

LinuxQMISDK-Lite

SLQS04.00.03

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Welcome to the Sierra Wireless Linux QMI SDK API Reference Guide</b>	<b>1</b>
1.1	Important Notice . . . . .	1
1.2	Limitation of Liability . . . . .	1
1.3	Patents . . . . .	2
1.4	Copyright . . . . .	2
1.5	Trademarks . . . . .	2
1.6	Contact Information . . . . .	2
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Namespace Index</b>	<b>5</b>
3.1	Namespace List . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>17</b>
5.1	File List . . . . .	17
<b>6</b>	<b>Module Documentation</b>	<b>19</b>
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

<b>7 Namespace Documentation</b>	<b>21</b>
7.1 Tables Namespace Reference	21
7.1.1 Detailed Description	21
<b>8 Data Structure Documentation</b>	<b>23</b>
8.1 _libSDP_FirmwareInfo_ Struct Reference	23
8.1.1 Detailed Description	23
8.1.2 Field Documentation	24
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 altSrcInfo_t Struct Reference	24
8.2.1 Detailed Description	24
8.2.2 Field Documentation	25
8.2.2.1 coverage	25
8.2.2.2 linkage	25
8.2.2.3 source	25
8.3 appStats Struct Reference	25
8.3.1 Detailed Description	25
8.3.2 Field Documentation	27
8.3.2.1 aidLength	27
8.3.2.2 aidVal	27
8.3.2.3 appState	28
8.3.2.4 appType	28
8.3.2.5 persoFeature	28
8.3.2.6 persoRetries	28
8.3.2.7 persoState	28
8.3.2.8 persoUnblockRetries	28

8.3.2.9	pin1Retries . . . . .	28
8.3.2.10	pin1State . . . . .	28
8.3.2.11	pin2Retries . . . . .	28
8.3.2.12	pin2State . . . . .	28
8.3.2.13	puk1Retries . . . . .	28
8.3.2.14	puk2Retries . . . . .	28
8.3.2.15	univPin . . . . .	28
8.4	CarrierImage_t Struct Reference . . . . .	28
8.4.1	Detailed Description . . . . .	29
8.4.2	Field Documentation . . . . .	29
8.4.2.1	m_FwBuildId . . . . .	29
8.4.2.2	m_FwImageld . . . . .	29
8.4.2.3	m_nCarrierId . . . . .	29
8.4.2.4	m_nFolderId . . . . .	29
8.4.2.5	m_nStorage . . . . .	29
8.4.2.6	m_PriBuildId . . . . .	29
8.4.2.7	m_PriImageld . . . . .	29
8.5	cdmaSSInfo Struct Reference . . . . .	29
8.5.1	Detailed Description . . . . .	30
8.5.2	Field Documentation . . . . .	30
8.5.2.1	ecio . . . . .	30
8.5.2.2	rsi . . . . .	30
8.6	connectionStatus Struct Reference . . . . .	30
8.6.1	Detailed Description . . . . .	30
8.6.2	Field Documentation . . . . .	30
8.6.2.1	MDMCallDuration . . . . .	30
8.6.2.2	MDMConnStatus . . . . .	30
8.7	crashInfoParams Struct Reference . . . . .	30
8.7.1	Detailed Description . . . . .	30
8.7.2	Field Documentation . . . . .	31

8.7.2.1	<a href="#">crashInfo</a>	31
8.7.2.2	<a href="#">crashStatus</a>	31
8.8	<a href="#">crashInformation Struct Reference</a>	31
8.8.1	<a href="#">Detailed Description</a>	31
8.8.2	<a href="#">Field Documentation</a>	32
8.8.2.1	<a href="#">crashData</a>	32
8.8.2.2	<a href="#">crashId</a>	32
8.8.2.3	<a href="#">crashString</a>	32
8.8.2.4	<a href="#">crashStrlen</a>	32
8.8.2.5	<a href="#">gcdumpString</a>	32
8.8.2.6	<a href="#">gcdumpStrlen</a>	32
8.8.2.7	<a href="#">numCrashes</a>	32
8.9	<a href="#">currNetworkInfo Struct Reference</a>	32
8.9.1	<a href="#">Detailed Description</a>	32
8.9.2	<a href="#">Field Documentation</a>	32
8.9.2.1	<a href="#">NetworkType</a>	32
8.9.2.2	<a href="#">RATMask</a>	32
8.9.2.3	<a href="#">SOMask</a>	32
8.10	<a href="#">dms_ActivationStatusTlv Struct Reference</a>	32
8.10.1	<a href="#">Detailed Description</a>	33
8.10.2	<a href="#">Field Documentation</a>	33
8.10.2.1	<a href="#">activationStatus</a>	33
8.10.2.2	<a href="#">TlvPresent</a>	33
8.11	<a href="#">dms_OperatingModeTlv Struct Reference</a>	33
8.11.1	<a href="#">Detailed Description</a>	33
8.11.2	<a href="#">Field Documentation</a>	34
8.11.2.1	<a href="#">operatingMode</a>	34
8.11.2.2	<a href="#">TlvPresent</a>	34
8.12	<a href="#">DMScustSettingInfo Struct Reference</a>	34
8.12.1	<a href="#">Detailed Description</a>	34

8.12.2	Field Documentation	34
8.12.2.1	cust_attr	34
8.12.2.2	cust_id	34
8.12.2.3	cust_value	35
8.12.2.4	id_length	35
8.12.2.5	value_length	35
8.13	DMScustSettingList Struct Reference	35
8.13.1	Detailed Description	35
8.13.2	Field Documentation	35
8.13.2.1	custSetting	35
8.13.2.2	list_type	35
8.13.2.3	num_instances	35
8.14	DMSgetCustomFeatureV2 Struct Reference	35
8.14.1	Detailed Description	36
8.14.2	Field Documentation	36
8.14.2.1	pCustSettingInfo	36
8.14.2.2	pCustSettingList	36
8.14.2.3	pGetCustomInput	36
8.15	DMSgetCustomInput Struct Reference	36
8.15.1	Detailed Description	36
8.15.2	Field Documentation	36
8.15.2.1	cust_id	36
8.15.2.2	list_type	36
8.16	dunchannelRate Struct Reference	36
8.16.1	Detailed Description	37
8.16.2	Field Documentation	37
8.16.2.1	CurrChanRxRate	37
8.16.2.2	CurrChanTxRate	37
8.16.2.3	MaxChanRxRate	37
8.16.2.4	MaxChanTxRate	37

