsDhcpv4OptionList, [409](#)
- numQosFlow
 - unpack_qos_SLQSQosSwiReadDataStats_t, [347](#)
- numRadioInterfaces
 - nas_servSystem, [142](#)
- NumRxFilters
 - unpack_qos_QosFlowInfo_t, [343](#)
- numSlot
 - uim_cardStatus, [252](#)
- NumSupUSBComps
 - unpack_dms_GetUSBComp_t, [274](#)
- NumTxFilters
 - unpack_qos_QosFlowInfo_t, [343](#)
- NxtHdrProto
 - unpack_qos_swiQosFilter_t, [353](#)
- OMADMEEnabled
 - unpack_swioma_SLQSOMADMGetSettings_t, [369](#)
- OfflineReason
 - unpack_dms_GetPower_t, [273](#)
- offset
 - uim_readTransparentInfo, [256](#)
- oldPINLen
 - uim_changeUIMPIN, [253](#)
- oldPINVal
 - uim_changeUIMPIN, [253](#)
- operatingMode
 - dms_OperatingModeTlv, [30](#)
- OperatingModeTlv
 - unpack_dms_SetEventReport_ind_t, [276](#)
- OperationMode
 - unpack_dms_GetPower_t, [273](#)
- optCode
 - wdsDhcpv4Option, [409](#)
- optVal
 - wdsDhcpv4Option, [409](#)
- optValLen
 - wdsDhcpv4Option, [409](#)
- otaMsgTlv
 - NASQmiCbkNasSwiOTAMessageInd, [170](#)
- p3GPP2TimeInfo
 - unpack_nas_SLQSGetNetworkTime_t, [315](#)
- p3GPPTIMEInfo
 - unpack_nas_SLQSGetNetworkTime_t, [315](#)
- p3GppNetworkInfoInstances
 - unpack_nas_PerformNetworkScan_t, [311](#)
- p3GppNetworkInstanceSize
 - unpack_nas_PerformNetworkScan_t, [311](#)
- p3gppRelease
 - pack_wds_SLQSSet3GPPConfigItem_t, [231](#)
- pAAASPI
 - pack_wds_SetMobileIPProfile_t, [226](#)
- PACK_WDS_IPV4
 - wds.h, [589](#)
- PACK_WDS_IPV6
 - wds.h, [589](#)
- pAPNClass
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [55](#)
- pAPNClass3GPP2
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [61](#)
- pAPNDisabledFlag
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [55](#)
- pAPNEnabled3GPP2
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [61](#)
- pAPNName
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [55](#)
- pAPNnameSize
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [55](#)
- pAcqOrder
 - nas_acqOrderPref, [79](#)
- pAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, [202](#)
- pAddCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pAddGSMSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pAddHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pAddLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pAddWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pAddrAllocPref
 - LibpackProfile3GPP, [44](#)
 - LibPackprofile_3GPP, [55](#)
- pAddress
 - pack_wds_SetMobileIPProfile_t, [226](#)
- pAllowLinger
 - LibpackProfile3GPP2, [50](#)
 - LibPackprofile_3GPP2, [61](#)
- pAltitudeAssumed

- unpack_loc_PositionRpt_Ind_t, 301
- pAltitudeWrtEllipsoid
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 301
- pAltitudeWrtMeanSeaLevel
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 301
- pApnString
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pApnStringSize
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pApnname
 - pack_wds_SetDefaultProfile_t, 224
- pAppName
 - loc_LocApplicationInfo, 74
- pAppPriority
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pAppProvider
 - loc_LocApplicationInfo, 74
- pAppType
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pAppVersion
 - loc_LocApplicationInfo, 74
- pApplicationInfo
 - pack_loc_Start_t, 186
- pAuth
 - pack_wds_SLQSSStartDataSession_t, 234
- pAuthPassword
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pAuthPassword_tSize
 - LibPackprofile_3GPP2, 61
- pAuthPasswordSize
 - LibpackProfile3GPP2, 50
- pAuthProtocol
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pAuthRetryCount
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pAuthTimeout
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pAuthenticationPref
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pAutosdm
 - pack_swima_SLQSOMADMSetSettings_t, 213
- pBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 202
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pBdsSVInfo
 - pack_loc_Delete_Assist_Data_t, 180
- pCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 192
- pCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pCDMAECIOThreshList
 - nas_CDMAECIOThresh, 82
- pCDMAInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 327
- pCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pCDMARSSIThreshList
 - nas_CDMARSSIThresh, 84
- pCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 330
- pCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- PCSCFAddrPCO
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- PCSCFFQDNAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- pCSGID
 - pack_nas_SLQSSetSysSelectionPref_t, 202
- pCardResult
 - unpack_uim_ReadTransparent_t, 372
- pCardStatus
 - unpack_uim_GetCardStatus_t, 371
 - unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 375
- pCellDb
 - pack_loc_Delete_Assist_Data_t, 180
- pChangeDuration
 - pack_nas_SLQSInitiateNetworkRegistration_t, 189
- pChgDuration
 - pack_nas_SLQSSetSysSelectionPref_t, 202
- pClkInfo
 - pack_loc_Delete_Assist_Data_t, 180
- pConfigAltitudeAssumed
 - pack_loc_Start_t, 186
- pCreateProfileOut
 - unpack_wds_SLQSCreateProfile_t, 384
- pCurProfile
 - pack_wds_SLQSCreateProfile_t, 227
- pCurrImgInfo
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 285
- pCustSettingInfo
 - DMSgetCustomFeatureV2, 33
- pCustSettingList
 - DMSgetCustomFeatureV2, 33
- pDDTMInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- PDOP
 - loc_precisionDilution, 75
- PDPTType

- unpack_wds_SLQSGetRuntimeSettings_t, 391
- pDataMode
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pDataRate
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pDayltSavAdj
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 329
- pDefaultPDNEnabled
 - pack_wds_SLQSSet3GPPConfigItem_t, 232
- pDeferTime
 - pack_swioama_SLQSOMADMSendSelection_t, 212
- pDestSMSContent
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, 178
 - packgetDyingGaspCfg, 236
- pDestSMSNum
 - pack_dms_SLQSSwiSetDyingGaspCfg_t, 178
 - packgetDyingGaspCfg, 236
- pDualStandByPrefInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pEmerMode
 - pack_nas_SLQSSetSysSelectionPref_t, 202
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pEnabled
 - pack_wds_SetMobileIPPProfile_t, 226
- pEncryptData
 - pack_uim_ReadTransparent_t, 215
- pEncryptedData
 - unpack_uim_ReadTransparent_t, 372
- pEncryptedPIN1
 - pack_uim_VerifyPin_t, 221
 - unpack_uim_ChangePin_t, 370
 - unpack_uim_SetPinProtection_t, 373
 - unpack_uim_UnblockPin_t, 375
 - unpack_uim_VerifyPin_t, 376
- pErrorRateInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pExtErrCode
 - UnPackGetProfileSettingOut, 398
- pExtErrorCode
 - unpack_wds_SLQSModifyProfile_t, 392
- pFailureReason
 - unpack_wds_SLQSStartDataSession_t, 396
- pFixId
 - unpack_loc_PositionRpt_Ind_t, 301
- pFwAutoCheck
 - pack_swioama_SLQSOMADMSetSettings_t, 213
- pGERANInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 327
- pGPRSMinimumQoS
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pGPRSRequestedQoS
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- pGSMCallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pGSMCipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pGSMRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pGSMRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pGSMRSSIThreshList
 - nas_GSMRSSIThresh, 101
- pGSMSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 330
- pGSMSSrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pGSMSSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pGWAcqOrderPref
 - pack_nas_SLQSSetSysSelectionPref_t, 202
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pGetCustomInput
 - DMSgetCustomFeatureV2, 33
- pGetDyingGaspCfg
 - unpack_dms_SLQSSwiGetDyingGaspCfg_t, 284
- pGetDyingGaspStatistics
 - unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 284
- pGnssData
 - pack_loc_Delete_Assist_Data_t, 181
- pGpsTime
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 301
- pHASPI
 - pack_wds_SetMobileIPPProfile_t, 226
- pHDRECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHDRECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHDRECIOThreshList
 - nas_HDRECIOThresh, 105
- pHDRIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHDRIOTThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHDRIOTThreshList
 - nas_HDRIOTThresh, 106
- pHDRNewUATIAssInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pHRRSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHRRSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHRRSSIThreshList
 - nas_HDRSSIThresh, 106
- pHRSINRDelta

- pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHRSINRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pHRSINRThreshList
 - nas_HRSINRThreshold, 107
- pHDRSessionCloseInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pHDRSigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 330
- pHDRSrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pHDRSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pHeading
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 301
- pHeadingUnc
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorCirConf
 - unpack_loc_BestAvailPos_Ind_t, 294
- pHorConfidence
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorEllpConf
 - unpack_loc_BestAvailPos_Ind_t, 294
- pHorReliability
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorUncCircular
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorUncEllipseOrientAzimuth
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorUncEllipseSemiMajor
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorUncEllipseSemiMinor
 - unpack_loc_BestAvailPos_Ind_t, 294
 - unpack_loc_PositionRpt_Ind_t, 302
- pHorizontalAccuracyLvl
 - pack_loc_Start_t, 186
- pHotSwapStatus
 - unpack_uim_GetCardStatus_t, 371
- pHwConfig
 - unpack_wds_SLQSSetDHCPv4ClientConfig_t, 395
- pIPv4AddrPref
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pIPv6AddPref
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- plgnoreHotSwapSwitch
 - pack_uim_SLQSUIMPowerUp_t, 218
- plmCnFlag
 - LibpackProfile3GPP, 44
 - LibPackprofile_3GPP, 56
- plmageList
 - pack_fms_SetImagesPreference_t, 180
 - unpack_fms_GetImagesPreference_t, 288
- plndicationToken
 - pack_uim_ChangePin_t, 214
 - pack_uim_ReadTransparent_t, 215
 - pack_uim_SetPinProtection_t, 216
 - pack_uim_UnblockPin_t, 220
 - pack_uim_VerifyPin_t, 221
 - unpack_uim_ChangePin_t, 370
 - unpack_uim_ReadTransparent_t, 372
 - unpack_uim_SetPinProtection_t, 373
 - unpack_uim_UnblockPin_t, 375
 - unpack_uim_VerifyPin_t, 376
- plIntermediateReportState
 - pack_loc_Start_t, 186
- plpcpAckTimeout
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- plpcpCreqRetryCount
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- plsPcscfAddressNedded
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pKeyReferenceID
 - pack_uim_ChangePin_t, 214
 - pack_uim_SetPinProtection_t, 217
 - pack_uim_UnblockPin_t, 220
 - pack_uim_VerifyPin_t, 221
- plTEAttachProfile
 - pack_wds_SLQSSet3GPPConfigItem_t, 232
- plTEAttachProfileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 232
- plTEBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, 202
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- plTECphyCa
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- plTEInfo
 - unpack_nas_SLQSNasSwiModemStatus_t, 330
- plTEInfoInterfreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 327
- plTEInfoIntrafreq
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 327
- plTEInfoNeighboringGSM
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 327
- plTEInfoNeighboringWCDMA
 - unpack_nas_SLQSNasGetCellLocationInfo_t, 327
- plTERSRPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- plTERSRPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- plTERSRPThreshList
 - nas_LTERSRPThresh, 119
- plTERSQRDelta

- pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTERSRQThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTERSRQThreshList
 - nas_LTERSRQThresh, 119
- pLTERSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTERSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTERSSIThreshList
 - nas_LTERSSIThresh, 120
- pLTESNRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTESNRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTESNRThreshList
 - nas_LTESNRThreshold, 121
- pLTESigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 330
- pLTESigRptConfig
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pLTESrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pLTESysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pLTVoiceSupportSysInfo
 - unpack_nas_SLQSGetSysInfo_t, 322
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pLatitude
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pLcpAckTimeout
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pLcpCreqRetryCount
 - LibpackProfile3GPP2, 50
 - LibPackprofile_3GPP2, 61
- pLeapSeconds
 - unpack_loc_PositionRpt_Ind_t, 302
- pLinktimer
 - pack_sms_SendSMS_t, 205
- pLongitude
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pMNAAs
 - pack_wds_SetMobileIPProfile_t, 226
- pMNCIncPCSDigStat
 - pack_nas_SLQSSetSysSelectionPref_t, 203
- pMNHA
 - pack_wds_SetMobileIPProfile_t, 226
- pMNRInfo
 - pack_nas_SLQSInitiateNetworkRegistration_t, 189
- pMagneticDeviation
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pManagedRoamingInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pMessage
 - pack_sms_SendSMS_t, 205
- pMessageIndex
 - pack_sms_SLQSDeleteSMS_t, 206
- pMessageMode
 - pack_sms_SLQSDeleteSMS_t, 207
 - pack_sms_SLQSGetSMS_t, 207
 - pack_sms_SLQSGetSMSList_t, 208
 - pack_sms_SLQSModifySMSStatus_t, 209
- pMessageTag
 - pack_sms_SLQSDeleteSMS_t, 207
- pMinIntervalTime
 - pack_loc_Start_t, 186
- pMncPcsDigitStatus
 - pack_nas_SLQSInitiateNetworkRegistration_t, 189
- pMncPcsStatus
 - pack_nas_SLQSGetPLMNName_t, 188
- pModePref
 - pack_nas_SLQSSetSysSelectionPref_t, 203
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pNAI
 - pack_wds_SetMobileIPProfile_t, 226
- pName
 - pack_wds_SetDefaultProfile_t, 224
- pNegoDnsSrvrPref
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pNetSelPref
 - pack_nas_SLQSSetSysSelectionPref_t, 203
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pNetworkTimeInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pNumberOfPhySlot
 - unpack_uim_SLQSUIMGetSlotsStatus_t, 374
- pOptList
 - wdsDhcpv4OptionList, 409
- pPcSInstance
 - unpack_nas_PerformNetworkScan_t, 311
- pPcSInstanceSize
 - unpack_nas_PerformNetworkScan_t, 311
- pPDNInactivTimeout
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPDNInactivTimeout3GPP2
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pPDType
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPRLPref
 - pack_nas_SLQSSetSysSelectionPref_t, 203
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pPass
 - pack_wds_SLQSStartDataSession_t, 235
- pPassword
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56

- pack_wds_SetDefaultProfile_t, 224
- pPasswordSize
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPcscfAddrUsingDhcp
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPcscfAddrUsingPCO
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPdnType
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pPdpAccessConFlag
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPdpContext
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPdpDataCompType
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPdpHdrCompType
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPppSessCloseTimer1x
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pPppSessCloseTimerDO
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pPrecisionDilution
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pPriDNSIPv4AddPref
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPriDNSIPv6addpref
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPriV6DnsAddress
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pPrimaryHA
 - pack_wds_SetMobileIPProfile_t, 226
- pPrimaryID
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pPrimaryV4DnsAddress
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pProfileID
 - unpack_wds_SLQSCreateProfile_t, 384
- pProfileId
 - pack_wds_SLQSCreateProfile_t, 227
 - pack_wds_SLQSModifyProfile_t, 230
 - pack_wds_SLQSSetDHCPv4ClientConfig_t, 233
- pProfileList
 - pack_wds_SLQSSet3GPPConfigItem_t, 232
- pProfileSettings
 - unpack_wds_SLQSGetProfileSettings_t, 389
- pProfileType
 - pack_wds_SLQSCreateProfile_t, 227
 - pack_wds_SLQSModifyProfile_t, 230
- pProfilename
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pProfilenameSize
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pQosClassID
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pRAT
 - pack_nas_SLQSSetSysSelectionPref_t, 203
- pRATInstance
 - unpack_nas_PerformNetworkScan_t, 311
- pRATInstanceSize
 - unpack_nas_PerformNetworkScan_t, 311
- pRATType
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- PRIStrng
 - unpack_dms_GetFirmwareRevision_t, 269
 - unpack_dms_GetFirmwareRevisions_t, 269
- PRLInd
 - unpack_nas_SLQSGetServingSystem_t, 318
- PRLPTlv
 - NASQmiCbkNasSystemSelPrefInd, 171
- PRLPref
 - NASPRLPreferenceTlv, 170
- pRXDroppedCount
 - unpack_wds_GetPacketStatistics_t, 382
- pRXOKBytesLastCall
 - unpack_wds_GetPacketStatistics_t, 382
- pRXOkBytesCount
 - unpack_wds_GetPacketStatistics_t, 382
- pRXPacketErrors
 - unpack_wds_GetPacketStatistics_t, 382
- pRXPacketOverflows
 - unpack_wds_GetPacketStatistics_t, 382
- pRXPacketSuccesses
 - unpack_wds_GetPacketStatistics_t, 382
- pRXTotalBytes
 - unpack_wds_GetByteTotals_t, 376
- pRadioInterface
 - unpack_nas_SLQSNasNetworkTimeCallback_ind-
_t, 329
- pRankIndicatorInd
 - pack_nas_SLQSNasSwiOTAMessageCallback_t,
198
- pReadResult
 - unpack_uim_ReadTransparent_t, 372
- pReasonMask
 - unpack_dms_GetOfflineReason_t, 273
- pRecurrenceType

- pack_loc_Start_t, 186
- pRejectReason
 - pack_swima_SLQSOMADMSendSelection_t, 212
- pRemainingRetries
 - unpack_uim_ChangePin_t, 370
 - unpack_uim_SetPinProtection_t, 373
 - unpack_uim_UnblockPin_t, 375
 - unpack_uim_VerifyPin_t, 376
- pReportChannelRate
 - pack_wds_SLQSGetDUNCallInfo_t, 228
- pReportConnStatus
 - pack_wds_SLQSGetDUNCallInfo_t, 228
- pReportDataBearerTech
 - pack_wds_SLQSGetDUNCallInfo_t, 228
- pReportDormStatus
 - pack_wds_SLQSGetDUNCallInfo_t, 228
- pReqSettings
 - pack_wds_SLQSGetRuntimeSettings_t, 229
- pRequestOptionList
 - unpack_wds_SLQSSetDHCPv4ClientConfig_t, 395
- pRequestedTag
 - pack_sms_SLQSGetSMSList_t, 208
- pRevInUse
 - nas_CDMA SysInfo, 88
- pRevInUseValid
 - nas_CDMA SysInfo, 88
- pRevTunneling
 - pack_wds_SetMobileIPProfile_t, 226
- pRoamPref
 - pack_nas_SLQSSetSysSelectionPref_t, 203
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pRscp
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, 330
- PSDomain
 - unpack_nas_GetServingNetwork_t, 309
- pSMSAttemptedFlag
 - packgetDyingGaspStatistics, 237
- pSV
 - loc_BdsSVInfo, 69
 - loc_SVInfo, 77
- pSVInfo
 - pack_loc_Delete_Assist_Data_t, 181
- pScanResult
 - unpack_nas_PerformNetworkScan_t, 311
- pScCRxInfo
 - unpack_nas_SLQSSwiGetLteScCRxInfo_t, 333
- pSecDNSIPv4AddPref
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pSecDNSIPv6addpref
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pSecV6DnsAddress
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pSecondaryFlag
 - LibpackProfile3GPP, 45
 - LibPackprofile_3GPP, 56
- pSecondaryHA
 - pack_wds_SetMobileIPProfile_t, 226
- pSecondaryV4DnsAddress
 - LibpackProfile3GPP2, 51
 - LibPackprofile_3GPP2, 62
- pSensorDataUsage
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pServingSystemInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pSigIndReq
 - pack_nas_SLQSSetSignalStrengthsCallback_t, 198
- pSignalStrengthInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pSourceIP
 - LibPackTFTIDParams, 65
- pSpeedHorizontal
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pSpeedUnc
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pSpeedVertical
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pSpeedVerticalUnc
 - unpack_loc_BestAvailPos_Ind_t, 295
- pSrvDomainPref
 - pack_nas_SLQSSetSysSelectionPref_t, 203
 - unpack_nas_SLQSGetSysSelectionPref_t, 326
- pSrvRegRestriction
 - pack_nas_SLQSSetSysSelectionPref_t, 203
- pStatMask
 - pack_wds_GetPacketStatistics_t, 223
- pSubscriptionInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pSvUsedforFix
 - unpack_loc_BestAvailPos_Ind_t, 295
 - unpack_loc_PositionRpt_Ind_t, 302
- pSysInfoInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 196
- pSysInfoNoChange
 - unpack_nas_SLQSSysInfoCallback_ind_t, 335
- pSystemSelectionInd
 - pack_nas_SLQSNasIndicationRegisterExt_t, 197
- pTDSCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pTDSCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pTDSCDMAECIOThreshList
 - nas_TDSCDMAECIOThresh, 149
- pTDSCDMARSCPDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, 193
- pTDSCDMARSCPThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, 193