8.17 eriDataparams Struct Reference . . . . .	37
8.17.1 Field Documentation . . . . .	37
8.17.1.1 eriData . . . . .	37
8.17.1.2 eriDataLen . . . . .	37
8.18 eTWSPLMNInfoTlv Struct Reference . . . . .	37
8.18.1 Detailed Description . . . . .	37
8.18.2 Field Documentation . . . . .	38
8.18.2.1 ETWSPLMNInfo . . . . .	38
8.18.2.2 TlvPresent . . . . .	38
8.19 FMSImageElement Struct Reference . . . . .	38
8.19.1 Detailed Description . . . . .	38
8.19.2 Field Documentation . . . . .	38
8.19.2.1 buildId . . . . .	38
8.19.2.2 buildIdLength . . . . .	39
8.19.2.3 imageId . . . . .	39
8.19.2.4 imageType . . . . .	39
8.20 FMSImageIdElement Struct Reference . . . . .	39
8.20.1 Detailed Description . . . . .	39
8.20.2 Field Documentation . . . . .	39
8.20.2.1 buildID . . . . .	39
8.20.2.2 buildIDLength . . . . .	39
8.20.2.3 failureCount . . . . .	39
8.20.2.4 imageID . . . . .	40
8.20.2.5 storageIndex . . . . .	40
8.21 FMSImageIDEntries Struct Reference . . . . .	40
8.21.1 Detailed Description . . . . .	40
8.21.2 Field Documentation . . . . .	40
8.21.2.1 executingImage . . . . .	40
8.21.2.2 imageIDElement . . . . .	40
8.21.2.3 imageIDSize . . . . .	40



8.21.2.4	imageType	41
8.21.2.5	maxImages	41
8.22	FMSImageList Struct Reference	41
8.22.1	Detailed Description	41
8.22.2	Field Documentation	41
8.22.2.1	imageIDEntries	41
8.22.2.2	listSize	41
8.23	FMSPrefImageList Struct Reference	41
8.23.1	Detailed Description	41
8.23.2	Field Documentation	42
8.23.2.1	listEntries	42
8.23.2.2	listSize	42
8.24	hdrSSInfo Struct Reference	42
8.24.1	Detailed Description	42
8.24.2	Field Documentation	42
8.24.2.1	ecio	42
8.24.2.2	io	42
8.24.2.3	rsi	42
8.24.2.4	sinr	42
8.25	image_info_t Struct Reference	42
8.25.1	Field Documentation	43
8.25.1.1	buildID	43
8.25.1.2	buildIDLen	43
8.25.1.3	imageType	43
8.25.1.4	uniqueID	43
8.26	ipv6AddressInfo Struct Reference	43
8.26.1	Detailed Description	43
8.26.2	Field Documentation	43
8.26.2.1	IPAddressV6	43
8.26.2.2	IPv6PrefixLen	43

8.27 LibPackGPRSRequestedQoS Struct Reference . . . . .	43
8.27.1 Detailed Description . . . . .	43
8.27.2 Field Documentation . . . . .	44
8.27.2.1 delayClass . . . . .	44
8.27.2.2 meanThroughputClass . . . . .	44
8.27.2.3 peakThroughputClass . . . . .	44
8.27.2.4 precedenceClass . . . . .	44
8.27.2.5 reliabilityClass . . . . .	44
8.28 LibpackProfile3GPP Struct Reference . . . . .	44
8.28.1 Detailed Description . . . . .	45
8.28.2 Field Documentation . . . . .	49
8.28.2.1 pAddrAllocPref . . . . .	49
8.28.2.2 pAPNClass . . . . .	49
8.28.2.3 pAPNDisabledFlag . . . . .	49
8.28.2.4 pAPNName . . . . .	49
8.28.2.5 pAPNnameSize . . . . .	49
8.28.2.6 pAuthenticationPref . . . . .	49
8.28.2.7 pGPRSMinimumQoS . . . . .	49
8.28.2.8 pGPRSRequestedQos . . . . .	49
8.28.2.9 pImCnFlag . . . . .	49
8.28.2.10 pIPv4AddrPref . . . . .	49
8.28.2.11 pIPv6AddPref . . . . .	49
8.28.2.12 pPassword . . . . .	49
8.28.2.13 pPasswordSize . . . . .	49
8.28.2.14 pPcscfAddrUsingDhcp . . . . .	49
8.28.2.15 pPcscfAddrUsingPCO . . . . .	49
8.28.2.16 pPDNInactivTimeout . . . . .	49
8.28.2.17 pPdpAccessConFlag . . . . .	49
8.28.2.18 pPdpContext . . . . .	49
8.28.2.19 pPdpDataCompType . . . . .	50

8.28.2.20 pPdpHdrCompType . . . . .	50
8.28.2.21 pPDPTtype . . . . .	50
8.28.2.22 pPriDNSIPv4AddPref . . . . .	50
8.28.2.23 pPriDNSIPv6addpref . . . . .	50
8.28.2.24 pPrimaryID . . . . .	50
8.28.2.25 pProfilename . . . . .	50
8.28.2.26 pProfilenameSize . . . . .	50
8.28.2.27 pQosClassID . . . . .	50
8.28.2.28 pSecDNSIPv4AddPref . . . . .	50
8.28.2.29 pSecDNSIPv6addpref . . . . .	50
8.28.2.30 pSecondaryFlag . . . . .	50
8.28.2.31 pTFTID1Params . . . . .	50
8.28.2.32 pTFTID2Params . . . . .	50
8.28.2.33 pUMTSMinQoS . . . . .	50
8.28.2.34 pUMTSMinQoSSigInd . . . . .	50
8.28.2.35 pUMTSReqQoS . . . . .	50
8.28.2.36 pUMTSReqQoSSigInd . . . . .	50
8.28.2.37 pUsername . . . . .	50
8.28.2.38 pUsernameSize . . . . .	50
8.29 LibpackProfile3GPP2 Struct Reference . . . . .	50
8.29.1 Detailed Description . . . . .	51
8.29.2 Field Documentation . . . . .	54
8.29.2.1 pAllowLinger . . . . .	54
8.29.2.2 pAPNClass3GPP2 . . . . .	55
8.29.2.3 pAPNEnabled3GPP2 . . . . .	55
8.29.2.4 pApnString . . . . .	55
8.29.2.5 pApnStringSize . . . . .	55
8.29.2.6 pAppPriority . . . . .	55
8.29.2.7 pAppType . . . . .	55
8.29.2.8 pAuthPassword . . . . .	55