- pTDSCDMARSCPThreshList
 - nas_TDSCDMARSCPThresh, [149](#)
- pTDSCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [193](#)
- pTDSCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [193](#)
- pTDSCDMARSSIThreshList
 - nas_TDSCDMARSSIThresh, [150](#)
- pTDSCDMASINRDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [194](#)
- pTDSCDMASINRThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [194](#)
- pTDSCDMASINRThreshList
 - nas_TDSCDMASINRThresh, [150](#)
- pTDSCDMASigInfoExt
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, [330](#)
- pTFTID1Params
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [56](#)
- pTFTID2Params
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [56](#)
- pTXDroppedCount
 - unpack_wds_GetPacketStatistics_t, [382](#)
- pTXOKBytesLastCall
 - unpack_wds_GetPacketStatistics_t, [382](#)
- pTXOKBytesCount
 - unpack_wds_GetPacketStatistics_t, [382](#)
- pTXPacketErrors
 - unpack_wds_GetPacketStatistics_t, [382](#)
- pTXPacketOverflows
 - unpack_wds_GetPacketStatistics_t, [382](#)
- pTXPacketSuccesses
 - unpack_wds_GetPacketStatistics_t, [382](#)
- pTXTotalBytes
 - unpack_wds_GetByteTotals_t, [376](#)
- pTdsdmaBandPref
 - pack_nas_SLQSSetSysSelectionPref_t, [203](#)
- pTech
 - pack_wds_SLQSStartDataSession_t, [235](#)
- pTechnologyMask
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pTimeSrc
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pTimeStamp
 - packgetDyingGaspStatistics, [237](#)
- pTimeUnc
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pTimeZone
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, [329](#)
- pTimestampUtc
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pTransferStatInd
 - pack_wds_SLQSGetDUNCallInfo_t, [228](#)
- pUMTSCellID
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [327](#)
- pUMTSInfo
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [328](#)
- pUMTSMInQoS
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [56](#)
- pUMTSMInQoSSigInd
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUMTSReqQoS
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUMTSReqQoSSigInd
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
- pUimSlotsStatus
 - unpack_uim_SLQSUIMGetSlotsStatus_t, [374](#)
- pUser
 - pack_wds_SLQSStartDataSession_t, [235](#)
- pUserId
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [62](#)
- pUserIdSize
 - LibpackProfile3GPP2, [51](#)
 - LibPackprofile_3GPP2, [62](#)
- pUsername
 - LibpackProfile3GPP, [45](#)
 - LibPackprofile_3GPP, [57](#)
 - pack_wds_SetDefaultProfile_t, [224](#)
- pUsernameSize
 - LibpackProfile3GPP, [46](#)
 - LibPackprofile_3GPP, [57](#)
- pVerboseFailReasonType
 - unpack_wds_SLQSStartDataSession_t, [396](#)
- pVerboseFailureReason
 - unpack_wds_SLQSStartDataSession_t, [396](#)
- pVertConfidence
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pVertReliability
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pVertUnc
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
 - unpack_loc_PositionRpt_Ind_t, [302](#)
- pWCDMACallBarringSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pWCDMACipherDomainSysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pWCDMAECIODelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [194](#)
- pWCDMAECIOThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [194](#)
- pWCDMAECIOThreshList

- nas_WCDMAECIOThresh, [158](#)
- pWCDMAInfoLTENeighborCell
 - unpack_nas_SLQSNasGetCellLocationInfo_t, [328](#)
- pWCDMARSSIDelta
 - pack_nas_SLQSNasConfigSigInfo2_t, [194](#)
- pWCDMARSSIThresh
 - pack_nas_SLQSNasConfigSigInfo2_t, [194](#)
- pWCDMARSSIThreshList
 - nas_WCDMARSSIThresh, [160](#)
- pWCDMASigInfo
 - unpack_nas_SLQSNasSigInfoCallback_ind_t, [330](#)
- pWCDMASrvStatusInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pWCDMASysInfo
 - unpack_nas_SLQSGetSysInfo_t, [322](#)
 - unpack_nas_SLQSSysInfoCallback_ind_t, [335](#)
- pXid
 - unpack_loc_BestAvailPos_Ind_t, [295](#)
- pack_dms_GetActivationState
 - dms.h, [421](#)
- pack_dms_GetBandCapability
 - dms.h, [421](#)
- pack_dms_GetCrashAction
 - dms.h, [421](#)
- pack_dms_GetCustFeature
 - dms.h, [422](#)
- pack_dms_GetCustFeaturesV2
 - dms.h, [422](#)
- pack_dms_GetCustFeaturesV2_t, [174](#)
 - cust_id, [174](#)
 - list_type, [174](#)
 - Tlvresult, [174](#)
- pack_dms_GetDeviceCap
 - dms.h, [422](#)
- pack_dms_GetDeviceCapabilities
 - dms.h, [422](#)
- pack_dms_GetDeviceHardwareRev
 - dms.h, [423](#)
- pack_dms_GetDeviceMfr
 - dms.h, [423](#)
- pack_dms_GetDeviceSerialNumbers
 - dms.h, [423](#)
- pack_dms_GetFSN
 - dms.h, [425](#)
- pack_dms_GetFirmwareInfo
 - dms.h, [424](#)
- pack_dms_GetFirmwareRevision
 - dms.h, [424](#)
- pack_dms_GetFirmwareRevisions
 - dms.h, [425](#)
- pack_dms_GetHardwareRevision
 - dms.h, [425](#)
- pack_dms_GetIMSI
 - dms.h, [426](#)
- pack_dms_GetManufacturer
 - dms.h, [426](#)
- pack_dms_GetModelID
 - dms.h, [426](#)
- pack_dms_GetNetworkTime
 - dms.h, [427](#)
- pack_dms_GetOfflineReason
 - dms.h, [427](#)
- pack_dms_GetPRLVersion
 - dms.h, [428](#)
- pack_dms_GetPower
 - dms.h, [427](#)
- pack_dms_GetSerialNumbers
 - dms.h, [428](#)
- pack_dms_GetUSBComp
 - dms.h, [428](#)
- pack_dms_GetVoiceNumber
 - dms.h, [429](#)
- pack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [432](#)
- pack_dms_SLQSDmsSwiIndicationRegister
 - dms.h, [432](#)
- pack_dms_SLQSDmsSwiIndicationRegister_t, [177](#)
 - resetInfoInd, [177](#)
- pack_dms_SLQSGetBandCapability
 - dms.h, [433](#)
- pack_dms_SLQSSwiClearDyingGaspStatistics
 - dms.h, [433](#)
- pack_dms_SLQSSwiGetDyingGaspCfg
 - dms.h, [434](#)
- pack_dms_SLQSSwiGetDyingGaspStatistics
 - dms.h, [434](#)
- pack_dms_SLQSSwiGetFirmwareCurr
 - dms.h, [434](#)
- pack_dms_SLQSSwiGetFwUpdateStatus
 - dms.h, [435](#)
- pack_dms_SLQSSwiSetDyingGaspCfg
 - dms.h, [435](#)
- pack_dms_SLQSSwiSetDyingGaspCfg_t, [177](#)
 - pDestSMSContent, [178](#)
 - pDestSMSNum, [178](#)
- pack_dms_SetCrashAction
 - dms.h, [429](#)
- pack_dms_SetCrashAction_t, [174](#)
 - crashAction, [175](#)
- pack_dms_SetCustFeature
 - dms.h, [430](#)
- pack_dms_SetCustFeature_t, [175](#)
 - DHCPRelayEnabled, [175](#)
 - DisableIMSI, [175](#)
 - GPSPMP, [175](#)
 - GPSSel, [175](#)
 - GpsEnable, [175](#)
 - IPFamSupport, [175](#)
 - IsVoiceEnabled, [175](#)
 - RMAutoConnect, [175](#)
 - SMSSupport, [175](#)
- pack_dms_SetCustFeaturesV2
 - dms.h, [430](#)
- pack_dms_SetCustFeaturesV2_t, [175](#)
 - cust_id, [176](#)

- cust_value, 176
 - Tlvresult, 176
 - value_length, 176
- pack_dms_SetEventReport
 - dms.h, 431
- pack_dms_SetEventReport_t, 176
 - mode, 176
- pack_dms_SetFirmwarePreference
 - dms.h, 431
- pack_dms_SetPower
 - dms.h, 431
- pack_dms_SetPower_t, 176
 - mode, 176
 - Tlvresult, 177
- pack_dms_SetUSBComp
 - dms.h, 432
- pack_dms_SetUSBComp_t, 177
 - Tlvresult, 177
 - USBComp, 177
- pack_dms_UIMGetICCID
 - dms.h, 435
- pack_dms_UIMGetICCID_t, 178
 - Tlvresult, 178
- pack_fms_GetImagesPreference
 - fms.h, 453
- pack_fms_GetImagesPreference_t, 178
 - Tlvresult, 178
- pack_fms_GetStoredImages
 - fms.h, 453
- pack_fms_GetStoredImages_t, 179
 - Tlvresult, 179
- pack_fms_SetImagesPreference
 - fms.h, 453
- pack_fms_SetImagesPreference_t, 179
 - bForceDownload, 180
 - imageListSize, 180
 - modemindex, 180
 - pImageList, 180
 - Tlvresult, 180
- pack_loc_Delete_Assist_Data_t, 180
 - pBdsSVInfo, 180
 - pCellDb, 180
 - pClkInfo, 180
 - pGnssData, 181
 - pSVInfo, 181
 - Tlvresult, 181
- pack_loc_DeleteAssistData
 - loc.h, 465
- pack_loc_EventRegister
 - loc.h, 466
- pack_loc_EventRegister_t, 181
 - eventRegister, 183
 - Tlvresult, 183
- pack_loc_SLQSLOCGetBestAvailPos
 - loc.h, 467
- pack_loc_SLQSLOCGetBestAvailPos_t, 184
 - Tlvresult, 184
 - xid, 184
- pack_loc_SetExtPowerState
 - loc.h, 466
- pack_loc_SetExtPowerState_t, 183
 - extPowerState, 183
 - Tlvresult, 183
- pack_loc_SetOperationMode
 - loc.h, 466
- pack_loc_SetOperationMode_t, 183
 - mode, 184
 - Tlvresult, 184
- pack_loc_Start
 - loc.h, 467
- pack_loc_Start_t, 184
 - pApplicationInfo, 186
 - pConfigAltitudeAssumed, 186
 - pHorizontalAccuracyLvl, 186
 - pIntermediateReportState, 186
 - pMinIntervalTime, 186
 - pRecurrenceType, 186
 - SessionId, 186
 - Tlvresult, 186
- pack_loc_Stop
 - loc.h, 467
- pack_loc_Stop_t, 186
 - SessionId, 186
 - Tlvresult, 186
- pack_nas_GetACCOLC
 - nas.h, 478
- pack_nas_GetANAAAAAuthenticationStatus
 - nas.h, 479
- pack_nas_GetCDMANetworkParameters
 - nas.h, 479
- pack_nas_GetHomeNetwork
 - nas.h, 479
- pack_nas_GetNetworkPreference
 - nas.h, 480
- pack_nas_GetRFInfo
 - nas.h, 480
- pack_nas_GetServingNetwork
 - nas.h, 480
- pack_nas_GetServingNetworkCapabilities
 - nas.h, 480
- pack_nas_GetSignalStrengths
 - nas.h, 481
- pack_nas_PerformNetworkScan
 - nas.h, 481
- pack_nas_SLQSGetNetworkTime
 - nas.h, 483
- pack_nas_SLQSGetPLMNName
 - nas.h, 483
- pack_nas_SLQSGetPLMNName_t, 188
 - mcc, 188
 - mnc, 188
 - pMncPcsStatus, 188
- pack_nas_SLQSGetServingSystem
 - nas.h, 483
- pack_nas_SLQSGetSignalStrength
 - nas.h, 483

- pack_nas_SLQSGetSysInfo
 - nas.h, [484](#)
- pack_nas_SLQSGetSysSelectionPref
 - nas.h, [484](#)
- pack_nas_SLQSInitiateNetworkRegistration
 - nas.h, [484](#)
- pack_nas_SLQSInitiateNetworkRegistration_t, [188](#)
 - pChangeDuration, [189](#)
 - pMNRInfo, [189](#)
 - pMncPcsDigitStatus, [189](#)
 - regAction, [189](#)
- pack_nas_SLQSNasConfigSigInfo2
 - nas.h, [485](#)
- pack_nas_SLQSNasConfigSigInfo2_t, [189](#)
 - pHDRIODelta, [193](#)
 - pHDRIOTresh, [193](#)
 - pLTESigRptConfig, [193](#)
- pack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [485](#)
- pack_nas_SLQSNasGetSigInfo
 - nas.h, [485](#)
- pack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [486](#)
- pack_nas_SLQSNasIndicationRegisterExt_t, [194](#)
 - pDDTMInd, [196](#)
 - pDualStandByPrefInd, [196](#)
 - pErrorRateInd, [196](#)
 - pHDRSessionCloseInd, [196](#)
 - pLTECphyCa, [196](#)
 - pManagedRoamingInd, [196](#)
 - pNetworkTimeInd, [196](#)
 - pServingSystemInd, [196](#)
 - pSignalStrengthInd, [196](#)
 - pSubscriptionInfoInd, [196](#)
 - pSysInfoInd, [196](#)
 - pSystemSelectionInd, [197](#)
- pack_nas_SLQSNasSmiModemStatus
 - nas.h, [486](#)
- pack_nas_SLQSNasSmiOTAMessageCallback
 - nas.h, [486](#)
- pack_nas_SLQSNasSmiOTAMessageCallback_t, [197](#)
 - gsmUmtsDI, [197](#)
 - gsmUmtsUI, [198](#)
 - lteEmmDI, [198](#)
 - lteEmmUI, [198](#)
 - lteEsmDI, [198](#)
 - lteEsmUI, [198](#)
 - pRankIndicatorInd, [198](#)
- pack_nas_SLQSSetBandPreference
 - nas.h, [487](#)
- pack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [487](#)
- pack_nas_SLQSSetSignalStrengthsCallback_t, [198](#)
 - bEnable, [198](#)
 - pSigIndReq, [198](#)
- pack_nas_SLQSSetSysSelectionPref
 - nas.h, [487](#)
- pack_nas_SLQSSetSysSelectionPref_t, [198](#)
- pAcqOrderPref, [202](#)
- pBandPref, [202](#)
- pCSGID, [202](#)
- pChgDuration, [202](#)
- pEmerMode, [202](#)
- pGWAOrderPref, [202](#)
- pLTEBandPref, [202](#)
- pModePref, [203](#)
- pNetSelPref, [203](#)
- pPRLPref, [203](#)
- pRAT, [203](#)
- pRoamPref, [203](#)
- pSrvDomainPref, [203](#)
- pSrvRegRestriction, [203](#)
- pTdsdmaBandPref, [203](#)
- pack_nas_SLQSSwiGetLteCQI
 - nas.h, [488](#)
- pack_nas_SLQSSwiGetLteSccRxInfo
 - nas.h, [488](#)
- pack_nas_SetACCOLC
 - nas.h, [481](#)
- pack_nas_SetACCOLC_t, [186](#)
 - accolc, [187](#)
 - spc, [187](#)
- pack_nas_SetLURejectCallback
 - nas.h, [482](#)
- pack_nas_SetNetworkPreference
 - nas.h, [482](#)
- pack_nas_SetNetworkPreference_t, [187](#)
 - Duration, [187](#)
 - TechnologyPref, [188](#)
 - Tlvresult, [188](#)
- pack_nas_SetRFInfoCallback
 - nas.h, [482](#)
- pack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [482](#)
- pack_qmi_t, [203](#)
 - msgid, [203](#)
 - svc, [203](#)
 - timeout, [203](#)
 - xid, [203](#)
- pack_qos_SLQSQosGetNetworkStatus
 - qos.h, [547](#)
- pack_qos_SLQSQosSmiReadApnExtraParams
 - qos.h, [547](#)
- pack_qos_SLQSQosSmiReadApnExtraParams_t, [203](#)
 - apnId, [204](#)
- pack_qos_SLQSQosSmiReadDataStats
 - qos.h, [548](#)
- pack_qos_SLQSQosSmiReadDataStats_t, [204](#)
 - apnId, [204](#)
- pack_qos_SLQSSetQosEventCallback
 - qos.h, [549](#)
- pack_qos_SLQSSetQosEventCallback_t, [204](#)
 - enable, [205](#)
- pack_sms_SLQSDeleteSMS
 - sms.h, [558](#)
- pack_sms_SLQSDeleteSMS_t, [206](#)

- pMessageIndex, [206](#)
 - pMessageMode, [207](#)
 - pMessageTag, [207](#)
 - storageType, [207](#)
- pack_sms_SLQSGetSMS
 - sms.h, [558](#)
- pack_sms_SLQSGetSMS_t, [207](#)
 - messageIndex, [207](#)
 - pMessageMode, [207](#)
 - storageType, [207](#)
- pack_sms_SLQSGetSMSList
 - sms.h, [558](#)
- pack_sms_SLQSGetSMSList_t, [207](#)
 - pMessageMode, [208](#)
 - pRequestedTag, [208](#)
 - storageType, [208](#)
- pack_sms_SLQSModifySMSStatus
 - sms.h, [559](#)
- pack_sms_SLQSModifySMSStatus_t, [208](#)
 - messageIndex, [209](#)
 - messageTag, [209](#)
 - pMessageMode, [209](#)
 - storageType, [209](#)
- pack_sms_SendSMS
 - sms.h, [557](#)
- pack_sms_SendSMS_t, [205](#)
 - messageFormat, [205](#)
 - messageSize, [205](#)
 - pLinktimer, [205](#)
 - pMessage, [205](#)
- pack_sms_SetNewSMSCallback
 - sms.h, [557](#)
- pack_sms_SetNewSMSCallback_t, [205](#)
 - status, [206](#)
- pack_swiloc_SwiLocGetAutoStart
 - swiloc.h, [564](#)
- pack_swiloc_SwiLocSetAutoStart
 - swiloc.h, [564](#)
- pack_swiloc_SwiLocSetAutoStart_t, [209](#)
 - fix_rate, [210](#)
 - fix_type, [210](#)
 - function, [210](#)
 - max_dist, [210](#)
 - max_time, [210](#)
 - set_fix_rate, [210](#)
 - set_fix_type, [210](#)
 - set_function, [210](#)
 - set_max_dist, [210](#)
 - set_max_time, [210](#)
- pack_swioama_SLQSOMADMAAlertCallback
 - swioama.h, [566](#)
- pack_swioama_SLQSOMADMCancelSession
 - swioama.h, [567](#)
- pack_swioama_SLQSOMADMCancelSession_t, [210](#)
 - sessionType, [211](#)
- pack_swioama_SLQSOMADMGetSessionInfo
 - swioama.h, [567](#)
- pack_swioama_SLQSOMADMGetSessionInfo_t, [211](#)
 - SessionType, [211](#)
- pack_swioama_SLQSOMADMGetSettings
 - swioama.h, [568](#)
- pack_swioama_SLQSOMADMSelectSelection
 - swioama.h, [568](#)
- pack_swioama_SLQSOMADMSelectSelection_t, [211](#)
 - pDeferTime, [212](#)
 - pRejectReason, [212](#)
 - selection, [212](#)
- pack_swioama_SLQSOMADMSetSettings
 - swioama.h, [569](#)
- pack_swioama_SLQSOMADMSetSettings_t, [212](#)
 - FOTAdownload, [213](#)
 - pAutosdm, [213](#)
 - pFwAutoCheck, [213](#)
- pack_swioama_SLQSOMADMStartSession
 - swioama.h, [570](#)
- pack_swioama_SLQSOMADMStartSession_t, [213](#)
 - sessionType, [213](#)
- pack_uim_ChangePin
 - uim.h, [576](#)
- pack_uim_ChangePin_t, [214](#)
 - changePIN, [214](#)
 - EncryptedPIN1, [214](#)
 - pIndicationToken, [214](#)
 - pKeyReferenceID, [214](#)
 - sessionInfo, [214](#)
 - Tlvresult, [214](#)
- pack_uim_GetCardStatus
 - uim.h, [577](#)
- pack_uim_ReadTransparent
 - uim.h, [577](#)
- pack_uim_ReadTransparent_t, [215](#)
 - fileIndex, [215](#)
 - pEncryptData, [215](#)
 - pIndicationToken, [215](#)
 - readTransparent, [215](#)
 - sessionInfo, [216](#)
 - Tlvresult, [216](#)
- pack_uim_SLQSUIMEventRegister
 - uim.h, [578](#)
- pack_uim_SLQSUIMEventRegister_t, [217](#)
 - eventMask, [217](#)
- pack_uim_SLQSUIMGetSlotsStatus
 - uim.h, [578](#)
- pack_uim_SLQSUIMPowerDown
 - uim.h, [578](#)
- pack_uim_SLQSUIMPowerDown_t, [217](#)
 - slot, [217](#)
- pack_uim_SLQSUIMPowerUp
 - uim.h, [579](#)
- pack_uim_SLQSUIMPowerUp_t, [217](#)
 - plgnoreHotSwapSwitch, [218](#)
 - slot, [218](#)
- pack_uim_SLQSUIMSwitchSlot
 - uim.h, [579](#)
- pack_uim_SLQSUIMSwitchSlot_t, [218](#)
 - bLogicalSlot, [219](#)

- ulPhysicalSlot, [219](#)
- pack_uim_SetPinProtection
 - uim.h, [577](#)
- pack_uim_SetPinProtection_t, [216](#)
 - EncryptedPIN1, [216](#)
 - pIndicationToken, [216](#)
 - pKeyReferenceID, [217](#)
 - pinProtection, [216](#)
 - sessionInfo, [217](#)
 - Tlvresult, [217](#)
- pack_uim_UnblockPin
 - uim.h, [579](#)
- pack_uim_UnblockPin_t, [219](#)
 - EncryptedPIN1, [220](#)
 - pIndicationToken, [220](#)
 - pKeyReferenceID, [220](#)
 - pinProtection, [220](#)
 - sessionInfo, [220](#)
 - Tlvresult, [220](#)
- pack_uim_VerifyPin
 - uim.h, [580](#)
- pack_uim_VerifyPin_t, [220](#)
 - pEncryptedPIN1, [221](#)
 - pIndicationToken, [221](#)
 - pKeyReferenceID, [221](#)
 - sessionInfo, [221](#)
 - Tlvresult, [221](#)
 - verifyPIN, [221](#)
- pack_wds_GetByteTotals
 - wds.h, [590](#)
- pack_wds_GetConnectionRate
 - wds.h, [590](#)
- pack_wds_GetDefaultProfile
 - wds.h, [590](#)
- pack_wds_GetDefaultProfile_t, [221](#)
 - profiletype, [221](#)
- pack_wds_GetDefaultProfileNum
 - wds.h, [591](#)
- pack_wds_GetDefaultProfileNum_t, [221](#)
 - family, [222](#)
 - type, [222](#)
- pack_wds_GetDormancyState
 - wds.h, [591](#)
- pack_wds_GetDormancyState_t, [222](#)
- pack_wds_GetLastMobileIPError
 - wds.h, [592](#)
- pack_wds_GetLastMobileIPError_t, [222](#)
- pack_wds_GetMobileIP
 - wds.h, [592](#)
- pack_wds_GetMobileIP_t, [222](#)
- pack_wds_GetMobileIPProfile
 - wds.h, [592](#)
- pack_wds_GetMobileIPProfile_t, [222](#)
 - index, [222](#)
- pack_wds_GetPacketStatistics
 - wds.h, [593](#)
- pack_wds_GetPacketStatistics_t, [222](#)
 - pStatMask, [223](#)
- pack_wds_GetPacketStatus
 - wds.h, [593](#)
- pack_wds_GetPacketStatus_t, [223](#)
 - statmask, [223](#)
- pack_wds_GetSessionDuration
 - wds.h, [594](#)
- pack_wds_GetSessionDuration_t, [223](#)
- pack_wds_GetSessionState
 - wds.h, [594](#)
- pack_wds_RMSetTransferStatistics
 - wds.h, [595](#)
- pack_wds_RMSetTransferStatistics_t, [223](#)
 - RmTrasnferStaticsReq, [223](#)
- pack_wds_SLQSCreateProfile
 - wds.h, [596](#)
- pack_wds_SLQSCreateProfile_t, [226](#)
 - pCurProfile, [227](#)
 - pProfileId, [227](#)
 - pProfileType, [227](#)
- pack_wds_SLQSDeleteProfile
 - wds.h, [596](#)
- pack_wds_SLQSDeleteProfile_t, [227](#)
 - profileIndex, [227](#)
 - profileType, [227](#)
- pack_wds_SLQSGet3GPPConfigItem
 - wds.h, [597](#)
- pack_wds_SLQSGetCurrDataSystemStat
 - wds.h, [597](#)
- pack_wds_SLQSGetCurrDataSystemStat_t, [227](#)
- pack_wds_SLQSGetCurrentChannelRate
 - wds.h, [598](#)
- pack_wds_SLQSGetDUNCallInfo
 - wds.h, [599](#)
- pack_wds_SLQSGetDUNCallInfo_t, [227](#)
 - Mask, [228](#)
 - pReportChannelRate, [228](#)
 - pReportConnStatus, [228](#)
 - pReportDataBearerTech, [228](#)
 - pReportDormStatus, [228](#)
 - pTransferStatInd, [228](#)
- pack_wds_SLQSGetDataBearerTechnology
 - wds.h, [598](#)
- pack_wds_SLQSGetDataBearerTechnology_t, [227](#)
- pack_wds_SLQSGetProfileSettings
 - wds.h, [599](#)
- pack_wds_SLQSGetProfileSettings_t, [228](#)
 - ProfileId, [229](#)
 - ProfileType, [229](#)
- pack_wds_SLQSGetRuntimeSettings
 - wds.h, [599](#)
- pack_wds_SLQSGetRuntimeSettings_t, [229](#)
 - pReqSettings, [229](#)
- pack_wds_SLQSModifyProfile
 - wds.h, [600](#)
- pack_wds_SLQSModifyProfile_t, [229](#)
 - curProfile, [230](#)
 - pProfileId, [230](#)
 - pProfileType, [230](#)

- pack_wds_SLQSSGetDHCPv4ClientConfig
 - wds.h, 601
- pack_wds_SLQSSGetDHCPv4ClientConfig_t, 233
 - pProfileId, 233
- pack_wds_SLQSSGetLoopback
 - wds.h, 602
- pack_wds_SLQSSSetLoopback
 - wds.h, 602
- pack_wds_SLQSSSetLoopback_t, 233
 - loopbackMode, 234
 - loopbackMultiplier, 234
- pack_wds_SLQSSSet3GPPConfigItem
 - wds.h, 600
- pack_wds_SLQSSSet3GPPConfigItem_t, 230
 - p3gppRelease, 231
 - pProfileList, 232
- pack_wds_SLQSSetIPFamilyPreference
 - wds.h, 601
- pack_wds_SLQSSetIPFamilyPreference_t, 232
 - IPFamilyPreference, 232
- pack_wds_SLQSSetWdsEventCallback
 - wds.h, 601
- pack_wds_SLQSSetWdsEventCallback_t, 232
 - currentDataBearer, 233
 - dataBearer, 233
 - dataSystemStatus, 233
 - dormancyStatus, 233
 - interval, 233
 - mobileIP, 233
 - transferStats, 233
- pack_wds_SLQSSStartDataSession
 - wds.h, 603
- pack_wds_SLQSSStartDataSession_t, 234
 - pAuth, 234
 - pPass, 235
 - pTech, 235
 - pUser, 235
 - pprofileid3gpp, 235
 - pprofileid3gpp2, 235
- pack_wds_SLQSSStopDataSession
 - wds.h, 603
- pack_wds_SLQSSStopDataSession_t, 235
 - psid, 235
- pack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, 603
- pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 235
 - contextId, 235
 - contextType, 235
- pack_wds_SetDefaultProfile
 - wds.h, 595
- pack_wds_SetDefaultProfile_t, 224
 - authentication, 224
 - ipAddress, 224
 - pApnname, 224
 - pName, 224
 - pPassword, 224
 - pUsername, 224
 - pdpType, 224
 - primaryDNS, 224
 - profileType, 224
 - secondaryDNS, 224
- pack_wds_SetDefaultProfileNum
 - wds.h, 595
- pack_wds_SetDefaultProfileNum_t, 225
 - family, 225
 - index, 225
 - type, 225
- pack_wds_SetMobileIPProfile
 - wds.h, 596
- pack_wds_SetMobileIPProfile_t, 225
 - index, 225
 - pAAASPI, 226
 - pAddress, 226
 - pEnabled, 226
 - pHASPI, 226
 - pMNAHA, 226
 - pMNHA, 226
 - pNAI, 226
 - pPrimaryHA, 226
 - pRevTunneling, 226
 - pSecondaryHA, 226
 - spc, 226
- PackCreateProfileOut, 235
 - ExtErrorCode, 236
 - ProfileIndex, 236
 - ProfileType, 236
- package_name
 - unpack_omaDmFotaTlv_t, 338
- packageid_str
 - unpack_dms_GetFirmwareInfo_t, 268
- packetZone
 - nas_CDMASysInfo, 88
- packetZoneValid
 - nas_CDMASysInfo, 88
- packgetDyingGaspCfg, 236
 - pDestSMSContent, 236
 - pDestSMSNum, 236
- packgetDyingGaspStatistics, 236
 - pSMSAttemptedFlag, 237
 - pTimeStamp, 237
- path
 - uim_fileInfo, 255
- pathLen
 - uim_fileInfo, 255
- pbPlatform
 - unpack_dms_GetOfflineReason_t, 273
- pci
 - nas_cellParams, 89
 - nas_PhyCaAggPcellInfo, 128
 - nas_PhyCaAggScellIndType, 130
 - nas_PhyCaAggScellInfo, 133
 - nas_umtsLTENbrCell, 156
 - NASPhyCaAggPcellInfo, 167
 - NASPhyCaAggScellIndType, 169
 - NASPhyCaAggScellInfo, 170
- pcsfQDNAddress

- wds_PCSCFFQDNAddressList, [404](#)
- pdpType
 - pack_wds_SetDefaultProfile_t, [224](#)
- pdptype
 - unpack_wds_GetDefaultProfile_t, [378](#)
- peakRate
 - unpack_qos_tokenBucket_t, [358](#)
- peakThroughputClass
 - LibPackGPRSRequestedQoS, [40](#)
 - wds_GPRSQoS, [402](#)
- PersistentTechPref
 - unpack_nas_GetNetworkPreference_t, [308](#)
- persoFeature
 - appStats, [26](#)
 - uim_appStatus, [250](#)
- persoRetries
 - appStats, [26](#)
 - uim_appStatus, [250](#)
- persoState
 - appStats, [26](#)
 - uim_appStatus, [250](#)
- persoUnblockRetries
 - appStats, [26](#)
 - uim_appStatus, [250](#)
- PhyCaAggPcellInfo
 - NasGetLTECphyCaInfo, [164](#)
- PhyCaAggScellIDBw
 - NasGetLTECphyCaInfo, [164](#)
- PhyCaAggScellIndType
 - NasGetLTECphyCaInfo, [164](#)
- PhyCaAggScellIndex
 - NasGetLTECphyCaInfo, [164](#)
- PhyCaAggScellInfo
 - NasGetLTECphyCaInfo, [164](#)
- pin1Len
 - uim_encryptedPIN1, [254](#)
- pin1Retries
 - appStats, [27](#)
 - uim_appStatus, [251](#)
- pin1State
 - appStats, [27](#)
 - uim_appStatus, [251](#)
- pin1Val
 - uim_encryptedPIN1, [254](#)
- pin2Retries
 - appStats, [27](#)
 - uim_appStatus, [251](#)
- pin2State
 - appStats, [27](#)
 - uim_appStatus, [251](#)
- pinID
 - uim_changeUIMPIN, [253](#)
 - uim_setPINProtection, [259](#)
 - uim_unblockUIMPIN, [262](#)
 - uim_verifyUIMPIN, [263](#)
- pinLen
 - uim_changeUIMPIN, [253](#)
 - uim_verifyUIMPIN, [263](#)
- pinLength
 - uim_setPINProtection, [259](#)
- pinOperation
 - uim_setPINProtection, [259](#)
- pinProtection
 - pack_uim_SetPinProtection_t, [216](#)
 - pack_uim_UnblockPin_t, [220](#)
- pinVal
 - uim_changeUIMPIN, [253](#)
 - uim_verifyUIMPIN, [263](#)
- pinValue
 - uim_setPINProtection, [259](#)
- PkgDescLength
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, [367](#)
- PkgDescription
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, [367](#)
- PkgName
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, [367](#)
- PkgNameLength
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, [367](#)
- pkgver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, [285](#)
- PktErrRate
 - unpack_qos_swiQosFlow_t, [357](#)
- plmn
 - nas_GERANInfo, [98](#)
 - nas_LTEInfoIntrafreq, [116](#)
 - nas_UMTSInfo, [154](#)
- port
 - unpack_qos_Port_t, [342](#)
- pprofileid3gpp
 - pack_wds_SLQSStartDataSession_t, [235](#)
- pprofileid3gpp2
 - pack_wds_SLQSStartDataSession_t, [235](#)
- prDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- prDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- prPCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- prPCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- Precedence
 - unpack_qos_swiQosFilter_t, [353](#)
- precedenceClass
 - LibPackGPRSRequestedQoS, [40](#)
 - wds_GPRSQoS, [402](#)
- prefNetwork
 - unpack_wds_SLQSGetCurrDataSystemStat_t, [387](#)

- unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- Preferred
 - nas_QmiNas3GppNetworkInfo, 134
- prefixLen
 - unpack_qos_IPv6Addr_t, 340
- priSize
 - unpack_dms_GetFirmwareRevisions_t, 269
- pridns
 - unpack_wds_GetDefaultProfile_t, 378
- pridnsv6
 - unpack_wds_GetDefaultProfile_t, 378
- primaryDNS
 - pack_wds_SetDefaultProfile_t, 224
- PrimaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- PrimaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- primaryHA
 - unpack_wds_GetMobileIPProfile_t, 380
- priver
 - unpack_dms_SLQSSwiGetFirmwareCurr_t, 285
- priversion_str
 - unpack_dms_GetFirmwareInfo_t, 268
- ProfileID
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- ProfileId
 - pack_wds_SLQSGetProfileSettings_t, 229
- profileId
 - wdsDhcpv4ProfileId, 410
- ProfileId3GPP2
 - unpack_qos_swiQosFlow_t, 357
- ProfileIndex
 - PackCreateProfileOut, 236
- profileIndex
 - pack_wds_SLQSDeleteProfile_t, 227
 - wds_ProfileIdentifier, 405
- profileList
 - unpack_wds_SLQSGet3GPPConfigItem_t, 386
- ProfileName
 - unpack_wds_SLQSGetRuntimeSettings_t, 391
- ProfileType
 - pack_wds_SLQSGetProfileSettings_t, 229
 - PackCreateProfileOut, 236
 - unpack_wds_SLQSGetProfileSettings_t, 390
- profileType
 - pack_wds_SetDefaultProfile_t, 224
 - pack_wds_SLQSDeleteProfile_t, 227
 - wds_ProfileIdentifier, 405
 - wdsDhcpv4ProfileId, 410
- profiletype
 - pack_wds_GetDefaultProfile_t, 221
- Protocol
 - unpack_nas_GetCDMANetworkParameters_t, 306
- psAttachState
 - nas_servSystem, 142
 - NASServingSystemInfo, 173
- psBarStatus
 - nas_CallBarringSysInfo, 81
 - nas_callBarStatus, 82
- psState
 - nas_CommInfo, 91
- psc
 - nas_UMTSInfo, 154
 - nas_wcdmaCellInfo, 158
 - nas_WCDMASysInfo, 163
- pscValid
 - nas_WCDMASysInfo, 163
- pscsfIPv4Addr
 - wds_PCSCFIPv4ServerAddressList, 404
- psid
 - pack_wds_SLQSStopDataSession_t, 235
 - unpack_wds_SLQSStartDataSession_t, 396
- puk1Retries
 - appStats, 27
 - uim_appStatus, 251
- puk2Retries
 - appStats, 27
 - uim_appStatus, 251
- pukLen
 - uim_unblockUIMPIN, 262
- pukVal
 - uim_unblockUIMPIN, 262
- QCI
 - LibPackQosClassID, 63
- QFlowState
 - unpack_qos_QosFlowInfo_t, 343
- QMI pack/unpack (pack), 19
- qaGobiApiTableBandClasses.h, 500
- qaGobiApiTableCallControlReturnReasons.h, 503
- qaGobiApiTableCallEndReasons.h, 504
- qaGobiApiTableCarrierCodes.h, 520
- qaGobiApiTableCodingScheme.h, 522
- qaGobiApiTableGpsCapabilityCodes.h, 524
- qaGobiApiTablePowerModes.h, 525
- qaGobiApiTableRadioInterfaces.h, 526
- qaGobiApiTableRegionCodes.h, 526
- qaGobiApiTableServiceOptions.h, 527
- qaGobiApiTableSupServiceInfoClasses.h, 529
- qaGobiApiTableSwtAudio.h, 529
- qaGobiApiTableSwtOMADMUpdateCompleteStatus.h, 530
- qaGobiApiTableVoiceCallEndReasons.h, 531
- qm_wds_ds_profile_extended_err_codes
 - qmerrno.h, 544
- qmerrno.h
 - eQCWWAN_ERR_API_MUTEX_TIMEOUT, 541
 - eQCWWAN_ERR_BUFFER_SZ, 540
 - eQCWWAN_ERR_CANCEL_OP, 541
 - eQCWWAN_ERR_DRIVER, 541
 - eQCWWAN_ERR_ENUM_BEGIN, 540
 - eQCWWAN_ERR_ENUM_END, 541
 - eQCWWAN_ERR_FILE_COPY, 540
 - eQCWWAN_ERR_FILE_OPEN, 540
 - eQCWWAN_ERR_GENERAL, 540
 - eQCWWAN_ERR_INTERNAL, 540

- eQCWWAN_ERR_INVALID_ARG, 540
- eQCWWAN_ERR_INVALID_DEVID, 540
- eQCWWAN_ERR_INVALID_FILE, 540
- eQCWWAN_ERR_INVALID_QMI_RSP, 540
- eQCWWAN_ERR_INVALID_XID, 541
- eQCWWAN_ERR_MALFORMED_QMI_RSP, 540
- eQCWWAN_ERR_MEMORY, 540
- eQCWWAN_ERR_MULTIPLE_DEVICES, 541
- eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED, 541
- eQCWWAN_ERR_NO_CANCELABLE_OP, 541
- eQCWWAN_ERR_NO_CONNECTION, 540
- eQCWWAN_ERR_NO_DEVICE, 540
- eQCWWAN_ERR_NO_SIGNAL, 541
- eQCWWAN_ERR_NONE, 540
- eQCWWAN_ERR_NULL_TLV, 544
- eQCWWAN_ERR_OFFLINE, 541
- eQCWWAN_ERR_PDU_GENERATION, 541
- eQCWWAN_ERR_QMI_ABORTED, 541
- eQCWWAN_ERR_QMI_ACCESS_DENIED, 543
- eQCWWAN_ERR_QMI_ACK_NOT_SENT, 543
- eQCWWAN_ERR_QMI_ARG_TOO_LONG, 541
- eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, 542
- eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, 542
- eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, 543
- eQCWWAN_ERR_QMI_CALL_FAILED, 541
- eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 544
- eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, 543
- eQCWWAN_ERR_QMI_CAT_END, 544
- eQCWWAN_ERR_QMI_CAT_START, 544
- eQCWWAN_ERR_QMI_CAUSE_CODE, 542
- eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, 541
- eQCWWAN_ERR_QMI_CONNECT, 540
- eQCWWAN_ERR_QMI_DEVICE_IN_USE, 541
- eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 542
- eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, 542
- eQCWWAN_ERR_QMI_DISABLED, 542
- eQCWWAN_ERR_QMI_ENCODING, 542
- eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, 544
- eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 544
- eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 543
- eQCWWAN_ERR_QMI_FDN_RESTRICT, 543
- eQCWWAN_ERR_QMI_FLOW_SUSPENDED, 542
- eQCWWAN_ERR_QMI_GENERAL, 542
- eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, 543
- eQCWWAN_ERR_QMI_IFACE, 540
- eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, 543
- eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, 542
- eQCWWAN_ERR_QMI_INCORRECT_PIN, 541
- eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, 543
- eQCWWAN_ERR_QMI_INJECT_TIMEOUT, 543
- eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, 542
- eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, 542
- eQCWWAN_ERR_QMI_INTERNAL, 541
- eQCWWAN_ERR_QMI_INVALID_ARG, 542
- eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, 541
- eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, 542
- eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, 544
- eQCWWAN_ERR_QMI_INVALID_HANDLE, 541
- eQCWWAN_ERR_QMI_INVALID_ID, 542
- eQCWWAN_ERR_QMI_INVALID_INDEX, 542
- eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, 542
- eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, 542
- eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, 542
- eQCWWAN_ERR_QMI_INVALID_OPERATION, 543
- eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, 541
- eQCWWAN_ERR_QMI_INVALID_PINID, 541
- eQCWWAN_ERR_QMI_INVALID_PROFILE, 541
- eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, 541
- eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, 542
- eQCWWAN_ERR_QMI_INVALID_QMI_CMD, 543
- eQCWWAN_ERR_QMI_INVALID_QOS_ID, 542
- eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, 542
- eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, 541
- eQCWWAN_ERR_QMI_INVALID_TECH_PREF, 541
- eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, 544
- eQCWWAN_ERR_QMI_INVALID_TRANSITION, 542
- eQCWWAN_ERR_QMI_INVALID_TX_ID, 541
- eQCWWAN_ERR_QMI_MALFORMED_MSG, 541
- eQCWWAN_ERR_QMI_MAX, 543
- eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, 542
- eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, 542

- eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, [542](#)
- eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, [542](#)
- eQCWWAN_ERR_QMI_MISSING_ARG, [541](#)
- eQCWWAN_ERR_QMI_MSG_BLOCKED, [543](#)
- eQCWWAN_ERR_QMI_NETWORK_ABORTED, [543](#)
- eQCWWAN_ERR_QMI_NETWORK_NOT_READ_Y, [542](#)
- eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, [542](#)
- eQCWWAN_ERR_QMI_NO_EFFECT, [541](#)
- eQCWWAN_ERR_QMI_NO_ENTRY, [542](#)
- eQCWWAN_ERR_QMI_NO_FREE_PROFILE, [541](#)
- eQCWWAN_ERR_QMI_NO_MEMORY, [541](#)
- eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, [541](#)
- eQCWWAN_ERR_QMI_NO_RADIO, [543](#)
- eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, [543](#)
- eQCWWAN_ERR_QMI_NO_THRESHOLDS, [541](#)
- eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, [542](#)
- eQCWWAN_ERR_QMI_NOT_PROVISIONED, [541](#)
- eQCWWAN_ERR_QMI_NOT_SUPPORTED, [543](#)
- eQCWWAN_ERR_QMI_OFFSET, [541](#)
- eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, [541](#)
- eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, [541](#)
- eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, [543](#)
- eQCWWAN_ERR_QMI_OUT_OF_CALL, [541](#)
- eQCWWAN_ERR_QMI_PIN_BLOCKED, [542](#)
- eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, [542](#)
- eQCWWAN_ERR_QMI_POLICY_MISMATCH, [543](#)
- eQCWWAN_ERR_QMI_REQ, [540](#)
- eQCWWAN_ERR_QMI_REQ_SCH, [540](#)
- eQCWWAN_ERR_QMI_REQ_TO, [540](#)
- eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, [542](#)
- eQCWWAN_ERR_QMI_RSP, [540](#)
- eQCWWAN_ERR_QMI_RSP_TO, [540](#)
- eQCWWAN_ERR_QMI_SEGMENT_ORDER, [543](#)
- eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, [543](#)
- eQCWWAN_ERR_QMI_SESSION_INACTIVE, [542](#)
- eQCWWAN_ERR_QMI_SESSION_INVALID, [542](#)
- eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, [542](#)
- eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, [543](#)
- eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, [542](#)
- eQCWWAN_ERR_QMI_SMSC_ADDR, [543](#)
- eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, [543](#)
- eQCWWAN_ERR_QMI_TPDU_TYPE, [543](#)
- eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, [541](#)
- eQCWWAN_ERR_QMI_UNKNOWN, [542](#)
- eQCWWAN_ERR_QMI_WIDTH, [544](#)
- eQCWWAN_ERR_RESET, [541](#)
- eQCWWAN_ERR_SWICM_AM_VERS_ERROR, [543](#)
- eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, [543](#)
- eQCWWAN_ERR_SWICM_END, [544](#)
- eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, [543](#)
- eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, [543](#)
- eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, [543](#)
- eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, [543](#)
- eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, [543](#)
- eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, [543](#)
- eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, [543](#)
- eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, [543](#)
- eQCWWAN_ERR_SWICM_SOCKET_IN_USE, [543](#)
- eQCWWAN_ERR_SWICM_START, [543](#)
- eQCWWAN_ERR_SWICM_TIMEOUT, [543](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6DWN, [543](#)
- eQCWWAN_ERR_SWICM_V4DWN_V6UP, [543](#)
- eQCWWAN_ERR_SWICM_V4UP_V6DWN, [543](#)
- eQCWWAN_ERR_SWICM_V4UP_V6UP, [543](#)
- eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, [544](#)
- eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, [544](#)
- eQCWWAN_ERR_SWIDCS_END, [544](#)
- eQCWWAN_ERR_SWIDCS_FILEIO_ERR, [544](#)
- eQCWWAN_ERR_SWIDCS_IOCTL_ERR, [544](#)
- eQCWWAN_ERR_SWIDCS_START, [544](#)
- eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, [544](#)
- eQCWWAN_ERR_SWIIM_END, [544](#)
- eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, [544](#)
- eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, [544](#)
- eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, [544](#)

- eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, [544](#)
- eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, [544](#)
- eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, [544](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, [544](#)
- eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, [544](#)
- eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, [544](#)
- eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, [544](#)
- eQCWWAN_ERR_SWIIM_INVALID_PATH, [544](#)
- eQCWWAN_ERR_SWIIM_OPENING_DIR, [544](#)
- eQCWWAN_ERR_SWIIM_OPENING_FILE, [544](#)
- eQCWWAN_ERR_SWIIM_START, [544](#)
- eQCWWAN_ERR_SWISM_END, [544](#)
- eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, [544](#)
- eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, [544](#)
- eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, [544](#)
- eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, [544](#)
- eQCWWAN_ERR_SWISMS_START, [544](#)
- eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, [545](#)
- eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, [545](#)
- eWDS_ERR_PROFILE_REG_END, [545](#)
- eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_FAIL, [545](#)
- eWDS_ERR_PROFILE_REG_RESULT_LIST_END, [545](#)
- qmerrno.h, [538](#)
- eQCWWANError, [540](#)
- qm_wds_ds_profile_extended_err_codes, [544](#)
- qmiSmsMessageList, [237](#)
- messageIndex, [237](#)
- messageTag, [237](#)
- qmiWDSDataBearerTechnology, [237](#)
- currentNetwork, [238](#)
- ratMask, [238](#)
- soMask, [238](#)
- qos.h, [545](#)
- pack_qos_SLQSQosGetNetworkStatus, [547](#)
- pack_qos_SLQSQosSmiReadApnExtraParams, [547](#)
- pack_qos_SLQSQosSmiReadDataStats, [548](#)
- pack_qos_SLQSSetQosEventCallback, [549](#)
- unpack_qos_SLQSQosGetNetworkStatus, [549](#)
- unpack_qos_SLQSQosSmiReadApnExtraParams, [550](#)
- unpack_qos_SLQSQosSmiReadDataStats, [550](#)
- unpack_qos_SLQSSetQosEventCallback, [551](#)
- unpack_qos_SLQSSetQosEventCallback_ind, [551](#)
- unpack_qos_SLQSSetQosNWStatusCallback_ind, [551](#)
- unpack_qos_SLQSSetQosPriEventCallback_ind, [552](#)
- unpack_qos_SLQSSetQosStatusCallback_ind, [553](#)
- qosDeliveryOrder
- LibPackUMTSQoS, [67](#)
- wds_UMTSMInQoS, [407](#)
- qosFlow
- unpack_qos_SLQSQosSmiReadDataStats_t, [347](#)
- QosFlowInfo
- unpack_qos_SLQSSetQosEventCallback_ind_t, [348](#)
- RAN
- unpack_nas_GetServingNetwork_t, [309](#)
- RAT
- nas_QmiNas3GppNetworkRAT, [135](#)
- RATMask
- currNetworkInfo, [29](#)
- wds_currNetworkInfo, [400](#)
- RFBandInfoElements, [238](#)
- activeBandClass, [238](#)
- activeChannel, [238](#)
- radioInterface, [238](#)
- unpack_nas_GetRFInfo_t, [308](#)
- RFTiv

- unpack_nas_SetEventReportInd_t, 312
- RMAutoConnect
 - pack_dms_SetCustFeature_t, 175
 - unpack_dms_GetCustFeature_t, 264
- RPTlv
 - NASQmiCbkNasSystemSelPrefInd, 171
- RRTlv
 - unpack_nas_SetEventReportInd_t, 312
- RXChan
 - nas_LTEInfo, 113
- rXDroppedCount
 - unpack_wds_GetPacketStatus_t, 383
- rXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, 383
- rXOkBytesCount
 - unpack_wds_GetPacketStatus_t, 383
- rXPacketErrors
 - unpack_wds_GetPacketStatus_t, 383
- rXPacketOverflows
 - unpack_wds_GetPacketStatus_t, 383
- rXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, 383
- radio
 - unpack_nas_GetSignalStrengths_t, 310
- radiolf
 - nas_ecioListElement, 95
 - nas_errorRateListElement, 96
 - nas_rsrqInformation, 138
 - nas_rxSignalStrengthListElement, 140
- Radiolfaces
 - unpack_dms_GetDeviceCap_t, 265
 - unpack_dms_GetDeviceCapabilities_t, 266
 - unpack_nas_GetServingNetwork_t, 309
- RadiolfacesSize
 - unpack_dms_GetDeviceCap_t, 265
 - unpack_nas_GetServingNetwork_t, 309
- radiolfacesSize
 - unpack_dms_GetDeviceCapabilities_t, 266
- radiolInterface
 - nas_RFInfoTlv, 137
 - nas_roamIndList, 137
 - nas_servSystem, 142
 - nas_SignalStrengthTlv, 143
 - nas_timeInfo, 152
 - RFBandInfoElements, 238
- radiolInterfaceList
 - NASServingSystemInfo, 173
- radiolInterfaceNo
 - NASServingSystemInfo, 173
- radiolInterfaceSize
 - nas_RFInfoTlv, 137
- range
 - unpack_qos_Port_t, 342
- rat
 - nas_CSgid, 92
 - nas_MNRInfo, 126
- ratMask
 - qmiWDSDataBearerTechnology, 238
- unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- readTransparent
 - pack_uim_ReadTransparent_t, 215
- reason
 - unpack_qos_SLQSSetQosStatusCallback_ind_t, 350
- reconfigReqd
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 393
- refData
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 286
- refString
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 286
- refpn
 - nas_CDMAInfo, 84
- regAction
 - pack_nas_SLQSSInitiateNetworkRegistration_t, 189
- RegForeignNID
 - unpack_nas_GetCDMANetworkParameters_t, 306
- RegForeignSID
 - unpack_nas_GetCDMANetworkParameters_t, 306
- RegHomeSID
 - unpack_nas_GetCDMANetworkParameters_t, 306
- regPrd
 - nas_AddCDMASysInfo, 80
- regRejectInfoValid
 - nas_GSMsSysInfo, 104
 - nas_LTESysInfo, 124
 - nas_WCDMASysInfo, 163
- regState
 - nas_servSystem, 142
- RegistrationState
 - unpack_nas_GetServingNetwork_t, 309
- registrationState
 - NASServingSystemInfo, 173
- rejCause
 - nas_GSMsSysInfo, 104
 - nas_LTESysInfo, 124
 - nas_WCDMASysInfo, 163
- rejectCause
 - nas_RejectReasonTlv, 136
- rejectSrvDomain
 - nas_GSMsSysInfo, 105
 - nas_LTESysInfo, 124
 - nas_WCDMASysInfo, 163
- reliabilityClass
 - LibPackGPRSRequestedQoS, 40
 - wds_GPRSQoS, 402
- resBerRatio
 - LibPackUMTSQoS, 67
 - wds_UMTSMInQoS, 407
- ResCode
 - unpack_dms_SLQSSwiGetFwUpdateStatus_t, 286
- resetInfoInd
 - pack_dms_SLQSDmsSwiIndicationRegister_t, 177
- RetryCount

- unpack_swioma_SLQSOMADMGetSessionInfo_t, 367
- revTunneling
 - unpack_wds_GetMobileIPProfile_t, 381
- RmTrasnferStaticsReq
 - pack_wds_RMSetTransferStatistics_t, 223
- rmTrasnferStaticsReq, 238
 - bResetStatistics, 238
 - ulMask, 238
- roamIndicator
 - nas_roamIndList, 137
- RoamIndicatorVal
 - unpack_nas_SLQSGetServingSystem_t, 318
- RoamPref
 - NASRoamPreferenceTlv, 171
- roamStatus
 - nas_sysInfoCommon, 148
- roamStatusValid
 - nas_sysInfoCommon, 148
- Roaming
 - nas_QmiNas3GppNetworkInfo, 134
 - unpack_nas_GetCDMANetworkParameters_t, 306
 - unpack_nas_GetServingNetwork_t, 309
- roaming
 - unpack_nas_SetRoamingIndicatorCallback_ind_t, 314
- RoamingIndicatorList
 - unpack_nas_SLQSGetServingSystem_t, 318
- rptRate
 - nas_LTESigRptConfig, 120
- rscp
 - nas_UMTSInfo, 154
 - tdscdmaSigInfoExt, 247
- rsrp
 - lteSSInfo, 78
 - nas_cellParams, 89
 - nas_RxSigInfo, 139
 - nas_umtsLTENbrCell, 156
- rsrplevel
 - nas_lteRsrpinformation, 118
- rsrq
 - lteSSInfo, 78
 - nas_cellParams, 89
 - nas_rsrqInformation, 138
 - nas_SccRxInfo, 141
 - nas_umtsLTENbrCell, 156
- rsrqDelta
 - nas_SLQSSignalStrengthsIndReq, 144
- rsrqInfo
 - nas_SLQSSignalStrengthsInformation, 145
 - unpack_nas_SLQSGetSignalStrength_t, 319
- rsi
 - cdmaSSInfo, 28
 - hdrSSInfo, 38
 - lteSSInfo, 78
 - nas_cellParams, 89
 - nas_gsmCellInfo, 100
 - tdscdmaSigInfoExt, 247
 - unpack_nas_GetSignalStrengths_t, 310
- rx_bytes
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- rx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- rxChainIndex
 - nas_RxSigInfo, 139
- rxLev
 - nas_GERANInfo, 98
- rxOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- rxPower
 - nas_RxSigInfo, 139
- RxQFilter
 - unpack_qos_QosFlowInfo_t, 343
- RxQFlowGranted
 - unpack_qos_QosFlowInfo_t, 343
- rxSignalStrength
 - nas_rxSignalStrengthListElement, 140
- rxSignalStrengthDelta
 - nas_SLQSSignalStrengthsIndReq, 144
- rxSignalStrengthInfo
 - nas_SLQSSignalStrengthsInformation, 145
- rxSignalStrengthList
 - unpack_nas_SLQSGetSignalStrength_t, 319
- rxSignalStrengthListLen
 - unpack_nas_SLQSGetSignalStrength_t, 319
- SCI
 - unpack_nas_GetCDMANetworkParameters_t, 306
- SCM
 - unpack_nas_GetCDMANetworkParameters_t, 306
- SDPTlv
 - NASQmiCbkNasSystemSelPrefInd, 171
- SDU_HDR_LEN
 - common.h, 413
- SHORT
 - SwiDataTypes.h, 563
- sIntraSearch
 - nas_LTEInfoIntrafreq, 116
- SLQSSSTlv
 - unpack_nas_SetEventReportInd_t, 312
- sMSCAddress, 241
 - data, 242
 - length, 242
- sMSCAddressInfo
 - sms.h, 555
- sMSCAddressTlv, 242
 - SMSCInfo, 242
 - TlvPresent, 242
- SMSCInfo
 - sMSCAddressTlv, 242
- SMSCtlv
 - unpack_sms_SetNewSMSCallback_ind_t, 361
- sMSEtwMessage, 242
 - data, 243
 - length, 243

- notificationType, [243](#)
- sMSEtwsMessageInfo
 - sms.h, [555](#)
- sMSEtwsMessageTlv, [243](#)
 - EtwsMessageInfo, [243](#)
 - TlvPresent, [243](#)
- sMSEtwsPlmn, [243](#)
 - mobileCountryCode, [244](#)
 - mobileNetworkCode, [244](#)
- sMSEtwsPlmnInfo
 - sms.h, [556](#)
- sMSMTMessage, [244](#)
 - messageIndex, [245](#)
 - storageType, [245](#)
- sMSMTMessageInfo
 - sms.h, [556](#)
- sMSMessageMode, [244](#)
 - messageMode, [244](#)
- sMSMessageModelInfo
 - sms.h, [556](#)
- sMSOnIMS, [245](#)
 - smsOnIMS, [245](#)
- sMSOnIMSInfo
 - sms.h, [556](#)
- sMSOnIMSTlv, [245](#)
 - IMSInfo, [245](#)
 - TlvPresent, [245](#)
- SMSSupport
 - pack_dms_SetCustFeature_t, [175](#)
 - unpack_dms_GetCustFeature_t, [264](#)
- sMSTransferRouteMTMessage, [245](#)
 - ackIndicator, [246](#)
 - data, [246](#)
 - format, [246](#)
 - length, [246](#)
 - transactionID, [246](#)
- sMSTransferRouteMTMessageInfo
 - sms.h, [556](#)
- sNonIntraSearch
 - nas_LTEInfoIntrafreq, [116](#)
- SOMask
 - currNetworkInfo, [29](#)
 - wds_currNetworkInfo, [400](#)
- sPhyCaAggPcellInfo
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [312](#)
- sPhyCaAggScellIDIBw
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [312](#)
- sPhyCaAggScellIndType
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [313](#)
- sPhyCaAggScellIndex
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [313](#)
- sPhyCaAggScellInfo
 - unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [313](#)
- SSInfo
 - unpack_nas_SetServingSystemCallback_ind_t, [314](#)
- sSLQSSignalStrengthsInfo
 - nas_SLQSSignalStrengthsTlv, [145](#)
- SSTlv
 - unpack_nas_SetEventReportInd_t, [312](#)
- SWI_API
 - SwiDataTypes.h, [563](#)
- SWIWWANCMAPI.h, [574](#)
- scell_idx
 - nas_PhyCaAggScellIndex, [129](#)
 - NASPhyCaAggScellIndex, [168](#)
- scell_state
 - nas_PhyCaAggScellIndType, [130](#)
 - nas_PhyCaAggScellInfo, [133](#)
 - NASPhyCaAggScellIndType, [169](#)
 - NASPhyCaAggScellInfo, [170](#)
- sduErrorRatio
 - LibPackUMTSQoS, [67](#)
 - wds_UMTSMInQoS, [408](#)
- seDNSIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- seDNSIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- sePCSCFIPv4Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- sePCSCFIPv6Address
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, [398](#)
- secdns
 - unpack_wds_GetDefaultProfile_t, [378](#)
- secdnsv6
 - unpack_wds_GetDefaultProfile_t, [378](#)
- second
 - nas_timeInfo, [152](#)
 - nas_UniversalTime, [157](#)
- secondaryDNS
 - pack_wds_SetDefaultProfile_t, [224](#)
- SecondaryDNSV4
 - unpack_wds_SLQSGetRuntimeSettings_t, [391](#)
- SecondaryDNSV6
 - unpack_wds_SLQSGetRuntimeSettings_t, [392](#)
- secondaryHA
 - unpack_wds_GetMobileIPPProfile_t, [381](#)
- selNetwork
 - nas_servSystem, [142](#)
- selectedNetwork
 - NASServingSystemInfo, [173](#)
- selection
 - pack_swima_SLQSOMADMSendSelection_t, [212](#)
- ServerAddrList
 - unpack_wds_SLQSGetRuntimeSettings_t, [392](#)
- serviceDomain

- nas_RejectReasonTlv, 136
- servingCellId
 - nas_LTEInfoIntraFreq, 116
- ServingSystem
 - unpack_nas_SLQSGetServingSystem_t, 318
- sessionEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 393
- SessionId
 - pack_loc_Start_t, 186
 - pack_loc_Stop_t, 186
- sessionId
 - unpack_loc_PositionRpt_Ind_t, 302
- sessionInfo
 - pack_uim_ChangePin_t, 214
 - pack_uim_ReadTransparent_t, 216
 - pack_uim_SetPinProtection_t, 217
 - pack_uim_UnblockPin_t, 220
 - pack_uim_VerifyPin_t, 221
- SessionInfoConfig
 - unpack_swioma_SLQSOMADMAAlertCallback_ind_t, 365
- SessionInfoFota
 - unpack_swioma_SLQSOMADMAAlertCallback_ind_t, 365
- SessionInfoNotification
 - unpack_swioma_SLQSOMADMAAlertCallback_ind_t, 365
- SessionState
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 367
- sessionStatus
 - unpack_loc_PositionRpt_Ind_t, 302
 - unpack_omaDmNotificationsTlv_t, 339
- SessionType
 - pack_swioma_SLQSOMADMGetSessionInfo_t, 211
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 367
- sessionType
 - pack_swioma_SLQSOMADMCancelSession_t, 211
 - pack_swioma_SLQSOMADMStartSession_t, 213
 - uim_sessionInformation, 258
 - uim_UIMSessionInformation, 261
 - unpack_omaDmFotaTlv_t, 338
- set_fix_rate
 - pack_swiloc_SwiLocSetAutoStart_t, 210
- set_fix_type
 - pack_swiloc_SwiLocSetAutoStart_t, 210
- set_function
 - pack_swiloc_SwiLocSetAutoStart_t, 210
- set_max_dist
 - pack_swiloc_SwiLocSetAutoStart_t, 210
- set_max_time
 - pack_swiloc_SwiLocSetAutoStart_t, 210
- Severity
 - unpack_swioma_SLQSOMADMGetSessionInfo_t, 367
- severity
 - unpack_omaDmFotaTlv_t, 338
- shortName
 - unpack_nas_SLQSGetPLMNName_t, 316
- shortNameCI
 - unpack_nas_SLQSGetPLMNName_t, 316
- shortNameEn
 - unpack_nas_SLQSGetPLMNName_t, 316
- shortNameLen
 - unpack_nas_SLQSGetPLMNName_t, 316
- shortNameSB
 - unpack_nas_SLQSGetPLMNName_t, 316
- sid
 - nas_CDMAInfo, 84
 - unpack_nas_GetHomeNetwork_t, 307
- SigInd
 - LibPackUMTSReqQoSsigInd, 68
- sigInfo
 - nas_SccRxInfo, 141
- signalStrength
 - nas_SignalStrengthTlv, 143
- signalStrengthReqMask
 - unpack_nas_SLQSGetSignalStrength_t, 319
- SimCapability
 - unpack_dms_GetDeviceCap_t, 265
- simCapability
 - unpack_dms_GetDeviceCapabilities_t, 266
- sinr
 - hdrSSInfo, 38
 - nas_SLQSSignalStrengthsInformation, 145
 - tdscdmaSigInfoExt, 247
 - unpack_nas_SLQSGetSignalStrength_t, 319
- sinrDelta
 - nas_SLQSSignalStrengthsIndReq, 144
- sinrThresholdList
 - nas_SLQSSignalStrengthsIndReq, 144
- sinrThresholdListLen
 - nas_SLQSSignalStrengthsIndReq, 144
- sku_str
 - unpack_dms_GetFirmwareInfo_t, 268
- slot
 - pack_uim_SLQSUIPowerDown_t, 217
 - pack_uim_SLQSUIPowerUp_t, 218
- slot_t, 239
 - bICCID, 239
 - bICCIDLength, 239
 - bLogicalSlot, 239
 - uPhyCardStatus, 239
 - uPhySlotStatus, 239
- slotInf, 240
 - AppStatus, 241
 - cardState, 241
 - errorState, 241
 - numApp, 241
 - upinRetries, 241
 - upinState, 241