8.29.2.9 pAuthPasswordSize . . . . .	55
8.29.2.10 pAuthProtocol . . . . .	55
8.29.2.11 pAuthRetryCount . . . . .	55
8.29.2.12 pAuthTimeout . . . . .	55
8.29.2.13 pDataMode . . . . .	55
8.29.2.14 pDataRate . . . . .	55
8.29.2.15 plpcpAckTimeout . . . . .	55
8.29.2.16 plpcpCreqRetryCount . . . . .	55
8.29.2.17 plsPcscfAddressNedded . . . . .	55
8.29.2.18 pLcpAckTimeout . . . . .	55
8.29.2.19 pLcpCreqRetryCount . . . . .	55
8.29.2.20 pNegoDnsSrvrPref . . . . .	55
8.29.2.21 pPDNInactivTimeout3GPP2 . . . . .	55
8.29.2.22 pPdnType . . . . .	55
8.29.2.23 pPppSessCloseTimer1x . . . . .	55
8.29.2.24 pPppSessCloseTimerDO . . . . .	55
8.29.2.25 pPrimaryV4DnsAddress . . . . .	56
8.29.2.26 pPriV6DnsAddress . . . . .	56
8.29.2.27 pRATType . . . . .	56
8.29.2.28 pSecondaryV4DnsAddress . . . . .	56
8.29.2.29 pSecV6DnsAddress . . . . .	56
8.29.2.30 pUserId . . . . .	56
8.29.2.31 pUserIdSize . . . . .	56
8.30 LibPackprofile_3GPP Struct Reference . . . . .	56
8.30.1 Detailed Description . . . . .	57
8.30.2 Field Documentation . . . . .	60
8.30.2.1 pAddrAllocPref . . . . .	60
8.30.2.2 pAPNClass . . . . .	60
8.30.2.3 pAPNDisabledFlag . . . . .	60
8.30.2.4 pAPNName . . . . .	60

8.30.2.5	pAPNnameSize	60
8.30.2.6	pAuthenticationPref	60
8.30.2.7	pGPRSMinimumQoS	61
8.30.2.8	pGPRSRequestedQos	61
8.30.2.9	pImCnFlag	61
8.30.2.10	pIPv4AddrPref	61
8.30.2.11	pIPv6AddPref	61
8.30.2.12	pPassword	61
8.30.2.13	pPasswordSize	61
8.30.2.14	pPscfAddrUsingDhcp	61
8.30.2.15	pPscfAddrUsingPCO	61
8.30.2.16	pPDNInactivTimeout	61
8.30.2.17	pPdpAccessConFlag	61
8.30.2.18	pPdpContext	61
8.30.2.19	pPdpDataCompType	61
8.30.2.20	pPdpHdrCompType	61
8.30.2.21	pPDPTtype	61
8.30.2.22	pPriDNSIPv4AddPref	61
8.30.2.23	pPriDNSIPv6addpref	61
8.30.2.24	pPrimaryID	61
8.30.2.25	pProfilename	61
8.30.2.26	pProfilenameSize	61
8.30.2.27	pQosClassID	61
8.30.2.28	pSecDNSIPv4AddPref	61
8.30.2.29	pSecDNSIPv6addpref	61
8.30.2.30	pSecondaryFlag	62
8.30.2.31	pTFTID1Params	62
8.30.2.32	pTFTID2Params	62
8.30.2.33	pUMTSMInQoS	62
8.30.2.34	pUMTSMInQoSSigInd	62

8.30.2.35 pUMTSReqQoS . . . . .	62
8.30.2.36 pUMTSReqQoSSigInd . . . . .	62
8.30.2.37 pUsername . . . . .	62
8.30.2.38 pUsernameSize . . . . .	62
8.31 LibPackprofile_3GPP2 Struct Reference . . . . .	62
8.31.1 Detailed Description . . . . .	63
8.31.2 Field Documentation . . . . .	66
8.31.2.1 pAllowLinger . . . . .	66
8.31.2.2 pAPNClass3GPP2 . . . . .	66
8.31.2.3 pAPNEnabled3GPP2 . . . . .	66
8.31.2.4 pApnString . . . . .	66
8.31.2.5 pApnStringSize . . . . .	66
8.31.2.6 pAppPriority . . . . .	66
8.31.2.7 pAppType . . . . .	66
8.31.2.8 pAuthPassword . . . . .	66
8.31.2.9 pAuthPassword_tSize . . . . .	66
8.31.2.10 pAuthProtocol . . . . .	66
8.31.2.11 pAuthRetryCount . . . . .	66
8.31.2.12 pAuthTimeout . . . . .	66
8.31.2.13 pDataMode . . . . .	66
8.31.2.14 pDataRate . . . . .	66
8.31.2.15 plpcpAckTimeout . . . . .	66
8.31.2.16 plpcpCreqRetryCount . . . . .	67
8.31.2.17 plsPcscfAddressNedded . . . . .	67
8.31.2.18 pLcpAckTimeout . . . . .	67
8.31.2.19 pLcpCreqRetryCount . . . . .	67
8.31.2.20 pNegoDnsSrvrPref . . . . .	67
8.31.2.21 pPDNInactivTimeout3GPP2 . . . . .	67
8.31.2.22 pPdnType . . . . .	67
8.31.2.23 pPppSessCloseTimer1x . . . . .	67