- upukRetries, 241
- SlotInfo
 - uim_cardStatus, 252
- slots_t, 241
 - uimSlotStatus, 241
- slotsstatusChange
 - unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 373
- SlqsProfile3GPP
 - unpackWdsProfileParam, 398
 - wds_profileInfo, 405
- SlqsProfile3GPP2
 - unpackWdsProfileParam, 398
 - wds_profileInfo, 405
- sms.h
 - LIBPACK_QMI_CBK_PARAM_NOCHANGE, 557
 - LIBPACK_QMI_CBK_PARAM_RESET, 557
 - LIBPACK_QMI_CBK_PARAM_SET, 557
- sms.h, 553
 - eqmiCbKsetStatus, 557
 - MAX_MSE_TWS_MSG, 555
 - MAX_SMS_LIST_SIZE, 555
 - pack_sms_SLQSDDeleteSMS, 558
 - pack_sms_SLQSGetSMS, 558
 - pack_sms_SLQSGetSMSList, 558
 - pack_sms_SLQSMModifySMSStatus, 559
 - pack_sms_SendSMS, 557
 - pack_sms_SetNewSMSCallback, 557
 - sMSCAddressInfo, 555
 - sMSEtwsMessageInfo, 555
 - sMSEtwsPlmnInfo, 556
 - sMSMTMessageInfo, 556
 - sMSMessageModelInfo, 556
 - sMSOnIMSInfo, 556
 - sMSTransferRouteMTMessageInfo, 556
 - unpack_sms_SLQSDDeleteSMS, 560
 - unpack_sms_SLQSGetSMS, 561
 - unpack_sms_SLQSGetSMSList, 561
 - unpack_sms_SLQSMModifySMSStatus, 561
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind, 562
 - unpack_sms_SendSMS, 559
 - unpack_sms_SetNewSMSCallback, 560
 - unpack_sms_SetNewSMSCallback_ind, 560
- smsOnIMS
 - sMSOnIMS, 245
- snr
 - lteSSInfo, 78
 - nas_SccRxInfo, 141
- snrlevel
 - nas_lteSnrInformation, 121
- soMask
 - qmiWDSDataBearerTechnology, 238
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- Source
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 368
- source
 - unpack_dms_GetNetworkTime_t, 272
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 278
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 279
- sourceIPMask
 - LibPackTFTIDParams, 65
- SourceLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 368
- spc
 - pack_nas_SetACCOLC_t, 187
 - pack_wds_SetMobileIPProfile_t, 226
- spn
 - unpack_nas_SLQSGetPLMNName_t, 316
- spnEncoding
 - unpack_nas_SLQSGetPLMNName_t, 316
- spnLength
 - unpack_nas_SLQSGetPLMNName_t, 316
- srcPortRangeEnd
 - LibPackTFTIDParams, 65
- srcPortRangeStart
 - LibPackTFTIDParams, 65
- srvCapability
 - nas_detailSvcInfo, 95
 - nas_sysInfoCommon, 148
- srvCapabilityValid
 - nas_sysInfoCommon, 148
- srvDomain
 - nas_sysInfoCommon, 148
- SrvDomainPref
 - NASServDomainPrefTlv, 171
- srvDomainValid
 - nas_sysInfoCommon, 148
- srvStatus
 - nas_detailSvcInfo, 95
 - nas_GSMSrvStatusInfo, 102
 - nas_SrvStatusInfo, 146
- srxlev
 - nas_cellParams, 89
 - nas_gsmCellInfo, 100
 - nas_umtsLTENbrCell, 156
 - nas_wcdmaCellInfo, 158
- state
 - unpack_dms_GetActivationState_t, 263
 - unpack_omaDmConfigTlv_t, 336
 - unpack_omaDmFotaTlv_t, 338
 - unpack_qos_QosFlowInfoState_t, 344
- statmask
 - pack_wds_GetPacketStatus_t, 223
- StatsMask
 - transferStatInd, 248
- StatsPeriod
 - transferStatInd, 248
- Status
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 368
- status

- pack_sms_SetNewSMSCallback_t, 206
- unpack_loc_BestAvailPos_Ind_t, 295
- unpack_loc_SetExtPowerConfig_Ind_t, 303
- unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 348
- unpack_qos_SLQSSetQosStatusCallback_ind_t, 350
- storageIndex
 - FMSImageldElement, 36
- storageType
 - pack_sms_SLQSDDeleteSMS_t, 207
 - pack_sms_SLQSGetSMS_t, 207
 - pack_sms_SLQSGetSMSList_t, 208
 - pack_sms_SLQSMModifySMSStatus_t, 209
 - sMSMTMessage, 245
 - unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 363
- Streaming Download Protocol (sdp), 20
- String
 - unpack_dms_GetDeviceHardwareRev_t, 266
 - unpack_dms_GetDeviceMfr_t, 267
 - unpack_dms_GetFSN_t, 270
 - unpack_dms_UIMGetICCID_t, 287
- stringSize
 - unpack_dms_GetDeviceHardwareRev_t, 266
 - unpack_dms_GetDeviceMfr_t, 267
 - unpack_dms_UIMGetICCID_t, 287
- subnetMask
 - unpack_qos_IPv4Addr_t, 340
- SubnetMaskV4
 - unpack_wds_SLQSGetRuntimeSettings_t, 392
- SupUSBComps
 - unpack_dms_GetUSBComp_t, 274
- svc
 - pack_qmi_t, 203
- sw1
 - uim_cardResult, 251
- sw2
 - uim_cardResult, 251
- SwiDataTypes.h, 562
 - BOOL, 563
 - BYTE, 563
 - CHAR, 563
 - FLOAT, 563
 - INT32, 563
 - INT8, 563
 - LPCSTR, 563
 - SHORT, 563
 - SWI_API, 563
 - ULONG, 563
 - ULONGLONG, 563
 - UNUSEDPARAM, 563
 - USHORT, 563
 - WORD, 563
- swiloc.h, 563
 - pack_swiloc_SwiLocGetAutoStart, 564
 - pack_swiloc_SwiLocSetAutoStart, 564
 - unpack_swiloc_SwiLocGetAutoStart, 564
 - unpack_swiloc_SwiLocSetAutoStart, 565
- swioma.h, 565
 - pack_swioma_SLQSOMADMAAlertCallback, 566
 - pack_swioma_SLQSOMADMCancelSession, 567
 - pack_swioma_SLQSOMADMGetSessionInfo, 567
 - pack_swioma_SLQSOMADMGetSettings, 568
 - pack_swioma_SLQSOMADMSendSelection, 568
 - pack_swioma_SLQSOMADMSetSettings, 569
 - pack_swioma_SLQSOMADMStartSession, 570
 - unpack_swioma_SLQSOMADMAAlertCallback, 570
 - unpack_swioma_SLQSOMADMAAlertCallback_ind, 571
 - unpack_swioma_SLQSOMADMCancelSession, 571
 - unpack_swioma_SLQSOMADMGetSessionInfo, 572
 - unpack_swioma_SLQSOMADMGetSettings, 572
 - unpack_swioma_SLQSOMADMSendSelection, 573
 - unpack_swioma_SLQSOMADMSetSettings, 573
 - unpack_swioma_SLQSOMADMStartSession, 573
- sysInfoCDMA
 - nas_CDMA SysInfo, 88
- sysInfoGSM
 - nas_GSM SysInfo, 105
- sysInfoHDR
 - nas_HDR SysInfo, 109
- sysInfoLTE
 - nas_LTE SysInfo, 124
- sysInfoWCDMA
 - nas_WCDMA SysInfo, 163
- system
 - loc_SV, 76
- SystemID
 - unpack_nas_SLQSGetServingSystem_t, 318
- systemID
 - nas_CDMA SysInfo, 88
- systemMode
 - nas_CommInfo, 91
- szCarrier_str
 - _libSDP_FirmwareInfo_, 23
- szCarrierPriversion_str
 - _libSDP_FirmwareInfo_, 24
- szFwversion_str
 - _libSDP_FirmwareInfo_, 24
- szModelid_str
 - _libSDP_FirmwareInfo_, 24
- szPackageid_str
 - _libSDP_FirmwareInfo_, 24
- szSku_str
 - _libSDP_FirmwareInfo_, 24
- TCPDStPort
 - unpack_qos_swiQosFilter_t, 353
- TCPSrcPort
 - unpack_qos_swiQosFilter_t, 353
- TDSCDMAECIOThreshListLen
 - nas_TDSCDMAECIOThresh, 149
- TDSCDMARSCPThreshListLen

- nas_TDSCDMARSCPThresh, [149](#)
- TDSCDMARSSIThreshListLen
 - nas_TDSCDMARSSIThresh, [150](#)
- TDSCDMASINRThreshListLen
 - nas_TDSCDMASINRThresh, [151](#)
- TRMessageTlv
 - unpack_sms_SetNewSMSCallback_ind_t, [361](#)
- TXChan
 - nas_LTEInfo, [113](#)
- tXDroppedCount
 - unpack_wds_GetPacketStatus_t, [383](#)
- tXOKBytesLastCall
 - unpack_wds_GetPacketStatus_t, [383](#)
- tXOkBytesCount
 - unpack_wds_GetPacketStatus_t, [383](#)
- tXPacketErrors
 - unpack_wds_GetPacketStatus_t, [383](#)
- tXPacketOverflows
 - unpack_wds_GetPacketStatus_t, [383](#)
- tXPacketSuccesses
 - unpack_wds_GetPacketStatus_t, [383](#)
- Tables, [21](#)
- tac
 - nas_LTEInfoIntrafreq, [116](#)
 - nas_LTESysInfo, [124](#)
- tacValid
 - nas_LTESysInfo, [124](#)
- TdsBandCapability
 - unpack_dms_SLQSGetBandCapability_t, [283](#)
- tdscdmaSigInfoExt, [246](#)
 - ecio, [247](#)
 - rscp, [247](#)
 - rssr, [247](#)
 - sinr, [247](#)
- techName
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, [393](#)
- Technology
 - unpack_wds_SLQSGetRuntimeSettings_t, [392](#)
- TechnologyPref
 - pack_nas_SetNetworkPreference_t, [188](#)
- temperature
 - nas_CommInfo, [91](#)
- threshGsmHigh
 - nas_lteGsmCellInfo, [111](#)
- threshGsmLow
 - nas_lteGsmCellInfo, [111](#)
- threshServingLow
 - nas_LTEInfoIntrafreq, [116](#)
- threshXHigh
 - nas_infoInterFreq, [110](#)
- threshXLow
 - nas_infoInterFreq, [110](#)
- threshXhigh
 - nas_lteWcdmaCellInfo, [125](#)
- threshXlow
 - nas_lteWcdmaCellInfo, [125](#)
- Time
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [368](#)
- time
 - NASTimeInfoTlv, [173](#)
- TimeLength
 - unpack_swima_SLQSOMADMGetSessionInfo_t, [368](#)
- timeTlv
 - NASQmiCbkNasSwiOTAMessageInd, [170](#)
- timeZone
 - nas_timeInfo, [152](#)
- timeout
 - pack_qmi_t, [203](#)
- timestamp
 - unpack_dms_GetNetworkTime_t, [272](#)
- timingAdvance
 - nas_GERANInfo, [98](#)
- TlvPresent
 - dms_ActivationStatusTlv, [30](#)
 - dms_OperatingModeTlv, [30](#)
 - eTWSPLMNInfoTlv, [34](#)
 - messageModeTlv, [79](#)
 - nas_PhyCaAggPcellInfo, [128](#)
 - nas_PhyCaAggScellIDBw, [129](#)
 - nas_PhyCaAggScellIndex, [129](#)
 - nas_PhyCaAggScellIndType, [130](#)
 - nas_PhyCaAggScellInfo, [133](#)
 - nas_RejectReasonTlv, [136](#)
 - nas_RFInfoTlv, [137](#)
 - nas_SccRxInfo, [141](#)
 - nas_SignalStrengthTlv, [143](#)
 - nas_SLQSSignalStrengthsTlv, [145](#)
 - nas_timeInfo, [152](#)
 - NASBandPreferenceTlv, [163](#)
 - NASEmergencyModeTlv, [164](#)
 - NASGWAcqOrderPrefTlv, [164](#)
 - NASLTEBandPreferenceTlv, [165](#)
 - NASLteNasReleaseInfoTlv, [165](#)
 - NASModePreferenceTlv, [165](#)
 - NASNetSelPreferenceTlv, [166](#)
 - NASOTAMessageTlv, [166](#)
 - NASPhyCaAggPcellInfo, [167](#)
 - NASPhyCaAggScellIDBw, [167](#)
 - NASPhyCaAggScellIndex, [168](#)
 - NASPhyCaAggScellIndType, [169](#)
 - NASPhyCaAggScellInfo, [170](#)
 - NASPRLPreferenceTlv, [170](#)
 - NASRoamPreferenceTlv, [171](#)
 - NASServDomainPrefTlv, [171](#)
 - NASTimeInfoTlv, [173](#)
 - newMTMessageTlv, [174](#)
 - sMSCAddressTlv, [242](#)
 - sMSEtwSMessageTlv, [243](#)
 - sMSOnIMSTlv, [245](#)
 - transferRouteMessageTlv, [247](#)
- Tlvresult
 - pack_dms_GetCustFeaturesV2_t, [174](#)
 - pack_dms_SetCustFeaturesV2_t, [176](#)

- pack_dms_SetPower_t, 177
- pack_dms_SetUSBComp_t, 177
- pack_dms_UIMGetICCID_t, 178
- pack_fms_GetImagesPreference_t, 178
- pack_fms_GetStoredImages_t, 179
- pack_fms_SetImagesPreference_t, 180
- pack_loc_Delete_Assist_Data_t, 181
- pack_loc_EventRegister_t, 183
- pack_loc_SetExtPowerState_t, 183
- pack_loc_SetOperationMode_t, 184
- pack_loc_SLQSLOCGetBestAvailPos_t, 184
- pack_loc_Start_t, 186
- pack_loc_Stop_t, 186
- pack_nas_SetNetworkPreference_t, 188
- pack_uim_ChangePin_t, 214
- pack_uim_ReadTransparent_t, 216
- pack_uim_SetPinProtection_t, 217
- pack_uim_UnblockPin_t, 220
- pack_uim_VerifyPin_t, 221
- unpack_dms_GetBandCapability_t, 264
- unpack_dms_GetCrashAction_t, 264
- unpack_dms_GetCustFeature_t, 264
- unpack_dms_GetCustFeaturesV2_t, 265
- unpack_dms_GetDeviceCap_t, 265
- unpack_dms_GetDeviceHardwareRev_t, 266
- unpack_dms_GetDeviceMfr_t, 267
- unpack_dms_GetDeviceSerialNumbers_t, 267
- unpack_dms_GetFirmwareInfo_t, 268
- unpack_dms_GetFirmwareRevision_t, 269
- unpack_dms_GetFirmwareRevisions_t, 270
- unpack_dms_GetFSN_t, 270
- unpack_dms_GetIMSI_t, 270
- unpack_dms_GetManufacturer_t, 271
- unpack_dms_GetModelID_t, 271
- unpack_dms_GetNetworkTime_t, 272
- unpack_dms_GetOfflineReason_t, 273
- unpack_dms_GetPower_t, 273
- unpack_dms_GetPRLVersion_t, 274
- unpack_dms_GetUSBComp_t, 274
- unpack_dms_GetVoiceNumber_t, 275
- unpack_dms_SetCustFeature_t, 275
- unpack_dms_SetCustFeaturesV2_t, 276
- unpack_dms_SetEventReport_ind_t, 276
- unpack_dms_SetEventReport_t, 277
- unpack_dms_SetFirmwarePreference_t, 277
- unpack_dms_SetPower_t, 277
- unpack_dms_SetUSBComp_t, 277
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 278
- unpack_dms_SLQSDmsSwiGetResetInfo_t, 279
- unpack_dms_SLQSDmsSwiIndicationRegister_t, 280
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 283
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, 284
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 284
- unpack_dms_SLQSSwiGetFwUpdateStatus_t, 286
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, 287
- unpack_dms_UIMGetICCID_t, 287
- unpack_fms_GetImagesPreference_t, 288
- unpack_fms_GetStoredImages_t, 289
- unpack_fms_SetImagesPreference_t, 289
- unpack_loc_BestAvailPos_Ind_t, 295
- unpack_loc_Delete_Assist_Data_t, 296
- unpack_loc_EngineState_Ind_t, 296
- unpack_loc_EventRegister_t, 297
- unpack_loc_PositionRpt_Ind_t, 302
- unpack_loc_SetExtPowerConfig_Ind_t, 303
- unpack_loc_SetExtPowerState_t, 304
- unpack_loc_SetOperationMode_t, 304
- unpack_loc_SLQSLOCGetBestAvailPos_t, 304
- unpack_loc_Start_t, 305
- unpack_loc_Stop_t, 305
- unpack_nas_GetNetworkPreference_t, 308
- unpack_nas_SetNetworkPreference_t, 313
- unpack_nas_SetServingSystemCallback_ind_t, 314
- unpack_nas_SlqsGetLTECphyCAInfo_t, 315
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 331
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 331
- unpack_uim_ChangePin_t, 370
- unpack_uim_GetCardStatus_t, 371
- unpack_uim_ReadTransparent_t, 372
- unpack_uim_SetPinProtection_t, 373
- unpack_uim_UnblockPin_t, 375
- unpack_uim_VerifyPin_t, 376
- unpack_wds_SLQSCreateProfile_t, 385
- unpack_wds_SLQSGetProfileSettings_t, 390
- unpack_wds_SLQSSetIPFamilyPreference_t, 392
- TokenBucket
 - unpack_qos_swiQosFlow_t, 357
- tokenRate
 - unpack_qos_tokenBucket_t, 358
- tosMask
 - LibPackTFTIDParams, 65
- total_rx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 347
- total_rx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 347
- total_tx_bytes
 - unpack_qos_SLQSQosSwiReadDataStats_t, 347
- total_tx_bytes_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 347
- total_tx_pkt
 - unpack_qos_SLQSQosSwiReadDataStats_t, 347
- total_tx_pkt_drp
 - unpack_qos_SLQSQosSwiReadDataStats_t, 347
- TrackAreaCode
 - unpack_nas_SLQSGetServingSystem_t, 318
- TrafficClass
 - unpack_qos_swiQosFlow_t, 357
- trafficClass
 - LibPackUMTSQoS, 67

- wds_UMTSMinQoS, 408
- trafficPriority
 - LibPackUMTSQoS, 67
 - wds_UMTSMinQoS, 408
- TranDstPort
 - unpack_qos_swiQosFilter_t, 353
- TranSrcPort
 - unpack_qos_swiQosFilter_t, 353
- transactionID
 - sMSTransferRouteMTMessage, 246
- transferDelay
 - LibPackUMTSQoS, 67
 - wds_UMTSMinQoS, 408
- TransferRouteMTMessageInfo
 - transferRouteMessageTlv, 247
- transferRouteMessageTlv, 247
 - TlvPresent, 247
 - TransferRouteMTMessageInfo, 247
- transferStatInd, 247
 - StatsMask, 248
 - StatsPeriod, 248
- transferStats
 - pack_wds_SLQSSetWdsEventCallback_t, 233
- trueSrvStatus
 - nas_GSMSrvStatusInfo, 102
- tx_bytes
 - unpack_QosFlowStat_t, 359
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- tx_bytes_drp
 - unpack_QosFlowStat_t, 359
- tx_pkt
 - unpack_QosFlowStat_t, 359
- tx_pkt_drp
 - unpack_QosFlowStat_t, 359
- tx_pkts
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, 395
- txOKBytesCount
 - unpack_wds_SLQSGetDUNCallInfo_t, 389
- TxQFilter
 - unpack_qos_QosFlowInfo_t, 343
- TxQFlowGranted
 - unpack_qos_QosFlowInfo_t, 343
- type
 - pack_wds_GetDefaultProfileNum_t, 222
 - pack_wds_SetDefaultProfileNum_t, 225
 - unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 278
 - unpack_dms_SLQSDmsSwiGetResetInfo_t, 279
 - unpack_qmi_t, 339
- u16PRLVersion
 - unpack_dms_GetPRLVersion_t, 274
- u8PRLPreference
 - unpack_dms_GetPRLVersion_t, 274
- UDPDstPort
 - unpack_qos_swiQosFilter_t, 353
- UDPSrcPort
 - unpack_qos_swiQosFilter_t, 353
- ULONG
 - SwiDataTypes.h, 563
- ULONGLONG
 - SwiDataTypes.h, 563
- UMTSGrantedQoS
 - unpack_wds_SLQSGetRuntimeSettings_t, 392
- UMTSInstInfo
 - nas_UMTSInfo, 154
- UMTSLTENbrCell
 - nas_WCDMAInfoLTENeighborCell, 159
- UMTSReqQoS
 - LibPackUMTSReqQoSSigInd, 68
- UNIQUE_ID_LEN
 - dms.h, 421
- UNUSEDPARAM
 - common.h, 413
 - SwiDataTypes.h, 563
- uPhyCardStatus
 - slot_t, 239
- uPhySlotStatus
 - slot_t, 239
- USBComp
 - pack_dms_SetUSBComp_t, 177
 - unpack_dms_GetUSBComp_t, 275
- USHORT
 - SwiDataTypes.h, 563
- uarfcn
 - nas_lteWcdmaCellInfo, 125
 - nas_UMTSInfo, 154
- ueInIdle
 - nas_LTEInfoInterfreq, 114
 - nas_LTEInfoIntrafreq, 116
 - nas_LTEInfoNeighboringGSM, 117
 - nas_LTEInfoNeighboringWCDMA, 118
- uim.h, 574
 - MAX_ICCID_LENGTH, 576
 - MAX_NO_OF_SLOTS, 576
 - MAX_SLOTS_STATUS, 576
 - pack_uim_ChangePin, 576
 - pack_uim_GetCardStatus, 577
 - pack_uim_ReadTransparent, 577
 - pack_uim_SLQSUIEventRegister, 578
 - pack_uim_SLQSUIGetSlotsStatus, 578
 - pack_uim_SLQSUIPowerDown, 578
 - pack_uim_SLQSUIPowerUp, 579
 - pack_uim_SLQSUISwitchSlot, 579
 - pack_uim_SetPinProtection, 577
 - pack_uim_UnblockPin, 579
 - pack_uim_VerifyPin, 580
 - unpack_uim_ChangePin, 580
 - unpack_uim_GetCardStatus, 581
 - unpack_uim_ReadTransparent, 581
 - unpack_uim_SLQSUIEventRegister, 582
 - unpack_uim_SLQSUIGetSlotsStatus, 582
 - unpack_uim_SLQSUIPowerDown, 583
 - unpack_uim_SLQSUIPowerUp, 583

- unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind, 583
- unpack_uim_SLQSUIMSwitchSlot, 584
- unpack_uim_SetPinProtection, 581
- unpack_uim_SetUimSlotStatusChangeCallback_ind, 582
- unpack_uim_UnblockPin, 584
- unpack_uim_VerifyPin, 584
- uim_UIMSessionInformation, 260
 - aid, 261
 - aidLength, 261
 - sessionType, 261
- uim_appStatus, 248
 - aidLength, 250
 - aidVal, 250
 - appState, 250
 - appType, 250
 - persoFeature, 250
 - persoRetries, 250
 - persoState, 250
 - persoUnblockRetries, 250
 - pin1Retries, 251
 - pin1State, 251
 - pin2Retries, 251
 - pin2State, 251
 - puk1Retries, 251
 - puk2Retries, 251
 - univPin, 251
- uim_cardResult, 251
 - sw1, 251
 - sw2, 251
- uim_cardStatus, 251
 - index1xPri, 252
 - index1xSec, 252
 - indexGwPri, 252
 - indexGwSec, 252
 - numSlot, 252
 - SlotInfo, 252
- uim_changeUIMPIN, 253
 - oldPINLen, 253
 - oldPINVal, 253
 - pinID, 253
 - pinLen, 253
 - pinVal, 253
- uim_encryptedPIN1, 253
 - pin1Len, 254
 - pin1Val, 254
- uim_fileInfo, 254
 - fileID, 255
 - path, 255
 - pathLen, 255
- uim_hotSwapStatus, 255
 - hotSwap, 255
 - hotSwapLength, 255
- uim_readResult, 255
 - content, 256
 - contentLen, 256
- uim_readTransparentInfo, 256
 - length, 256
 - offset, 256
- uim_remainingRetries, 256
 - unblockLeft, 257
 - verifyLeft, 257
- uim_sessionInformation, 257
 - aid, 258
 - aidLength, 258
 - sessionType, 258
- uim_setPINProtection, 258
 - pinID, 259
 - pinLength, 259
 - pinOperation, 259
 - pinValue, 259
- uim_slotInfo, 259
 - AppStatus, 260
 - cardState, 260
 - errorState, 260
 - numApp, 260
 - upinRetries, 260
 - upinState, 260
 - upukRetries, 260
- uim_unblockUIMPIN, 261
 - newPINLen, 262
 - newPINVal, 262
 - pinID, 262
 - pukLen, 262
 - pukVal, 262
- uim_verifyUIMPIN, 262
 - pinID, 263
 - pinLen, 263
 - pinVal, 263
- uimSlotStatus
 - slots_t, 241
- ulMask
 - rmTrasnferStaticsReq, 238
- ulPhysicalSlot
 - pack_uim_SLQSUIMSwitchSlot_t, 219
- umtsEcio
 - nas_UMTSinstInfo, 155
- umtsInst
 - nas_UMTSInfo, 154
- umtsLTENbrCellLen
 - nas_WCDMAInfoLTENeighborCell, 159
- umtsPsc
 - nas_UMTSinstInfo, 155
- umtsRscp
 - nas_UMTSinstInfo, 155
- umtsUarfcn
 - nas_UMTSinstInfo, 155
- UnPackGetProfileSettingOut, 398
 - curProfile, 398
 - pExtErrCode, 398
- unblockLeft
 - uim_remainingRetries, 257
- uniqueID
 - image_info_t, 38
- univPin

- appStats, [27](#)
- uim_appStatus, [251](#)
- universalTime
 - unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, [329](#)
- unpack_QosFlowStat_t, [358](#)
 - bearerId, [359](#)
 - tx_bytes, [359](#)
 - tx_bytes_drp, [359](#)
 - tx_pkt, [359](#)
 - tx_pkt_drp, [359](#)
- unpack_dms_GetActivationState
 - dms.h, [436](#)
- unpack_dms_GetActivationState_t, [263](#)
 - state, [263](#)
- unpack_dms_GetBandCapability
 - dms.h, [436](#)
- unpack_dms_GetBandCapability_t, [263](#)
 - BandCapability, [263](#)
 - Tlvresult, [264](#)
- unpack_dms_GetCrashAction
 - dms.h, [436](#)
- unpack_dms_GetCrashAction_t, [264](#)
 - DevCrashState, [264](#)
 - Tlvresult, [264](#)
- unpack_dms_GetCustFeature
 - dms.h, [437](#)
- unpack_dms_GetCustFeature_t, [264](#)
 - DHCPRelayEnabled, [264](#)
 - DisableIMSI, [264](#)
 - GPSPMP, [264](#)
 - GPSSel, [264](#)
 - GpsEnable, [264](#)
 - IPFamSupport, [264](#)
 - IsVoiceEnabled, [264](#)
 - RMAutoConnect, [264](#)
 - SMSSupport, [264](#)
 - Tlvresult, [264](#)
- unpack_dms_GetCustFeaturesV2
 - dms.h, [437](#)
- unpack_dms_GetCustFeaturesV2_t, [265](#)
 - GetCustomFeatureV2, [265](#)
 - Tlvresult, [265](#)
- unpack_dms_GetDeviceCap
 - dms.h, [437](#)
- unpack_dms_GetDeviceCap_t, [265](#)
 - DataServiceCapability, [265](#)
 - MaxRXChannelRate, [265](#)
 - MaxTXChannelRate, [265](#)
 - Radiolfaces, [265](#)
 - RadiolfacesSize, [265](#)
 - SimCapability, [265](#)
 - Tlvresult, [265](#)
- unpack_dms_GetDeviceCapabilities
 - dms.h, [438](#)
- unpack_dms_GetDeviceCapabilities_t, [266](#)
 - dataServiceCaCapability, [266](#)
 - maxRxChannelRate, [266](#)
 - maxTxChannelRate, [266](#)
 - Radiolfaces, [266](#)
 - radiolfacesSize, [266](#)
 - simCapability, [266](#)
- unpack_dms_GetDeviceHardwareRev
 - dms.h, [438](#)
- unpack_dms_GetDeviceHardwareRev_t, [266](#)
 - String, [266](#)
 - stringSize, [266](#)
 - Tlvresult, [266](#)
- unpack_dms_GetDeviceMfr
 - dms.h, [438](#)
- unpack_dms_GetDeviceMfr_t, [267](#)
 - String, [267](#)
 - stringSize, [267](#)
 - Tlvresult, [267](#)
- unpack_dms_GetDeviceSerialNumbers
 - dms.h, [439](#)
- unpack_dms_GetDeviceSerialNumbers_t, [267](#)
 - ESNString, [267](#)
 - esnSize, [267](#)
 - IMEIString, [267](#)
 - imeiSize, [267](#)
 - imeiSvnSize, [267](#)
 - ImeiSvnString, [267](#)
 - MEIDString, [267](#)
 - meidSize, [267](#)
 - Tlvresult, [267](#)
- unpack_dms_GetFSN
 - dms.h, [440](#)
- unpack_dms_GetFSN_t, [270](#)
 - String, [270](#)
 - Tlvresult, [270](#)
- unpack_dms_GetFirmwareInfo
 - dms.h, [439](#)
- unpack_dms_GetFirmwareInfo_t, [268](#)
 - appversion_str, [268](#)
 - bootversion_str, [268](#)
 - carrier_str, [268](#)
 - cur_carr_name, [268](#)
 - cur_carr_rev, [268](#)
 - modelid_str, [268](#)
 - packageid_str, [268](#)
 - priversion_str, [268](#)
 - sku_str, [268](#)
 - Tlvresult, [268](#)
- unpack_dms_GetFirmwareRevision
 - dms.h, [439](#)
- unpack_dms_GetFirmwareRevision_t, [269](#)
 - AMSSString, [269](#)
 - amssSize, [269](#)
 - PRISString, [269](#)
 - Tlvresult, [269](#)
- unpack_dms_GetFirmwareRevisions
 - dms.h, [440](#)
- unpack_dms_GetFirmwareRevisions_t, [269](#)
 - AMSSString, [269](#)
 - amssSize, [269](#)

- bootSize, [269](#)
- BootString, [269](#)
- PRIString, [269](#)
- priSize, [269](#)
- Tlvresult, [270](#)
- unpack_dms_GetHardwareRevision
 - dms.h, [440](#)
- unpack_dms_GetHardwareRevision_t, [270](#)
 - hwVer, [270](#)
- unpack_dms_GetIMSI
 - dms.h, [441](#)
- unpack_dms_GetIMSI_t, [270](#)
 - imsi, [270](#)
 - Tlvresult, [270](#)
- unpack_dms_GetManufacturer
 - dms.h, [441](#)
- unpack_dms_GetManufacturer_t, [270](#)
 - manufacturer, [271](#)
 - Tlvresult, [271](#)
- unpack_dms_GetModelID
 - dms.h, [441](#)
- unpack_dms_GetModelID_t, [271](#)
 - modelid, [271](#)
 - Tlvresult, [271](#)
- unpack_dms_GetNetworkTime
 - dms.h, [442](#)
- unpack_dms_GetNetworkTime_t, [271](#)
 - source, [272](#)
 - timestamp, [272](#)
 - Tlvresult, [272](#)
- unpack_dms_GetOfflineReason
 - dms.h, [442](#)
- unpack_dms_GetOfflineReason_t, [272](#)
 - pReasonMask, [273](#)
 - pbPlatform, [273](#)
 - Tlvresult, [273](#)
- unpack_dms_GetPRLVersion
 - dms.h, [443](#)
- unpack_dms_GetPRLVersion_t, [273](#)
 - Tlvresult, [274](#)
 - u16PRLVersion, [274](#)
 - u8PRLPreference, [274](#)
- unpack_dms_GetPower
 - dms.h, [442](#)
- unpack_dms_GetPower_t, [273](#)
 - HardwareControlledMode, [273](#)
 - OfflineReason, [273](#)
 - OperationMode, [273](#)
 - Tlvresult, [273](#)
- unpack_dms_GetSerialNumbers
 - dms.h, [443](#)
- unpack_dms_GetSerialNumbers_t, [274](#)
 - esn, [274](#)
 - imei_no, [274](#)
 - imeisv_svn, [274](#)
 - meid, [274](#)
- unpack_dms_GetUSBComp
 - dms.h, [443](#)
- unpack_dms_GetUSBComp_t, [274](#)
 - NumSupUSBComps, [274](#)
 - SupUSBComps, [274](#)
 - Tlvresult, [274](#)
 - USBComp, [275](#)
- unpack_dms_GetVoiceNumber
 - dms.h, [444](#)
- unpack_dms_GetVoiceNumber_t, [275](#)
 - MIN, [275](#)
 - minSize, [275](#)
 - Tlvresult, [275](#)
 - VoiceNumber, [275](#)
 - voiceNumberSize, [275](#)
- unpack_dms_SLQSDmsSwiGetResetInfo
 - dms.h, [447](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind
 - dms.h, [447](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, [277](#)
 - source, [278](#)
 - Tlvresult, [278](#)
 - type, [278](#)
- unpack_dms_SLQSDmsSwiGetResetInfo_t, [278](#)
 - source, [279](#)
 - Tlvresult, [279](#)
 - type, [279](#)
- unpack_dms_SLQSDmsSwiIndicationRegister
 - dms.h, [448](#)
- unpack_dms_SLQSDmsSwiIndicationRegister_t, [279](#)
 - Tlvresult, [280](#)
- unpack_dms_SLQSGetBandCapability
 - dms.h, [448](#)
- unpack_dms_SLQSGetBandCapability_t, [280](#)
 - bandCapability, [283](#)
 - LteBandCapability, [283](#)
 - TdsBandCapability, [283](#)
- unpack_dms_SLQSSwiClearDyingGaspStatistics
 - dms.h, [448](#)
- unpack_dms_SLQSSwiClearDyingGaspStatistics_t, [283](#)
 - Tlvresult, [283](#)
- unpack_dms_SLQSSwiGetDyingGaspCfg
 - dms.h, [449](#)
- unpack_dms_SLQSSwiGetDyingGaspCfg_t, [283](#)
 - pGetDyingGaspCfg, [284](#)
 - Tlvresult, [284](#)
- unpack_dms_SLQSSwiGetDyingGaspStatistics
 - dms.h, [449](#)
- unpack_dms_SLQSSwiGetDyingGaspStatistics_t, [284](#)
 - pGetDyingGaspStatistics, [284](#)
 - Tlvresult, [284](#)
- unpack_dms_SLQSSwiGetFirmwareCurr
 - dms.h, [449](#)
- unpack_dms_SLQSSwiGetFirmwareCurr_t, [284](#)
 - carrier, [285](#)
 - fwvers, [285](#)
 - numEntries, [285](#)
 - pCurrImgInfo, [285](#)
 - pkgver, [285](#)

- priver, [285](#)
- unpack_dms_SLQSSwiGetFwUpdateStatus
 - dms.h, [450](#)
- unpack_dms_SLQSSwiGetFwUpdateStatus_t, [285](#)
 - imgType, [286](#)
 - logString, [286](#)
 - refData, [286](#)
 - refString, [286](#)
 - ResCode, [286](#)
 - Tlvresult, [286](#)
- unpack_dms_SLQSSwiSetDyingGaspCfg
 - dms.h, [450](#)
- unpack_dms_SLQSSwiSetDyingGaspCfg_t, [287](#)
 - Tlvresult, [287](#)
- unpack_dms_SetCrashAction
 - dms.h, [444](#)
- unpack_dms_SetCrashAction_t, [275](#)
 - notused, [275](#)
- unpack_dms_SetCustFeature
 - dms.h, [444](#)
- unpack_dms_SetCustFeature_t, [275](#)
 - Tlvresult, [275](#)
- unpack_dms_SetCustFeaturesV2
 - dms.h, [445](#)
- unpack_dms_SetCustFeaturesV2_t, [276](#)
 - Tlvresult, [276](#)
- unpack_dms_SetEventReport
 - dms.h, [445](#)
- unpack_dms_SetEventReport_ind
 - dms.h, [445](#)
- unpack_dms_SetEventReport_ind_t, [276](#)
 - ActivationStatusTlv, [276](#)
 - OperatingModeTlv, [276](#)
 - Tlvresult, [276](#)
- unpack_dms_SetEventReport_t, [277](#)
 - Tlvresult, [277](#)
- unpack_dms_SetFirmwarePreference
 - dms.h, [446](#)
- unpack_dms_SetFirmwarePreference_t, [277](#)
 - Tlvresult, [277](#)
- unpack_dms_SetPower
 - dms.h, [446](#)
- unpack_dms_SetPower_t, [277](#)
 - Tlvresult, [277](#)
- unpack_dms_SetUSBComp
 - dms.h, [446](#)
- unpack_dms_SetUSBComp_t, [277](#)
 - Tlvresult, [277](#)
- unpack_dms_UIMGetICCID
 - dms.h, [451](#)
- unpack_dms_UIMGetICCID_t, [287](#)
 - String, [287](#)
 - stringSize, [287](#)
 - Tlvresult, [287](#)
- unpack_fms_GetImagesPreference
 - fms.h, [453](#)
- unpack_fms_GetImagesPreference_t, [288](#)
 - ImageListSize, [288](#)
 - pImageList, [288](#)
 - Tlvresult, [288](#)
- unpack_fms_GetStoredImages
 - fms.h, [454](#)
- unpack_fms_GetStoredImages_t, [288](#)
 - imageList, [289](#)
 - imagelistSize, [289](#)
 - Tlvresult, [289](#)
- unpack_fms_SetImagesPreference
 - fms.h, [454](#)
- unpack_fms_SetImagesPreference_t, [289](#)
 - ImageTypes, [289](#)
 - ImageTypesSize, [289](#)
 - Tlvresult, [289](#)
- unpack_loc_BestAvailPos_Ind
 - loc.h, [468](#)
- unpack_loc_BestAvailPos_Ind_t, [289](#)
 - pAltitudeWrtEllipsoid, [294](#)
 - pAltitudeWrtMeanSeaLevel, [294](#)
 - pGpsTime, [294](#)
 - pHeading, [294](#)
 - pHeadingUnc, [294](#)
 - pHorCirConf, [294](#)
 - pHorEllpConf, [294](#)
 - pHorReliability, [294](#)
 - pHorUncCircular, [294](#)
 - pHorUncEllipseOrientAzimuth, [294](#)
 - pHorUncEllipseSemiMajor, [294](#)
 - pHorUncEllipseSemiMinor, [294](#)
 - pLatitude, [295](#)
 - pLongitude, [295](#)
 - pMagneticDeviation, [295](#)
 - pPrecisionDilution, [295](#)
 - pSensorDataUsage, [295](#)
 - pSpeedHorizontal, [295](#)
 - pSpeedUnc, [295](#)
 - pSpeedVertical, [295](#)
 - pSpeedVerticalUnc, [295](#)
 - pSvUsedforFix, [295](#)
 - pTechnologyMask, [295](#)
 - pTimeSrc, [295](#)
 - pTimeUnc, [295](#)
 - pTimestampUtc, [295](#)
 - pVertConfidence, [295](#)
 - pVertReliability, [295](#)
 - pVertUnc, [295](#)
 - pXid, [295](#)
 - status, [295](#)
 - Tlvresult, [295](#)
- unpack_loc_Delete_Assist_Data_t, [295](#)
 - Tlvresult, [296](#)
- unpack_loc_DeleteAssistData
 - loc.h, [468](#)
- unpack_loc_EngineState_Ind
 - loc.h, [468](#)
- unpack_loc_EngineState_Ind_t, [296](#)
 - engineState, [296](#)
 - Tlvresult, [296](#)

- unpack_loc_EventRegister
 - loc.h, [469](#)
- unpack_loc_EventRegister_t, [296](#)
 - Tlvresult, [297](#)
- unpack_loc_PositionRpt_Ind
 - loc.h, [469](#)
- unpack_loc_PositionRpt_Ind_t, [297](#)
 - pAltitudeAssumed, [301](#)
 - pAltitudeWrtEllipsoid, [301](#)
 - pAltitudeWrtMeanSeaLevel, [301](#)
 - pFixId, [301](#)
 - pGpsTime, [301](#)
 - pHeading, [301](#)
 - pHeadingUnc, [302](#)
 - pHorConfidence, [302](#)
 - pHorReliability, [302](#)
 - pHorUncCircular, [302](#)
 - pHorUncEllipseOrientAzimuth, [302](#)
 - pHorUncEllipseSemiMajor, [302](#)
 - pHorUncEllipseSemiMinor, [302](#)
 - pLatitude, [302](#)
 - pLeapSeconds, [302](#)
 - pLongitude, [302](#)
 - pMagneticDeviation, [302](#)
 - pPrecisionDilution, [302](#)
 - pSensorDataUsage, [302](#)
 - pSpeedHorizontal, [302](#)
 - pSpeedUnc, [302](#)
 - pSpeedVertical, [302](#)
 - pSvUsedforFix, [302](#)
 - pTechnologyMask, [302](#)
 - pTimeSrc, [302](#)
 - pTimeUnc, [302](#)
 - pTimestampUtc, [302](#)
 - pVertConfidence, [302](#)
 - pVertReliability, [302](#)
 - pVertUnc, [302](#)
 - sessionId, [302](#)
 - sessionStatus, [302](#)
 - Tlvresult, [302](#)
- unpack_loc_SLQSLOCGetBestAvailPos
 - loc.h, [471](#)
- unpack_loc_SLQSLOCGetBestAvailPos_t, [304](#)
 - Tlvresult, [304](#)
- unpack_loc_SetExtPowerConfig_Ind
 - loc.h, [469](#)
- unpack_loc_SetExtPowerConfig_Ind_t, [303](#)
 - status, [303](#)
 - Tlvresult, [303](#)
- unpack_loc_SetExtPowerState
 - loc.h, [470](#)
- unpack_loc_SetExtPowerState_t, [303](#)
 - Tlvresult, [304](#)
- unpack_loc_SetOperationMode
 - loc.h, [470](#)
- unpack_loc_SetOperationMode_t, [304](#)
 - Tlvresult, [304](#)
- unpack_loc_Start
 - loc.h, [471](#)
- unpack_loc_Start_t, [304](#)
 - Tlvresult, [305](#)
- unpack_loc_Stop
 - loc.h, [471](#)
- unpack_loc_Stop_t, [305](#)
 - Tlvresult, [305](#)
- unpack_nas_GetACCOLC
 - nas.h, [488](#)
- unpack_nas_GetANAAAAuthenticationStatus
 - nas.h, [489](#)
- unpack_nas_GetCDMANetworkParameters
 - nas.h, [489](#)
- unpack_nas_GetCDMANetworkParameters_t, [305](#)
 - Application, [306](#)
 - Broadcast, [306](#)
 - CustomSCP, [306](#)
 - ForceRev0, [306](#)
 - Protocol, [306](#)
 - RegForeignNID, [306](#)
 - RegForeignSID, [306](#)
 - RegHomeSID, [306](#)
 - Roaming, [306](#)
 - SCI, [306](#)
 - SCM, [306](#)
- unpack_nas_GetHomeNetwork
 - nas.h, [489](#)
- unpack_nas_GetHomeNetwork_t, [306](#)
 - mcc, [307](#)
 - mnc, [307](#)
 - name, [307](#)
 - nid, [307](#)
 - sid, [307](#)
- unpack_nas_GetNetworkPreference
 - nas.h, [490](#)
- unpack_nas_GetNetworkPreference_t, [307](#)
 - ActiveTechPref, [308](#)
 - Duration, [308](#)
 - PersistentTechPref, [308](#)
 - Tlvresult, [308](#)
- unpack_nas_GetRFInfo
 - nas.h, [490](#)
- unpack_nas_GetRFInfo_t, [308](#)
 - instancesSize, [308](#)
 - RFBandInfoElements, [308](#)
- unpack_nas_GetServingNetwork
 - nas.h, [490](#)
- unpack_nas_GetServingNetwork_t, [308](#)
 - CSDomain, [309](#)
 - DataCaps, [309](#)
 - DataCapsLen, [309](#)
 - MCC, [309](#)
 - MNC, [309](#)
 - Name, [309](#)
 - nameSize, [309](#)
 - PSDomain, [309](#)
 - RAN, [309](#)
 - Radiolfaces, [309](#)