8.31.2.24 pPppSessCloseTimerDO . . . . .	67
8.31.2.25 pPrimaryV4DnsAddress . . . . .	67
8.31.2.26 pPriV6DnsAddress . . . . .	67
8.31.2.27 pRATType . . . . .	67
8.31.2.28 pSecondaryV4DnsAddress . . . . .	67
8.31.2.29 pSecV6DnsAddress . . . . .	67
8.31.2.30 pUserId . . . . .	67
8.31.2.31 pUserIdSize . . . . .	67
8.32 LibPackQosClassID Struct Reference . . . . .	67
8.32.1 Detailed Description . . . . .	67
8.32.2 Field Documentation . . . . .	68
8.32.2.1 gDIBitRate . . . . .	68
8.32.2.2 gUIBitRate . . . . .	68
8.32.2.3 maxDIBitRate . . . . .	68
8.32.2.4 maxUIBitRate . . . . .	68
8.32.2.5 QCI . . . . .	68
8.33 LibPackTFTIDParams Struct Reference . . . . .	68
8.33.1 Detailed Description . . . . .	69
8.33.2 Field Documentation . . . . .	70
8.33.2.1 destPortRangeEnd . . . . .	70
8.33.2.2 destPortRangeStart . . . . .	70
8.33.2.3 eValid . . . . .	70
8.33.2.4 filterId . . . . .	70
8.33.2.5 flowLabel . . . . .	70
8.33.2.6 IPSECSPi . . . . .	70
8.33.2.7 ipVersion . . . . .	70
8.33.2.8 nextHeader . . . . .	70
8.33.2.9 pSourceIP . . . . .	70
8.33.2.10 sourceIPMask . . . . .	70
8.33.2.11 srcPortRangeEnd . . . . .	70

8.33.2.12 srcPortRangeStart . . . . .	70
8.33.2.13 tosMask . . . . .	70
8.34 LibPackUMTSQoS Struct Reference . . . . .	70
8.34.1 Detailed Description . . . . .	70
8.34.2 Field Documentation . . . . .	72
8.34.2.1 deliveryErrSDU . . . . .	72
8.34.2.2 grntDownlinkBitrate . . . . .	72
8.34.2.3 grntUplinkBitrate . . . . .	72
8.34.2.4 maxDownlinkBitrate . . . . .	72
8.34.2.5 maxSDUSize . . . . .	72
8.34.2.6 maxUplinkBitrate . . . . .	72
8.34.2.7 qosDeliveryOrder . . . . .	72
8.34.2.8 resBerRatio . . . . .	72
8.34.2.9 sduErrorRatio . . . . .	72
8.34.2.10 trafficClass . . . . .	72
8.34.2.11 trafficPriority . . . . .	72
8.34.2.12 transferDelay . . . . .	72
8.35 LibPackUMTSReqQoSsigInd Struct Reference . . . . .	72
8.35.1 Detailed Description . . . . .	73
8.35.2 Field Documentation . . . . .	73
8.35.2.1 SigInd . . . . .	73
8.35.2.2 UMTSReqQoS . . . . .	73
8.36 loc_BdsSV Struct Reference . . . . .	73
8.36.1 Detailed Description . . . . .	73
8.36.2 Field Documentation . . . . .	74
8.36.2.1 id . . . . .	74
8.36.2.2 mask . . . . .	74
8.37 loc_BdsSVInfo Struct Reference . . . . .	74
8.37.1 Detailed Description . . . . .	74
8.37.2 Field Documentation . . . . .	74



8.37.2.1	len	74
8.37.2.2	pSV	74
8.38	loc_CellDb Struct Reference	74
8.38.1	Detailed Description	74
8.38.2	Field Documentation	75
8.38.2.1	mask	75
8.39	loc_ClkInfo Struct Reference	75
8.39.1	Detailed Description	75
8.39.2	Field Documentation	76
8.39.2.1	mask	76
8.40	loc_GnssData Struct Reference	76
8.40.1	Detailed Description	77
8.40.2	Field Documentation	77
8.40.2.1	mask	78
8.41	loc_gpsTime Struct Reference	78
8.41.1	Detailed Description	78
8.41.2	Field Documentation	78
8.41.2.1	gpsTimeOfWeekMs	78
8.41.2.2	gpsWeek	78
8.42	loc_LocApplicationInfo Struct Reference	78
8.42.1	Detailed Description	78
8.42.2	Field Documentation	79
8.42.2.1	appNameLength	79
8.42.2.2	appProviderLength	79
8.42.2.3	appVersionLength	79
8.42.2.4	appVersionValid	79
8.42.2.5	pAppName	79
8.42.2.6	pAppProvider	79
8.42.2.7	pAppVersion	79
8.43	loc_precisionDilution Struct Reference	79

8.43.1 Detailed Description . . . . .	80
8.43.2 Field Documentation . . . . .	80
8.43.2.1 HDOP . . . . .	80
8.43.2.2 PDOP . . . . .	80
8.43.2.3 VDOP . . . . .	80
8.44 loc_satelliteInfo Struct Reference . . . . .	80
8.44.1 Detailed Description . . . . .	80
8.44.2 Field Documentation . . . . .	82
8.44.2.1 azimuth . . . . .	82
8.44.2.2 elevation . . . . .	82
8.44.2.3 gnssSvId . . . . .	82
8.44.2.4 healthStatus . . . . .	82
8.44.2.5 snr . . . . .	82
8.44.2.6 svInfoMask . . . . .	82
8.44.2.7 svListLen . . . . .	82
8.44.2.8 svStatus . . . . .	82
8.44.2.9 system . . . . .	82
8.44.2.10 validMask . . . . .	82
8.45 loc_sensorDataUsage Struct Reference . . . . .	82
8.45.1 Detailed Description . . . . .	83
8.45.2 Field Documentation . . . . .	83
8.45.2.1 aidingIndicatorMask . . . . .	83
8.45.2.2 usageMask . . . . .	83
8.46 loc_SV Struct Reference . . . . .	83
8.46.1 Detailed Description . . . . .	83
8.46.2 Field Documentation . . . . .	84
8.46.2.1 id . . . . .	84
8.46.2.2 mask . . . . .	84
8.46.2.3 system . . . . .	84
8.47 loc_SVInfo Struct Reference . . . . .	84

8.47.1 Detailed Description . . . . .	84
8.47.2 Field Documentation . . . . .	85
8.47.2.1 len . . . . .	85
8.47.2.2 pSV . . . . .	85
8.48 loc_svUsedforFix Struct Reference . . . . .	85
8.48.1 Detailed Description . . . . .	85
8.48.2 Field Documentation . . . . .	85
8.48.2.1 gnssSvUsedList . . . . .	85
8.48.2.2 gnssSvUsedList_len . . . . .	85
8.49 lteSSInfo Struct Reference . . . . .	85
8.49.1 Detailed Description . . . . .	86
8.49.2 Field Documentation . . . . .	86
8.49.2.1 rsrp . . . . .	86
8.49.2.2 rsrq . . . . .	86
8.49.2.3 rssi . . . . .	86
8.49.2.4 snr . . . . .	86
8.50 messageModeTlv Struct Reference . . . . .	86
8.50.1 Detailed Description . . . . .	86
8.50.2 Field Documentation . . . . .	86
8.50.2.1 MessageModelInfo . . . . .	86
8.50.2.2 TlvPresent . . . . .	86
8.51 nas_acqOrderPref Struct Reference . . . . .	86
8.51.1 Detailed Description . . . . .	87
8.51.2 Field Documentation . . . . .	87
8.51.2.1 acqOrdeLen . . . . .	87
8.51.2.2 pAcqOrder . . . . .	87
8.52 nas_AddCDMASysInfo Struct Reference . . . . .	87
8.52.1 Detailed Description . . . . .	87
8.52.2 Field Documentation . . . . .	88
8.52.2.1 geoSysIdx . . . . .	88

8.52.2.2	regPrd	88
8.53	nas_AddSysInfo Struct Reference	88
8.53.1	Detailed Description	88
8.53.2	Field Documentation	88
8.53.2.1	cellBroadcastCap	88
8.53.2.2	geoSysIdx	88
8.54	nas_CallBarringSysInfo Struct Reference	88
8.54.1	Detailed Description	89
8.54.2	Field Documentation	89
8.54.2.1	csBarStatus	89
8.54.2.2	psBarStatus	89
8.55	nas_callBarStatus Struct Reference	89
8.55.1	Detailed Description	89
8.55.2	Field Documentation	90
8.55.2.1	csBarStatus	90
8.55.2.2	psBarStatus	90
8.56	nas_CDMAECIOThresh Struct Reference	90
8.56.1	Detailed Description	90
8.56.2	Field Documentation	91
8.56.2.1	CDMAECIOThreshListLen	91
8.56.2.2	pCDMAECIOThreshList	91
8.57	nas_CDMAInfo Struct Reference	91
8.57.1	Detailed Description	91
8.57.2	Field Documentation	92
8.57.2.1	baseId	92
8.57.2.2	baseLat	92
8.57.2.3	baseLong	92
8.57.2.4	nid	92
8.57.2.5	refpn	92
8.57.2.6	sid	92

8.58 nas_CDMARSSIThresh Struct Reference . . . . .	92
8.58.1 Detailed Description . . . . .	92
8.58.2 Field Documentation . . . . .	92
8.58.2.1 CDMARSSIThreshListLen . . . . .	92
8.58.2.2 pCDMARSSIThreshList . . . . .	92
8.59 nas_CDMA SysInfo Struct Reference . . . . .	92
8.59.1 Detailed Description . . . . .	93
8.59.2 Field Documentation . . . . .	95
8.59.2.1 baseId . . . . .	95
8.59.2.2 baseLat . . . . .	95
8.59.2.3 baseLong . . . . .	95
8.59.2.4 bsInfoValid . . . . .	95
8.59.2.5 bsPRev . . . . .	96
8.59.2.6 bsPRevValid . . . . .	96
8.59.2.7 ccsSupported . . . . .	96
8.59.2.8 ccsSupportedValid . . . . .	96
8.59.2.9 cdmaSysIdValid . . . . .	96
8.59.2.10 isSysPrIMatch . . . . .	96
8.59.2.11 isSysPrIMatchValid . . . . .	96
8.59.2.12 MCC . . . . .	96
8.59.2.13 MNC . . . . .	96
8.59.2.14 networkId . . . . .	96
8.59.2.15 networkIdValid . . . . .	96
8.59.2.16 packetZone . . . . .	96
8.59.2.17 packetZoneValid . . . . .	96
8.59.2.18 pRevInUse . . . . .	96
8.59.2.19 pRevInUseValid . . . . .	96
8.59.2.20 sysInfoCDMA . . . . .	96
8.59.2.21 systemID . . . . .	96
8.60 nas_CDMA SysInfoExt Struct Reference . . . . .	96

8.60.1 Detailed Description . . . . .	96
8.60.2 Field Documentation . . . . .	97
8.60.2.1 imsi_11_12 . . . . .	97
8.60.2.2 MCC . . . . .	97
8.61 nas_cellParams Struct Reference . . . . .	97
8.61.1 Detailed Description . . . . .	97
8.61.2 Field Documentation . . . . .	98
8.61.2.1 pci . . . . .	98
8.61.2.2 rsrp . . . . .	98
8.61.2.3 rsrq . . . . .	98
8.61.2.4 rssi . . . . .	98
8.61.2.5 srxlev . . . . .	98
8.62 nas_CommInfo Struct Reference . . . . .	98
8.62.1 Detailed Description . . . . .	98
8.62.2 Field Documentation . . . . .	99
8.62.2.1 imsRegState . . . . .	99
8.62.2.2 modemMode . . . . .	99
8.62.2.3 psState . . . . .	99
8.62.2.4 systemMode . . . . .	99
8.62.2.5 temperature . . . . .	99
8.63 nas_CSGID Struct Reference . . . . .	99
8.63.1 Detailed Description . . . . .	100
8.63.2 Field Documentation . . . . .	100
8.63.2.1 id . . . . .	100
8.63.2.2 mcc . . . . .	100
8.63.2.3 mnc . . . . .	100
8.63.2.4 mncPcsDigits . . . . .	100
8.63.2.5 rat . . . . .	100
8.64 nas_currentPLMN Struct Reference . . . . .	100
8.64.1 Detailed Description . . . . .	101

8.64.2	Field Documentation	101
8.64.2.1	MCC	101
8.64.2.2	MNC	101
8.64.2.3	netDescr	101
8.64.2.4	netDescrLength	101
8.65	nas_dataSrvCapabilities Struct Reference	101
8.65.1	Detailed Description	101
8.65.2	Field Documentation	102
8.65.2.1	dataCapabilities	102
8.65.2.2	dataCapabilitiesLen	102
8.66	nas_detailSvcInfo Struct Reference	102
8.66.1	Detailed Description	102
8.66.2	Field Documentation	103
8.66.2.1	hdrHybrid	103
8.66.2.2	hdrSrvStatus	103
8.66.2.3	isSysForbidden	103
8.66.2.4	srvCapability	103
8.66.2.5	srvStatus	104
8.67	nas_ecioListElement Struct Reference	104
8.67.1	Detailed Description	104
8.67.2	Field Documentation	104
8.67.2.1	ecio	104
8.67.2.2	radiolf	104
8.68	nas_errorRateListElement Struct Reference	104
8.68.1	Detailed Description	104
8.68.2	Field Documentation	105
8.68.2.1	errorRate	105
8.68.2.2	radiolf	105
8.69	nas_GERANInfo Struct Reference	105
8.69.1	Detailed Description	106