- RadiolfacesSize, [309](#)
- RegistrationState, [309](#)
- Roaming, [309](#)
- unpack_nas_GetServingNetworkCapabilities
 - nas.h, [490](#)
- unpack_nas_GetServingNetworkCapabilities_t, [309](#)
 - DataCaps, [310](#)
 - DataCapsLen, [310](#)
- unpack_nas_GetSignalStrengths
 - nas.h, [491](#)
- unpack_nas_GetSignalStrengths_t, [310](#)
 - len, [310](#)
 - radio, [310](#)
 - rsi, [310](#)
- unpack_nas_PerformNetworkScan
 - nas.h, [491](#)
- unpack_nas_PerformNetworkScan_t, [310](#)
 - p3GppNetworkInfoInstances, [311](#)
 - p3GppNetworkInstanceSize, [311](#)
 - pPCInstance, [311](#)
 - pPCInstanceSize, [311](#)
 - pRATInstance, [311](#)
 - pRATInstanceSize, [311](#)
 - pScanResult, [311](#)
- unpack_nas_SLQSGetNetworkTime
 - nas.h, [493](#)
- unpack_nas_SLQSGetNetworkTime_t, [315](#)
 - p3GPP2TimeInfo, [315](#)
 - p3GPPTIMEInfo, [315](#)
- unpack_nas_SLQSGetPLMNName
 - nas.h, [494](#)
- unpack_nas_SLQSGetPLMNName_t, [315](#)
 - longName, [316](#)
 - longNameCI, [316](#)
 - longNameEn, [316](#)
 - longNameLen, [316](#)
 - longNameSB, [316](#)
 - shortName, [316](#)
 - shortNameCI, [316](#)
 - shortNameEn, [316](#)
 - shortNameLen, [316](#)
 - shortNameSB, [316](#)
 - spn, [316](#)
 - spnEncoding, [316](#)
 - spnLength, [316](#)
- unpack_nas_SLQSGetServingSystem
 - nas.h, [494](#)
- unpack_nas_SLQSGetServingSystem_t, [316](#)
 - BaseStationID, [317](#)
 - BaseStationLatitude, [318](#)
 - BaseStationLongitude, [318](#)
 - CDMASystemInfoExt, [318](#)
 - CallBarStatus, [318](#)
 - CellID, [318](#)
 - ConcSvcInfo, [318](#)
 - CurrentPLMN, [318](#)
 - DTMInd, [318](#)
 - DataSrvCapabilities, [318](#)
 - DefaultRoamInd, [318](#)
 - DetailedSvcInfo, [318](#)
 - Gpp2TimeZone, [318](#)
 - GppNetworkDSTAdjustment, [318](#)
 - GppTimeZone, [318](#)
 - HdrPersonality, [318](#)
 - Lac, [318](#)
 - NetworkID, [318](#)
 - PRLInd, [318](#)
 - RoamIndicatorVal, [318](#)
 - RoamingIndicatorList, [318](#)
 - ServingSystem, [318](#)
 - SystemID, [318](#)
 - TrackAreaCode, [318](#)
- unpack_nas_SLQSGetSignalStrength
 - nas.h, [494](#)
- unpack_nas_SLQSGetSignalStrength_t, [318](#)
 - ecioList, [319](#)
 - ecioListLen, [319](#)
 - errorRateList, [319](#)
 - errorRateListLen, [319](#)
 - lo, [319](#)
 - lrsrp, [319](#)
 - lrsnr, [319](#)
 - rsrqInfo, [319](#)
 - rxSignalStrengthList, [319](#)
 - rxSignalStrengthListLen, [319](#)
 - signalStrengthReqMask, [319](#)
 - sinr, [319](#)
- unpack_nas_SLQSGetSysInfo
 - nas.h, [495](#)
- unpack_nas_SLQSGetSysInfo_t, [320](#)
 - pAddCDMASysInfo, [322](#)
 - pAddGSMSysInfo, [322](#)
 - pAddHDRSysInfo, [322](#)
 - pAddLTESysInfo, [322](#)
 - pCDMASrvStatusInfo, [322](#)
 - pCDMASysInfo, [322](#)
 - pGSMCallBarringSysInfo, [322](#)
 - pGSMCipherDomainSysInfo, [322](#)
 - pGSMSrvStatusInfo, [322](#)
 - pGSMSysInfo, [322](#)
 - pHDSrvStatusInfo, [322](#)
 - pHDRSysInfo, [322](#)
 - pLTESrvStatusInfo, [322](#)
 - pLTESysInfo, [322](#)
 - pLTEVoiceSupportSysInfo, [322](#)
 - pWCDMASysInfo, [322](#)
- unpack_nas_SLQSGetSysSelectionPref
 - nas.h, [495](#)
- unpack_nas_SLQSGetSysSelectionPref_t, [322](#)
 - pBandPref, [326](#)
 - pEmerMode, [326](#)
 - pGWAqOrderPref, [326](#)
 - pLTEBandPref, [326](#)
 - pModePref, [326](#)
 - pNetSelPref, [326](#)
 - pPRLPref, [326](#)

- pRoamPref, [326](#)
- pSrvDomainPref, [326](#)
- unpack_nas_SLQSIInitiateNetworkRegistration
 - nas.h, [495](#)
- unpack_nas_SLQSNasConfigSigInfo2
 - nas.h, [495](#)
- unpack_nas_SLQSNasGetCellLocationInfo
 - nas.h, [496](#)
- unpack_nas_SLQSNasGetCellLocationInfo_t, [326](#)
 - pCDMAInfo, [327](#)
 - pLTEInfoInterfreq, [327](#)
 - pLTEInfoIntrafreq, [327](#)
 - pUMTSInfo, [328](#)
- unpack_nas_SLQSNasGetSigInfo
 - nas.h, [496](#)
- unpack_nas_SLQSNasGetSigInfo_t, [328](#)
 - CDMASSInfo, [328](#)
 - GSMSSInfo, [328](#)
 - HDRSSInfo, [328](#)
 - LTSSInfo, [328](#)
- unpack_nas_SLQSNasIndicationRegisterExt
 - nas.h, [496](#)
- unpack_nas_SLQSNasNetworkTimeCallBack_ind
 - nas.h, [497](#)
- unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, [328](#)
 - pDayltSavAdj, [329](#)
 - pRadioInterface, [329](#)
 - pTimeZone, [329](#)
 - universalTime, [329](#)
- unpack_nas_SLQSNasSigInfoCallback_ind
 - nas.h, [497](#)
- unpack_nas_SLQSNasSigInfoCallback_ind_t, [329](#)
 - pRscp, [330](#)
- unpack_nas_SLQSNasSwiModemStatus
 - nas.h, [497](#)
- unpack_nas_SLQSNasSwiModemStatus_t, [330](#)
 - commonInfo, [330](#)
 - pLTEInfo, [330](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback
 - nas.h, [498](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind
 - nas.h, [498](#)
- unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, [331](#)
 - Info, [331](#)
 - Tlvresult, [331](#)
- unpack_nas_SLQSNasSysInfoCallback_ind
 - nas.h, [498](#)
- unpack_nas_SLQSSetBandPreference
 - nas.h, [499](#)
- unpack_nas_SLQSSetSignalStrengthsCallback
 - nas.h, [499](#)
- unpack_nas_SLQSSetSysSelectionPref
 - nas.h, [499](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
 - nas.h, [499](#)
- unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, [331](#)
 - Info, [331](#)
 - Tlvresult, [331](#)
- unpack_nas_SLQSSwiGetLteCQI
 - nas.h, [500](#)
- unpack_nas_SLQSSwiGetLteCQI_t, [331](#)
 - ValidityCW0, [332](#)
 - ValidityCW1, [332](#)
- unpack_nas_SLQSSwiGetLteSccRxInfo
 - nas.h, [500](#)
- unpack_nas_SLQSSwiGetLteSccRxInfo_t, [332](#)
 - pSccRxInfo, [333](#)
- unpack_nas_SLQSSysInfoCallback_ind_t, [333](#)
 - pGSMSysInfo, [335](#)
 - pHDRSysInfo, [335](#)
 - pLTESysInfo, [335](#)
 - pSysInfoNoChange, [335](#)
- unpack_nas_SetACCOLC
 - nas.h, [491](#)
- unpack_nas_SetDataCapabilitiesCallback_ind
 - nas.h, [492](#)
- unpack_nas_SetDataCapabilitiesCallback_ind_t, [311](#)
 - dataCaps, [311](#)
 - dataCapsSize, [311](#)
- unpack_nas_SetEventReportInd
 - nas.h, [492](#)
- unpack_nas_SetEventReportInd_t, [311](#)
 - RFTlv, [312](#)
 - RRTlv, [312](#)
 - SLQSSSTlv, [312](#)
 - SSTlv, [312](#)
- unpack_nas_SetLURejectCallback
 - nas.h, [492](#)
- unpack_nas_SetNasLTECphyCaIndCallback_ind
 - nas.h, [492](#)
- unpack_nas_SetNasLTECphyCaIndCallback_ind_t, [312](#)
 - roaming, [314](#)
- unpack_nas_SetNetworkPreference
 - nas.h, [493](#)
- unpack_nas_SetNetworkPreference_t, [313](#)
 - Tlvresult, [313](#)
- unpack_nas_SetRFInfoCallback
 - nas.h, [493](#)
- unpack_nas_SetRoamingIndicatorCallback_ind
 - nas.h, [493](#)
- unpack_nas_SetRoamingIndicatorCallback_ind_t, [314](#)
 - roaming, [314](#)
- unpack_nas_SetServingSystemCallback_ind
 - nas.h, [493](#)
- unpack_nas_SetServingSystemCallback_ind_t, [314](#)
 - SSInfo, [314](#)
 - Tlvresult, [314](#)
- unpack_nas_SlqsGetLTECphyCAInfo
 - nas.h, [493](#)
- unpack_nas_SlqsGetLTECphyCAInfo_t, [314](#)
 - LTECphyCAInfo, [315](#)
 - Tlvresult, [315](#)
- unpack_omaDmConfigTlv_t, [335](#)

- alertmsg, 336
- alertmsglength, 336
- state, 336
- userInputReq, 336
- userInputTimeout, 336
- unpack_omaDmFotaTlv_t, 336
 - description, 338
 - descriptionlength, 338
 - fwloadsize, 338
 - fwloadComplete, 338
 - namelength, 338
 - package_name, 338
 - sessionType, 338
 - severity, 338
 - state, 338
 - updateCompleteStatus, 338
 - userInputReq, 338
 - userInputTimeout, 338
 - version, 338
 - versionlength, 338
- unpack_omaDmNotificationsTlv_t, 338
 - notification, 339
 - sessionStatus, 339
- unpack_qmi_t, 339
 - msgid, 339
 - type, 339
 - xid, 339
- unpack_qos_IPv4Addr_t, 340
 - addr, 340
 - subnetMask, 340
- unpack_qos_IPv6Addr_t, 340
 - addr, 340
 - prefixLen, 340
- unpack_qos_IPv6TrafCls_t, 341
 - mask, 341
 - val, 341
- unpack_qos_Port_t, 341
 - port, 342
 - range, 342
- unpack_qos_QosFlowInfo_t, 342
 - BearerID, 343
 - is_RxQFlowGranted_Available, 343
 - is_TxQFlowGranted_Available, 343
 - NumRxFilters, 343
 - NumTxFilters, 343
 - QFlowState, 343
 - RxQFilter, 343
 - RxQFlowGranted, 343
 - TxQFilter, 343
 - TxQFlowGranted, 343
- unpack_qos_QosFlowInfoState_t, 343
 - id, 344
 - isNewFlow, 344
 - state, 344
- unpack_qos_SLQSQosGetNetworkStatus
 - qos.h, 549
- unpack_qos_SLQSQosGetNetworkStatus_t, 344
 - NWQoSStatus, 344
- unpack_qos_SLQSQosSwiReadApnExtraParams
 - qos.h, 550
- unpack_qos_SLQSQosSwiReadApnExtraParams_t, 345
 - ambr_dl, 345
 - ambr_dl_ext, 345
 - ambr_dl_ext2, 346
 - ambr_ul, 346
 - ambr_ul_ext, 346
 - ambr_ul_ext2, 346
 - apnId, 346
- unpack_qos_SLQSQosSwiReadDataStats
 - qos.h, 550
- unpack_qos_SLQSQosSwiReadDataStats_t, 346
 - apnId, 347
 - numQosFlow, 347
 - qosFlow, 347
 - total_rx_bytes, 347
 - total_rx_pkt, 347
 - total_tx_bytes, 347
 - total_tx_pkt, 347
- unpack_qos_SLQSSetQosEventCallback
 - qos.h, 551
- unpack_qos_SLQSSetQosEventCallback_ind
 - qos.h, 551
- unpack_qos_SLQSSetQosEventCallback_ind_t, 347
 - NumFlows, 347
 - QosFlowInfo, 348
- unpack_qos_SLQSSetQosNWStatusCallback_ind
 - qos.h, 551
- unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 348
 - status, 348
- unpack_qos_SLQSSetQosPriEventCallback_ind
 - qos.h, 552
- unpack_qos_SLQSSetQosPriEventCallback_ind_t, 348
 - event, 348
- unpack_qos_SLQSSetQosStatusCallback_ind
 - qos.h, 553
- unpack_qos_SLQSSetQosStatusCallback_ind_t, 348
 - event, 350
 - id, 350
 - reason, 350
 - status, 350
- unpack_qos_Tos_t, 358
 - mask, 358
 - val, 358
- unpack_qos_dataRate_t, 339
 - dataRateMax, 339
 - guaranteedRate, 339
- unpack_qos_pktErrRate_t, 341
 - exponent, 341
 - multiplier, 341
- unpack_qos_swiQosFilter_t, 350
 - EspSpi, 352
 - IPv4DstAddr, 352
 - IPv4SrcAddr, 352
 - IPv4Tos, 352

- IPv6DstAddr, [352](#)
- IPv6Label, [352](#)
- IPv6SrcAddr, [352](#)
- IPv6TrafCls, [352](#)
- Id, [352](#)
- index, [352](#)
- is_EspSpi_Available, [352](#)
- is_IPv4DstAddr_Available, [352](#)
- is_IPv4SrcAddr_Available, [352](#)
- is_IPv4Tos_Available, [352](#)
- is_IPv6DstAddr_Available, [352](#)
- is_IPv6Label_Available, [352](#)
- is_IPv6SrcAddr_Available, [352](#)
- is_IPv6TrafCls_Available, [352](#)
- is_Id_Available, [352](#)
- is_NxtHdrProto_Available, [353](#)
- is_Precedence_Available, [353](#)
- is_TCPDstPort_Available, [353](#)
- is_TCPSrcPort_Available, [353](#)
- is_TranDstPort_Available, [353](#)
- is_TranSrcPort_Available, [353](#)
- is_UDPDstPort_Available, [353](#)
- is_UDPSrcPort_Available, [353](#)
- NxtHdrProto, [353](#)
- Precedence, [353](#)
- TCPDstPort, [353](#)
- TCPSrcPort, [353](#)
- TranDstPort, [353](#)
- TranSrcPort, [353](#)
- UDPDstPort, [353](#)
- UDPSrcPort, [353](#)
- version, [353](#)
- unpack_qos_swqosFlow_t, [353](#)
 - DataRate, [356](#)
 - index, [356](#)
 - is_DataRate_Available, [356](#)
 - is_Jitter_Available, [356](#)
 - is_Latency_Available, [356](#)
 - is_LteQci_Available, [356](#)
 - is_MaxAllowedPktSz_Available, [356](#)
 - is_MinPktErrRate_Available, [356](#)
 - is_ProfileId3GPP2_Available, [356](#)
 - is-TokenBucket_Available, [356](#)
 - is_TrafficClass_Available, [356](#)
 - is_val_3GPP2Pri_Available, [356](#)
 - is_val_3GPPImCn_Available, [356](#)
 - is_val_3GPPSigInd_Available, [357](#)
 - Jitter, [357](#)
 - Latency, [357](#)
 - LteQci, [357](#)
 - MaxAllowedPktSz, [357](#)
 - MinPktErrRate, [357](#)
 - PktErrRate, [357](#)
 - ProfileId3GPP2, [357](#)
 - TokenBucket, [357](#)
 - TrafficClass, [357](#)
 - val_3GPP2Pri, [357](#)
 - val_3GPPImCn, [357](#)
 - val_3GPPResResidualBER, [357](#)
 - val_3GPPSigInd, [357](#)
 - val_3GPPTraHdlPri, [357](#)
- unpack_qos_tokenBucket_t, [357](#)
 - bucketSz, [358](#)
 - peakRate, [358](#)
 - tokenRate, [358](#)
- unpack_result_code_only
 - common.h, [415](#)
- unpack_sms_SLQSDelSMS
 - sms.h, [560](#)
- unpack_sms_SLQSDelSMS_t, [361](#)
- unpack_sms_SLQSGetSMS
 - sms.h, [561](#)
- unpack_sms_SLQSGetSMS_t, [361](#)
 - message, [361](#)
 - messageFormat, [361](#)
 - messageSize, [362](#)
 - messageTag, [362](#)
- unpack_sms_SLQSGetSMSList
 - sms.h, [561](#)
- unpack_sms_SLQSGetSMSList_t, [362](#)
 - messageList, [362](#)
 - messageListSize, [362](#)
- unpack_sms_SLQSModifySMSStatus
 - sms.h, [561](#)
- unpack_sms_SLQSModifySMSStatus_t, [362](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind
 - sms.h, [562](#)
- unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, [362](#)
 - messageMode, [363](#)
 - storageType, [363](#)
- unpack_sms_SendSMS
 - sms.h, [559](#)
- unpack_sms_SendSMS_t, [359](#)
 - messageFailureCode, [359](#)
 - messageID, [359](#)
- unpack_sms_SetNewSMSCallback
 - sms.h, [560](#)
- unpack_sms_SetNewSMSCallback_ind
 - sms.h, [560](#)
- unpack_sms_SetNewSMSCallback_ind_t, [360](#)
 - ETWSTlv, [360](#)
 - IMSTlv, [360](#)
 - MMTlv, [360](#)
 - NewMMTlv, [360](#)
 - SMSCTlv, [361](#)
 - TRMessageTlv, [361](#)
- unpack_sms_SetNewSMSCallback_t, [361](#)
- unpack_swilc_SwiLocGetAutoStart
 - swilc.h, [564](#)
- unpack_swilc_SwiLocGetAutoStart_t, [363](#)
 - fix_rate, [364](#)
 - fix_rate_reported, [364](#)
 - fix_type, [364](#)
 - fix_type_reported, [364](#)

- function, 364
- function_reported, 364
- max_dist, 364
- max_dist_reported, 364
- max_time, 364
- max_time_reported, 364
- unpack_swiloc_SwiLocSetAutoStart
 - swiloc.h, 565
- unpack_swioima_SLQSOMADMAAlertCallback
 - swioima.h, 570
- unpack_swioima_SLQSOMADMAAlertCallback_ind
 - swioima.h, 571
- unpack_swioima_SLQSOMADMAAlertCallback_ind_t, 364
 - eventType, 365
- unpack_swioima_SLQSOMADMCancelSession
 - swioima.h, 571
- unpack_swioima_SLQSOMADMGetSessionInfo
 - swioima.h, 572
- unpack_swioima_SLQSOMADMGetSessionInfo_t, 365
 - Date, 367
 - DateLength, 367
 - PkgDescLength, 367
 - PkgDescription, 367
 - PkgName, 367
 - PkgNameLength, 367
 - RetryCount, 367
 - SessionState, 367
 - SessionType, 367
 - Severity, 367
 - Source, 368
 - SourceLength, 368
 - Status, 368
 - Time, 368
 - TimeLength, 368
 - UpdateCompleteStatus, 368
- unpack_swioima_SLQSOMADMGetSettings
 - swioima.h, 572
- unpack_swioima_SLQSOMADMGetSettings_t, 368
 - Autosdm, 369
 - FOTAdownload, 369
 - FwAutoCheck, 369
- unpack_swioima_SLQSOMADMSendSelection
 - swioima.h, 573
- unpack_swioima_SLQSOMADMSetSettings
 - swioima.h, 573
- unpack_swioima_SLQSOMADMStartSession
 - swioima.h, 573
- unpack_swioima_SLQSOMADMStartSession_t, 369
 - FwAvailability, 370
- unpack_uim_ChangePin
 - uim.h, 580
- unpack_uim_ChangePin_t, 370
 - pEncryptedPIN1, 370
 - pIndicationToken, 370
 - pRemainingRetries, 370
 - Tlvresult, 370
- unpack_uim_GetCardStatus
 - uim.h, 581
- unpack_uim_GetCardStatus_t, 371
 - pCardStatus, 371
 - pHotSwapStatus, 371
 - Tlvresult, 371
- unpack_uim_ReadTransparent
 - uim.h, 581
- unpack_uim_ReadTransparent_t, 371
 - pCardResult, 372
 - pEncryptedData, 372
 - pIndicationToken, 372
 - pReadResult, 372
 - Tlvresult, 372
- unpack_uim_SLQSUIEventRegister
 - uim.h, 582
- unpack_uim_SLQSUIEventRegister_t, 373
 - eventMask, 374
- unpack_uim_SLQSUIGetSlotsStatus
 - uim.h, 582
- unpack_uim_SLQSUIGetSlotsStatus_t, 374
 - pNumberOfPhySlot, 374
 - pUimSlotsStatus, 374
- unpack_uim_SLQSUIPowerDown
 - uim.h, 583
- unpack_uim_SLQSUIPowerUp
 - uim.h, 583
- unpack_uim_SLQSUISetStatusChangeCallBack_ind
 - uim.h, 583
- unpack_uim_SLQSUISetStatusChangeCallBack_ind_t, 374
- unpack_uim_SLQSUISwitchSlot
 - uim.h, 584
- unpack_uim_SetPinProtection
 - uim.h, 581
- unpack_uim_SetPinProtection_t, 372
 - pEncryptedPIN1, 373
 - pIndicationToken, 373
 - pRemainingRetries, 373
 - Tlvresult, 373
- unpack_uim_SetUimSlotStatusChangeCallback_ind
 - uim.h, 582
- unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 373
 - bNumberOfPhySlots, 373
 - slotsstatusChange, 373
- unpack_uim_UnblockPin
 - uim.h, 584
- unpack_uim_UnblockPin_t, 375
 - pEncryptedPIN1, 375
 - pIndicationToken, 375
 - pRemainingRetries, 375
 - Tlvresult, 375
- unpack_uim_VerifyPin
 - uim.h, 584
- unpack_uim_VerifyPin_t, 375
 - pEncryptedPIN1, 376
 - pIndicationToken, 376
 - pRemainingRetries, 376