8.69.2	Field Documentation	107
8.69.2.1	arfcn	107
8.69.2.2	bsic	107
8.69.2.3	cellID	107
8.69.2.4	insNmrCellInfo	107
8.69.2.5	lac	107
8.69.2.6	nmrInst	107
8.69.2.7	plmn	107
8.69.2.8	rxLev	107
8.69.2.9	timingAdvance	107
8.70	nas_geranInstInfo Struct Reference	107
8.70.1	Detailed Description	107
8.70.2	Field Documentation	108
8.70.2.1	geranArfcn	108
8.70.2.2	geranBsicBcc	108
8.70.2.3	geranBsicNcc	108
8.70.2.4	geranRssi	108
8.71	nas_gsmCellInfo Struct Reference	108
8.71.1	Detailed Description	108
8.71.2	Field Documentation	109
8.71.2.1	arfcn	109
8.71.2.2	band1900	109
8.71.2.3	bsicId	109
8.71.2.4	cellIdValid	109
8.71.2.5	rssi	109
8.71.2.6	srxlev	109
8.72	nas_GSMRSSIThresh Struct Reference	109
8.72.1	Detailed Description	109
8.72.2	Field Documentation	109
8.72.2.1	GSMRSSIThreshListLen	109



8.72.2.2	pGSMRSSIThreshList	109
8.73	nas_GSMSrvStatusInfo Struct Reference	109
8.73.1	Detailed Description	110
8.73.2	Field Documentation	110
8.73.2.1	isPrefDataPath	110
8.73.2.2	srvStatus	110
8.73.2.3	trueSrvStatus	110
8.74	nas_GSMSysInfo Struct Reference	110
8.74.1	Detailed Description	111
8.74.2	Field Documentation	113
8.74.2.1	cellId	113
8.74.2.2	cellIdValid	113
8.74.2.3	dtmSupp	113
8.74.2.4	dtmSuppValid	113
8.74.2.5	egprsSupp	113
8.74.2.6	egprsSuppValid	113
8.74.2.7	lac	113
8.74.2.8	lacValid	113
8.74.2.9	MCC	113
8.74.2.10	MNC	113
8.74.2.11	networkIdValid	113
8.74.2.12	regRejectInfoValid	113
8.74.2.13	rejCause	113
8.74.2.14	rejectSrvDomain	113
8.74.2.15	sysInfoGSM	113
8.75	nas_HDRECIOThresh Struct Reference	113
8.75.1	Detailed Description	113
8.75.2	Field Documentation	114
8.75.2.1	HDRECIOThreshListLen	114
8.75.2.2	pHDRECIOThreshList	114

8.76 nas_HDRIOTresh Struct Reference . . . . .	114
8.76.1 Detailed Description . . . . .	114
8.76.2 Field Documentation . . . . .	114
8.76.2.1 HDRIOTreshListLen . . . . .	114
8.76.2.2 pHDRIOTreshList . . . . .	114
8.77 nas_HDRRSSITresh Struct Reference . . . . .	114
8.77.1 Detailed Description . . . . .	115
8.77.2 Field Documentation . . . . .	115
8.77.2.1 HDRRSSITreshListLen . . . . .	115
8.77.2.2 pHDRRSSITreshList . . . . .	115
8.78 nas_HDRSINRThreshold Struct Reference . . . . .	115
8.78.1 Detailed Description . . . . .	115
8.78.2 Field Documentation . . . . .	116
8.78.2.1 HDRSINRThreshListLen . . . . .	116
8.78.2.2 pHDRSINRThreshList . . . . .	116
8.79 nas_HDRSysInfo Struct Reference . . . . .	116
8.79.1 Detailed Description . . . . .	116
8.79.2 Field Documentation . . . . .	118
8.79.2.1 hdrActiveProt . . . . .	118
8.79.2.2 hdrActiveProtValid . . . . .	118
8.79.2.3 hdrPersonality . . . . .	118
8.79.2.4 hdrPersonalityValid . . . . .	118
8.79.2.5 is856SysId . . . . .	118
8.79.2.6 is856SysIdValid . . . . .	118
8.79.2.7 isSysPrIMatch . . . . .	118
8.79.2.8 isSysPrIMatchValid . . . . .	118
8.79.2.9 sysInfoHDR . . . . .	118
8.80 nas_infoInterFreq Struct Reference . . . . .	118
8.80.1 Detailed Description . . . . .	118
8.80.2 Field Documentation . . . . .	119

8.80.2.1	cell_resel_priority	119
8.80.2.2	cellInterFreqParams	119
8.80.2.3	cells_len	119
8.80.2.4	earfcn	119
8.80.2.5	threshXHigh	119
8.80.2.6	threshXLow	119
8.81	nas_lteGsmCellInfo Struct Reference	119
8.81.1	Detailed Description	120
8.81.2	Field Documentation	120
8.81.2.1	cellReselPriority	120
8.81.2.2	cells_len	120
8.81.2.3	GsmCellInfo	120
8.81.2.4	nccPermitted	120
8.81.2.5	threshGsmHigh	120
8.81.2.6	threshGsmLow	120
8.82	nas_LTEInfo Struct Reference	121
8.82.1	Detailed Description	121
8.82.2	Field Documentation	122
8.82.2.1	band	122
8.82.2.2	bandwidth	122
8.82.2.3	emmConnState	122
8.82.2.4	emmState	122
8.82.2.5	emmSubState	122
8.82.2.6	RXChan	122
8.82.2.7	TXChan	123
8.83	nas_LTEInfoInterfreq Struct Reference	123
8.83.1	Detailed Description	123
8.83.2	Field Documentation	123
8.83.2.1	freqsLen	123
8.83.2.2	InfoInterfreq	123

8.83.2.3 ueIdle	123
8.84 nas_LTEInfoIntrafreq Struct Reference	123
8.84.1 Detailed Description	124
8.84.2 Field Documentation	125
8.84.2.1 CellParams	125
8.84.2.2 cellReselPriority	125
8.84.2.3 cellsLen	125
8.84.2.4 earfcn	125
8.84.2.5 globalCellId	125
8.84.2.6 plmn	125
8.84.2.7 servingCellId	125
8.84.2.8 sIntraSearch	125
8.84.2.9 sNonIntraSearch	125
8.84.2.10 tac	125
8.84.2.11 threshServingLow	125
8.84.2.12 ueIdle	126
8.85 nas_LTEInfoNeighboringGSM Struct Reference	126
8.85.1 Detailed Description	126
8.85.2 Field Documentation	126
8.85.2.1 freqsLen	126
8.85.2.2 LteGsmCellInfo	126
8.85.2.3 ueIdle	126
8.86 nas_LTEInfoNeighboringWCDMA Struct Reference	126
8.86.1 Detailed Description	127
8.86.2 Field Documentation	127
8.86.2.1 freqsLen	127
8.86.2.2 LTEWCDMACellInfo	127
8.86.2.3 ueIdle	127
8.87 nas_lteRsrpInformation Struct Reference	127
8.87.1 Detailed Description	127

8.87.2	Field Documentation	127
8.87.2.1	rsrlevel	128
8.88	nas_LTERSRPThresh Struct Reference	128
8.88.1	Detailed Description	128
8.88.2	Field Documentation	128
8.88.2.1	LTERSRPThreshListLen	128
8.88.2.2	pLTERSRPThreshList	128
8.89	nas_LTERSRQThresh Struct Reference	128
8.89.1	Detailed Description	128
8.89.2	Field Documentation	129
8.89.2.1	LTERSRQThreshListLen	129
8.89.2.2	pLTERSRQThreshList	129
8.90	nas_LTERSSIThresh Struct Reference	129
8.90.1	Detailed Description	129
8.90.2	Field Documentation	129
8.90.2.1	LTERSSIThreshListLen	129
8.90.2.2	pLTERSSIThreshList	129
8.91	nas_LTESigRptConfig Struct Reference	129
8.91.1	Detailed Description	129
8.91.2	Field Documentation	130
8.91.2.1	avgPeriod	130
8.91.2.2	rptRate	130
8.92	nas_LteSnrinformation Struct Reference	130
8.92.1	Detailed Description	130
8.92.2	Field Documentation	131
8.92.2.1	snrlevel	131
8.93	nas_LTESNRThreshold Struct Reference	131
8.93.1	Detailed Description	131
8.93.2	Field Documentation	131
8.93.2.1	LTESNRThreshListLen	131

8.93.2.2	pLTESNRThreshList	131
8.94	nas_LTESysInfo Struct Reference	131
8.94.1	Detailed Description	132
8.94.2	Field Documentation	133
8.94.2.1	cellId	133
8.94.2.2	cellIdValid	133
8.94.2.3	lac	133
8.94.2.4	lacValid	133
8.94.2.5	MCC	133
8.94.2.6	MNC	133
8.94.2.7	networkIdValid	133
8.94.2.8	regRejectInfoValid	133
8.94.2.9	rejCause	134
8.94.2.10	rejectSrvDomain	134
8.94.2.11	sysInfoLTE	134
8.94.2.12	tac	134
8.94.2.13	tacValid	134
8.95	nas_lteWcdmaCellInfo Struct Reference	134
8.95.1	Detailed Description	134
8.95.2	Field Documentation	135
8.95.2.1	cellReselPriority	135
8.95.2.2	cellsLen	135
8.95.2.3	threshXhigh	135
8.95.2.4	threshXlow	135
8.95.2.5	uarfcn	135
8.95.2.6	WCDMACellInfo	135
8.96	nas_MNRInfo Struct Reference	135
8.96.1	Detailed Description	135
8.96.2	Field Documentation	135
8.96.2.1	mcc	135

8.96.2.2	mnc	135
8.96.2.3	rat	136
8.97	nas_netSelectionPref Struct Reference	136
8.97.1	Detailed Description	136
8.97.2	Field Documentation	136
8.97.2.1	mcc	136
8.97.2.2	mnc	136
8.97.2.3	netReg	136
8.98	nas_nmrCellInfo Struct Reference	136
8.98.1	Detailed Description	137
8.98.2	Field Documentation	137
8.98.2.1	nmrArfcn	137
8.98.2.2	nmrBsic	137
8.98.2.3	nmrCellID	137
8.98.2.4	nmrLac	137
8.98.2.5	nmrPlmn	138
8.98.2.6	nmrRxLev	138
8.99	nas_PhyCaAggPcellInfo Struct Reference	138
8.99.1	Detailed Description	138
8.99.2	Field Documentation	138
8.99.2.1	dl_bw_value	138
8.99.2.2	freq	138
8.99.2.3	iLTEbandValue	138
8.99.2.4	pci	139
8.99.2.5	TlvPresent	139
8.100	nas_PhyCaAggScellIDBw Struct Reference	139
8.100.1	Detailed Description	139
8.100.2	Field Documentation	139
8.100.2.1	dl_bw_value	139
8.100.2.2	TlvPresent	139

8.101nas_PhyCaAggScellIndex Struct Reference . . . . .	139
8.101.1 Detailed Description . . . . .	139
8.101.2 Field Documentation . . . . .	140
8.101.2.1 scell_idx . . . . .	140
8.101.2.2 TlvPresent . . . . .	140
8.102nas_PhyCaAggScellIndType Struct Reference . . . . .	140
8.102.1 Detailed Description . . . . .	140
8.102.2 Field Documentation . . . . .	140
8.102.2.1 freq . . . . .	140
8.102.2.2 pci . . . . .	140
8.102.2.3 scell_state . . . . .	140
8.102.2.4 TlvPresent . . . . .	140
8.103nas_PhyCaAggScellInfo Struct Reference . . . . .	141
8.103.1 Detailed Description . . . . .	141
8.103.2 Field Documentation . . . . .	143
8.103.2.1 dl_bw_value . . . . .	143
8.103.2.2 freq . . . . .	143
8.103.2.3 iLTEbandValue . . . . .	143
8.103.2.4 pci . . . . .	143
8.103.2.5 scell_state . . . . .	143
8.103.2.6 TlvPresent . . . . .	143
8.104nas_qaQmi3Gpp2TimeZone Struct Reference . . . . .	143
8.104.1 Detailed Description . . . . .	143
8.104.2 Field Documentation . . . . .	144
8.104.2.1 daylightSavings . . . . .	144
8.104.2.2 leapSeconds . . . . .	144
8.104.2.3 localTimeOffset . . . . .	144
8.105nas_QmiNas3GppNetworkInfo Struct Reference . . . . .	144
8.105.1 Detailed Description . . . . .	144
8.105.2 Field Documentation . . . . .	144