- Tlvresult, [376](#)
- unpack_wds_GetByteTotals
 - wds.h, [604](#)
- unpack_wds_GetByteTotals_t, [376](#)
 - pRXTotalBytes, [376](#)
 - pTXTotalBytes, [376](#)
- unpack_wds_GetConnectionRate
 - wds.h, [604](#)
- unpack_wds_GetConnectionRate_t, [377](#)
 - currentChannelRXRate, [377](#)
 - currentChannelTXRate, [377](#)
 - maxChannelRXRate, [377](#)
 - maxChannelTXRate, [377](#)
- unpack_wds_GetDefaultProfile
 - wds.h, [605](#)
- unpack_wds_GetDefaultProfile_t, [377](#)
 - apnname, [378](#)
 - apnsize, [378](#)
 - auth, [378](#)
 - ipaddr, [378](#)
 - ipaddrv6, [378](#)
 - name, [378](#)
 - namesize, [378](#)
 - pdptype, [378](#)
 - pridns, [378](#)
 - pridnsv6, [378](#)
 - secdns, [378](#)
 - secdnsv6, [378](#)
 - username, [378](#)
 - usersize, [378](#)
- unpack_wds_GetDefaultProfileNum
 - wds.h, [605](#)
- unpack_wds_GetDefaultProfileNum_t, [378](#)
 - index, [379](#)
- unpack_wds_GetDormancyState
 - wds.h, [605](#)
- unpack_wds_GetDormancyState_t, [379](#)
 - dormancyState, [379](#)
- unpack_wds_GetLastMobileIPError
 - wds.h, [606](#)
- unpack_wds_GetLastMobileIPError_t, [379](#)
 - error, [379](#)
- unpack_wds_GetMobileIP
 - wds.h, [606](#)
- unpack_wds_GetMobileIP_t, [379](#)
 - mipMode, [380](#)
- unpack_wds_GetMobileIPProfile
 - wds.h, [606](#)
- unpack_wds_GetMobileIPProfile_t, [380](#)
 - AAASPI, [380](#)
 - AAASState, [380](#)
 - address, [380](#)
 - enabled, [380](#)
 - HASPI, [380](#)
 - HASState, [380](#)
 - NAI, [380](#)
 - naiSize, [380](#)
 - primaryHA, [380](#)
 - revTunneling, [381](#)
 - secondaryHA, [381](#)
- unpack_wds_GetPacketStatistics
 - wds.h, [607](#)
- unpack_wds_GetPacketStatistics_t, [381](#)
 - pRXDroppedCount, [382](#)
 - pRXOKBytesLastCall, [382](#)
 - pRXOkBytesCount, [382](#)
 - pRXPacketErrors, [382](#)
 - pRXPacketOverflows, [382](#)
 - pRXPacketSuccesses, [382](#)
 - pTXDroppedCount, [382](#)
 - pTXOKBytesLastCall, [382](#)
 - pTXOkBytesCount, [382](#)
 - pTXPacketErrors, [382](#)
 - pTXPacketOverflows, [382](#)
 - pTXPacketSuccesses, [382](#)
- unpack_wds_GetPacketStatus
 - wds.h, [607](#)
- unpack_wds_GetPacketStatus_t, [382](#)
 - rXDroppedCount, [383](#)
 - rXOKBytesLastCall, [383](#)
 - rXOkBytesCount, [383](#)
 - rXPacketErrors, [383](#)
 - rXPacketOverflows, [383](#)
 - rXPacketSuccesses, [383](#)
 - tXDroppedCount, [383](#)
 - tXOKBytesLastCall, [383](#)
 - tXOkBytesCount, [383](#)
 - tXPacketErrors, [383](#)
 - tXPacketOverflows, [383](#)
 - tXPacketSuccesses, [383](#)
- unpack_wds_GetSessionDuration
 - wds.h, [607](#)
- unpack_wds_GetSessionDuration_t, [383](#)
 - callDuration, [384](#)
- unpack_wds_GetSessionState
 - wds.h, [608](#)
- unpack_wds_GetSessionState_t, [384](#)
 - connectionStatus, [384](#)
- unpack_wds_RMSetTransferStatistics
 - wds.h, [608](#)
- unpack_wds_RMSetTransferStatistics_t, [384](#)
- unpack_wds_SLQSCreateProfile
 - wds.h, [609](#)
- unpack_wds_SLQSCreateProfile_t, [384](#)
 - pCreateProfileOut, [384](#)
 - pProfileID, [384](#)
 - Tlvresult, [385](#)
- unpack_wds_SLQSDeleteProfile
 - wds.h, [610](#)
- unpack_wds_SLQSDeleteProfile_t, [385](#)
 - extendedErrorCode, [385](#)
- unpack_wds_SLQSGet3GPPConfigItem
 - wds.h, [610](#)
- unpack_wds_SLQSGet3GPPConfigItem_t, [385](#)
 - _3gppRelease, [386](#)
 - defaultPDNEnabled, [386](#)

- LTEAttachProfile, [386](#)
 - profileList, [386](#)
- unpack_wds_SLQSGetCurrDataSystemStat
 - wds.h, [610](#)
- unpack_wds_SLQSGetCurrDataSystemStat_t, [386](#)
 - currNetworkInfo, [387](#)
 - networkInfoLen, [387](#)
 - prefNetwork, [387](#)
- unpack_wds_SLQSGetCurrentChannelRate
 - wds.h, [611](#)
- unpack_wds_SLQSGetCurrentChannelRate_t, [387](#)
 - max_channel_rx_rate, [388](#)
 - max_channel_tx_rate, [388](#)
- unpack_wds_SLQSGetDUNCallInfo
 - wds.h, [611](#)
- unpack_wds_SLQSGetDUNCallInfo_t, [388](#)
 - callEndReason, [389](#)
 - channelRate, [389](#)
 - connectionStatus, [389](#)
 - dataBearerTech, [389](#)
 - dormancyStatus, [389](#)
 - lastCallDataBearerTech, [389](#)
 - mdmCallDurationActive, [389](#)
 - rxOKBytesCount, [389](#)
 - txOKBytesCount, [389](#)
- unpack_wds_SLQSGetDataBearerTechnology
 - wds.h, [611](#)
- unpack_wds_SLQSGetDataBearerTechnology_t, [388](#)
 - curDataBearerTechnology, [388](#)
 - dataBearerMask, [388](#)
 - lastCallDataBearerTechnology, [388](#)
- unpack_wds_SLQSGetProfileSettings
 - wds.h, [612](#)
- unpack_wds_SLQSGetProfileSettings_t, [389](#)
 - pProfileSettings, [389](#)
 - ProfileType, [390](#)
 - Tlvresult, [390](#)
- unpack_wds_SLQSGetRuntimeSettings
 - wds.h, [612](#)
- unpack_wds_SLQSGetRuntimeSettings_t, [390](#)
 - APNName, [391](#)
 - Authentication, [391](#)
 - DomainList, [391](#)
 - GPRSGrantedQoS, [391](#)
 - GWAddressV4, [391](#)
 - IMCNflag, [391](#)
 - IPFamilyPreference, [391](#)
 - IPV6AddrInfo, [391](#)
 - IPV6GWAddrInfo, [391](#)
 - IPv4, [391](#)
 - Mtu, [391](#)
 - PDPTType, [391](#)
 - PrimaryDNSV4, [391](#)
 - PrimaryDNSV6, [391](#)
 - ProfileID, [391](#)
 - ProfileName, [391](#)
 - SecondaryDNSV4, [391](#)
 - SecondaryDNSV6, [392](#)
 - ServerAddrList, [392](#)
 - SubnetMaskV4, [392](#)
 - Technology, [392](#)
 - UMTSGrantedQoS, [392](#)
 - Username, [392](#)
- unpack_wds_SLQSMModifyProfile
 - wds.h, [612](#)
- unpack_wds_SLQSMModifyProfile_t, [392](#)
 - pExtErrorCode, [392](#)
- unpack_wds_SLQSSetDHCPv4ClientConfig
 - wds.h, [614](#)
- unpack_wds_SLQSSetDHCPv4ClientConfig_t, [395](#)
 - pHwConfig, [395](#)
- unpack_wds_SLQSSetLoopback
 - wds.h, [615](#)
- unpack_wds_SLQSSetLoopback_t, [395](#)
 - ByteLoopbackMode, [396](#)
 - ByteLoopbackMultiplier, [396](#)
- unpack_wds_SLQSSetLoopback
 - wds.h, [615](#)
- unpack_wds_SLQSSet3GPPConfigItem
 - wds.h, [613](#)
- unpack_wds_SLQSSetIPFamilyPreference
 - wds.h, [613](#)
- unpack_wds_SLQSSetIPFamilyPreference_t, [392](#)
 - Tlvresult, [392](#)
- unpack_wds_SLQSSetPacketSrvStatusCallback
 - wds.h, [613](#)
- unpack_wds_SLQSSetPacketSrvStatusCallback_t, [392](#)
 - bearerID, [393](#)
 - conn_status, [393](#)
 - ipFamily, [393](#)
 - reconfigReqd, [393](#)
 - sessionEndReason, [393](#)
 - techName, [393](#)
 - verboseSessnEndReason, [393](#)
 - verboseSessnEndReasonType, [393](#)
- unpack_wds_SLQSSetWdsEventCallback
 - wds.h, [614](#)
- unpack_wds_SLQSSetWdsEventCallback_ind
 - wds.h, [614](#)
- unpack_wds_SLQSSetWdsEventCallback_ind_t, [393](#)
 - currDBTechAvail, [394](#)
 - currNWInfo, [394](#)
 - dBTechAvail, [394](#)
 - dBTechnology, [394](#)
 - dataSysStatAvail, [394](#)
 - dormancyStatAvail, [394](#)
 - dormancyStatus, [394](#)
 - mipStatus, [394](#)
 - mipstatAvail, [394](#)
 - netInfoLen, [395](#)
 - prefNetwork, [395](#)
 - ratMask, [395](#)
 - rx_bytes, [395](#)
 - rx_pkts, [395](#)
 - soMask, [395](#)
 - tx_bytes, [395](#)

- tx_pkts, 395
- xferStatAvail, 395
- unpack_wds_SLQSSStartDataSession
 - wds.h, 615
- unpack_wds_SLQSSStartDataSession_t, 396
 - pFailureReason, 396
 - pVerboseFailReasonType, 396
 - pVerboseFailureReason, 396
 - psid, 396
- unpack_wds_SLQSSStopDataSession
 - wds.h, 616
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings
 - wds.h, 616
- unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 397
 - apnName, 397
 - bearerId, 397
 - contextId, 398
 - ipv4Address, 398
 - ipv4GWAddress, 398
 - ipv6Address, 398
 - ipv6GWAddress, 398
- unpack_wds_SetDefaultProfile
 - wds.h, 608
- unpack_wds_SetDefaultProfileNum
 - wds.h, 609
- unpack_wds_SetMobileIPProfile
 - wds.h, 609
- unpack_wds_SetMobileIPProfile_t, 384
- UnpackQmiProfileInfo
 - wds.h, 590
- unpackWdsProfileParam, 398
 - SlqsProfile3GPP, 398
 - SlqsProfile3GPP2, 398
- UpdateCompleteStatus
 - unpack_swima_SLQSOMADMGetSessionInfo_t, 368
- updateCompleteStatus
 - unpack_omaDmFotaTlv_t, 338
- upinRetries
 - slotInf, 241
 - uim_slotInfo, 260
- upinState
 - slotInf, 241
 - uim_slotInfo, 260
- upukRetries
 - slotInf, 241
 - uim_slotInfo, 260
- usageMask
 - loc_sensorDataUsage, 76
- userInputReq
 - unpack_omaDmConfigTlv_t, 336
 - unpack_omaDmFotaTlv_t, 338
- userInputTimeout
 - unpack_omaDmConfigTlv_t, 336
 - unpack_omaDmFotaTlv_t, 338
- Username
 - unpack_wds_SLQSGetRuntimeSettings_t, 392
- username
 - unpack_wds_GetDefaultProfile_t, 378
- usersize
 - unpack_wds_GetDefaultProfile_t, 378
- VDOP
 - loc_precisionDilution, 75
- val
 - unpack_qos_IPv6TrafCls_t, 341
 - unpack_qos_Tos_t, 358
- val_3GPP2Pri
 - unpack_qos_swiQosFlow_t, 357
- val_3GPPImCn
 - unpack_qos_swiQosFlow_t, 357
- val_3GPPResResidualBER
 - unpack_qos_swiQosFlow_t, 357
- val_3GPPSigInd
 - unpack_qos_swiQosFlow_t, 357
- val_3GPPTraHdlPri
 - unpack_qos_swiQosFlow_t, 357
- ValidityCW0
 - unpack_nas_SLQSSwiGetLteCQI_t, 332
- ValidityCW1
 - unpack_nas_SLQSSwiGetLteCQI_t, 332
- value_length
 - DMScustSettingInfo, 31
 - pack_dms_SetCustFeaturesV2_t, 176
- verboseSessnEndReason
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 393
- verboseSessnEndReasonType
 - unpack_wds_SLQSSetPacketSrvStatusCallback_t, 393
- verifyLeft
 - uim_remainingRetries, 257
- verifyPIN
 - pack_uim_VerifyPin_t, 221
- version
 - unpack_omaDmFotaTlv_t, 338
 - unpack_qos_swiQosFilter_t, 353
- versionlength
 - unpack_omaDmFotaTlv_t, 338
- VoiceNumber
 - unpack_dms_GetVoiceNumber_t, 275
- voiceNumberSize
 - unpack_dms_GetVoiceNumber_t, 275
- WCDMACellInfo
 - nas_lteWcdmaCellInfo, 125
- WCDMAECIOThreshListLen
 - nas_WCDMAECIOThresh, 158
- WCDMARSSIThreshListLen
 - nas_WCDMARSSIThresh, 160
- WCDMASSInfo
 - unpack_nas_SLQSNasGetSigInfo_t, 328
- WORD
 - SwiDataTypes.h, 563
- wcdmaRRCState
 - nas_WCDMAInfoLTENeighborCell, 159

- wds.h, 585
 - BYT_STAT_STAT_MASK, 589
 - PACK_WDS_IPV4, 589
 - PACK_WDS_IPV6, 589
 - pack_wds_GetByteTotals, 590
 - pack_wds_GetConnectionRate, 590
 - pack_wds_GetDefaultProfile, 590
 - pack_wds_GetDefaultProfileNum, 591
 - pack_wds_GetDormancyState, 591
 - pack_wds_GetLastMobileIPError, 592
 - pack_wds_GetMobileIP, 592
 - pack_wds_GetMobileIPProfile, 592
 - pack_wds_GetPacketStatistics, 593
 - pack_wds_GetPacketStatus, 593
 - pack_wds_GetSessionDuration, 594
 - pack_wds_GetSessionState, 594
 - pack_wds_RMSetTransferStatistics, 595
 - pack_wds_SLQSCreateProfile, 596
 - pack_wds_SLQSDeleteProfile, 596
 - pack_wds_SLQSGet3GPPConfigItem, 597
 - pack_wds_SLQSGetCurrDataSystemStat, 597
 - pack_wds_SLQSGetCurrentChannelRate, 598
 - pack_wds_SLQSGetDUNCallInfo, 599
 - pack_wds_SLQSGetDataBearerTechnology, 599
 - pack_wds_SLQSGetProfileSettings, 599
 - pack_wds_SLQSGetRuntimeSettings, 599
 - pack_wds_SLQSModifyProfile, 600
 - pack_wds_SLQSSetDHCPv4ClientConfig, 601
 - pack_wds_SLQSSetLoopback, 602
 - pack_wds_SLQSSetLoopback, 602
 - pack_wds_SLQSSet3GPPConfigItem, 600
 - pack_wds_SLQSSetIPFamilyPreference, 601
 - pack_wds_SLQSSetWdsEventCallback, 601
 - pack_wds_SLQSStartDataSession, 603
 - pack_wds_SLQSStopDataSession, 603
 - pack_wds_SLQSWdsSwiPDPRuntimeSettings, 603
 - pack_wds_SetDefaultProfile, 595
 - pack_wds_SetDefaultProfileNum, 595
 - pack_wds_SetMobileIPProfile, 596
 - unpack_wds_GetByteTotals, 604
 - unpack_wds_GetConnectionRate, 604
 - unpack_wds_GetDefaultProfile, 605
 - unpack_wds_GetDefaultProfileNum, 605
 - unpack_wds_GetDormancyState, 605
 - unpack_wds_GetLastMobileIPError, 606
 - unpack_wds_GetMobileIP, 606
 - unpack_wds_GetMobileIPProfile, 606
 - unpack_wds_GetPacketStatistics, 607
 - unpack_wds_GetPacketStatus, 607
 - unpack_wds_GetSessionDuration, 607
 - unpack_wds_GetSessionState, 608
 - unpack_wds_RMSetTransferStatistics, 608
 - unpack_wds_SLQSCreateProfile, 609
 - unpack_wds_SLQSDeleteProfile, 610
 - unpack_wds_SLQSGet3GPPConfigItem, 610
 - unpack_wds_SLQSGetCurrDataSystemStat, 610
 - unpack_wds_SLQSGetCurrentChannelRate, 611
 - unpack_wds_SLQSGetDUNCallInfo, 611
 - unpack_wds_SLQSGetDataBearerTechnology, 611
 - unpack_wds_SLQSGetProfileSettings, 612
 - unpack_wds_SLQSGetRuntimeSettings, 612
 - unpack_wds_SLQSModifyProfile, 612
 - unpack_wds_SLQSSetDHCPv4ClientConfig, 614
 - unpack_wds_SLQSSetLoopback, 615
 - unpack_wds_SLQSSetLoopback, 615
 - unpack_wds_SLQSSet3GPPConfigItem, 613
 - unpack_wds_SLQSSetIPFamilyPreference, 613
 - unpack_wds_SLQSSetPacketSrvStatusCallback, 613
 - unpack_wds_SLQSSetWdsEventCallback, 614
 - unpack_wds_SLQSSetWdsEventCallback_ind, 614
 - unpack_wds_SLQSStartDataSession, 615
 - unpack_wds_SLQSStopDataSession, 616
 - unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 616
 - unpack_wds_SetDefaultProfile, 608
 - unpack_wds_SetDefaultProfileNum, 609
 - unpack_wds_SetMobileIPProfile, 609
 - UnpackQmiProfileInfo, 590
- wds_Domain, 400
 - domainLen, 401
 - domainName, 401
- wds_DomainNameList, 401
 - domain, 401
 - numInstances, 401
- wds_GPRSQoS, 401
 - delayClass, 402
 - meanThroughputClass, 402
 - peakThroughputClass, 402
 - precedenceClass, 402
 - reliabilityClass, 402
- wds_IPV6AddressInfo, 402
 - IPAddressV6, 402
 - IPv6PrefixLen, 402
- wds_IPV6GWAddressInfo, 402
 - gwAddressV6, 403
 - gwV6PrefixLen, 403
- wds_PCSCFFQDNAddress, 403
 - fqdnAddr, 403
 - fqdnLen, 403
- wds_PCSCFFQDNAddressList, 403
 - numInstances, 404
 - pcsffQDNAddress, 404
- wds_PCSCFIPv4ServerAddressList, 404
 - numInstances, 404
 - pcsflIPv4Addr, 404
- wds_ProfileIdentifier, 404
 - profileIndex, 405
 - profileType, 405
- wds_UMTSMInQoS, 405
 - deliveryErrSDU, 407
 - grntDownlinkBitrate, 407

- grntUplinkBitrate, [407](#)
- maxDownlinkBitrate, [407](#)
- maxSDUSize, [407](#)
- maxUplinkBitrate, [407](#)
- qosDeliveryOrder, [407](#)
- resBerRatio, [407](#)
- sduErrorRatio, [408](#)
- trafficClass, [408](#)
- trafficPriority, [408](#)
- transferDelay, [408](#)
- wds_currNetworkInfo, [399](#)
 - NetworkType, [400](#)
 - RATMask, [400](#)
 - SOMask, [400](#)
- wds_profileInfo, [405](#)
 - SlqsProfile3GPP, [405](#)
 - SlqsProfile3GPP2, [405](#)
- wdsDhcpv4HwConfig, [408](#)
 - chaddr, [408](#)
 - chaddrLen, [408](#)
 - hwType, [408](#)
- wdsDhcpv4Option, [408](#)
 - optCode, [409](#)
 - optVal, [409](#)
 - optValLen, [409](#)
- wdsDhcpv4OptionList, [409](#)
 - numOpt, [409](#)
 - pOptList, [409](#)
- wdsDhcpv4ProfileId, [409](#)
 - profileId, [410](#)
 - profileType, [410](#)
- xferStatAvail
 - unpack_wds_SLQSSetWdsEventCallback_ind_t, [395](#)
- xid
 - pack_loc_SLQSLOCGetBestAvailPos_t, [184](#)
 - pack_qmi_t, [203](#)
 - unpack_qmi_t, [339](#)
- year
 - nas_timeInfo, [152](#)
 - nas_UniversalTime, [157](#)