8.105.2.1 Description . . . . .	144
8.105.2.2 Forbidden . . . . .	144
8.105.2.3 InUse . . . . .	144
8.105.2.4 MCC . . . . .	144
8.105.2.5 MNC . . . . .	144
8.105.2.6 Preferred . . . . .	144
8.105.2.7 Roaming . . . . .	145
8.106nas_QmiNas3GppNetworkRAT Struct Reference . . . . .	145
8.106.1 Detailed Description . . . . .	145
8.106.2 Field Documentation . . . . .	145
8.106.2.1 MCC . . . . .	145
8.106.2.2 MNC . . . . .	145
8.106.2.3 RAT . . . . .	145
8.107nas_QmisNasPcsDigit Struct Reference . . . . .	145
8.107.1 Detailed Description . . . . .	146
8.107.2 Field Documentation . . . . .	146
8.107.2.1 includes_pcs_digit . . . . .	146
8.107.2.2 MCC . . . . .	146
8.107.2.3 MNC . . . . .	146
8.108nas_RejectReasonTlv Struct Reference . . . . .	146
8.108.1 Detailed Description . . . . .	146
8.108.2 Field Documentation . . . . .	146
8.108.2.1 rejectCause . . . . .	146
8.108.2.2 serviceDomain . . . . .	146
8.108.2.3 TlvPresent . . . . .	147
8.109nas_RFInfoTlv Struct Reference . . . . .	147
8.109.1 Detailed Description . . . . .	147
8.109.2 Field Documentation . . . . .	147
8.109.2.1 activeBandClass . . . . .	147
8.109.2.2 activeChannel . . . . .	147

8.109.2.3 radiolInterface . . . . .	147
8.109.2.4 radiolInterfaceSize . . . . .	147
8.109.2.5 TlvPresent . . . . .	147
8.110nas_roamIndList Struct Reference . . . . .	147
8.110.1 Detailed Description . . . . .	147
8.110.2 Field Documentation . . . . .	148
8.110.2.1 numInstances . . . . .	148
8.110.2.2 radiolInterface . . . . .	148
8.110.2.3 roamIndicator . . . . .	148
8.111nas_rsrqInformation Struct Reference . . . . .	148
8.111.1 Detailed Description . . . . .	148
8.111.2 Field Documentation . . . . .	149
8.111.2.1 radiolf . . . . .	149
8.111.2.2 rsrq . . . . .	149
8.112nas_RxSigInfo Struct Reference . . . . .	149
8.112.1 Detailed Description . . . . .	149
8.112.2 Field Documentation . . . . .	150
8.112.2.1 isRadioTuned . . . . .	150
8.112.2.2 rsrp . . . . .	150
8.112.2.3 rxChainIndex . . . . .	150
8.112.2.4 rxPower . . . . .	150
8.113nas_rxSignalStrengthListElement Struct Reference . . . . .	150
8.113.1 Detailed Description . . . . .	150
8.113.2 Field Documentation . . . . .	150
8.113.2.1 radiolf . . . . .	150
8.113.2.2 rxSignalStrength . . . . .	151
8.114nas_SccRxInfo Struct Reference . . . . .	151
8.114.1 Detailed Description . . . . .	151
8.114.2 Field Documentation . . . . .	151
8.114.2.1 numInstances . . . . .	151

8.114.2.2 rsrq . . . . .	151
8.114.2.3 sigInfo . . . . .	151
8.114.2.4 snr . . . . .	152
8.114.2.5 TlvPresent . . . . .	152
8.115nas_servSystem Struct Reference . . . . .	152
8.115.1 Detailed Description . . . . .	152
8.115.2 Field Documentation . . . . .	153
8.115.2.1 csAttachState . . . . .	153
8.115.2.2 numRadioInterfaces . . . . .	153
8.115.2.3 psAttachState . . . . .	153
8.115.2.4 radiolInterface . . . . .	153
8.115.2.5 regState . . . . .	153
8.115.2.6 selNetwork . . . . .	153
8.116nas_SignalStrengthTlv Struct Reference . . . . .	153
8.116.1 Detailed Description . . . . .	153
8.116.2 Field Documentation . . . . .	154
8.116.2.1 radiolInterface . . . . .	154
8.116.2.2 signalStrength . . . . .	154
8.116.2.3 TlvPresent . . . . .	154
8.117nas_SLQSSignalStrengthsIndReq Struct Reference . . . . .	154
8.117.1 Detailed Description . . . . .	154
8.117.2 Field Documentation . . . . .	155
8.117.2.1 ecioDelta . . . . .	155
8.117.2.2 ecioThresholdList . . . . .	155
8.117.2.3 ecioThresholdListLen . . . . .	155
8.117.2.4 ioDelta . . . . .	155
8.117.2.5 lteRsrpDelta . . . . .	155
8.117.2.6 lteSnrDelta . . . . .	155
8.117.2.7 rsrqDelta . . . . .	155
8.117.2.8 rxSignalStrengthDelta . . . . .	155

8.117.2.9 sinrDelta . . . . .	155
8.117.2.10 sinrThresholdList . . . . .	155
8.117.2.11 sinrThresholdListLen . . . . .	155
8.118 nas_SLQSSignalStrengthsInformation Struct Reference . . . . .	155
8.118.1 Detailed Description . . . . .	155
8.118.2 Field Documentation . . . . .	156
8.118.2.1 eciInfo . . . . .	156
8.118.2.2 errorRateInfo . . . . .	156
8.118.2.3 io . . . . .	156
8.118.2.4 lteRsrpinfo . . . . .	156
8.118.2.5 lteSnrinfo . . . . .	156
8.118.2.6 rsrqInfo . . . . .	156
8.118.2.7 rxSignalStrengthInfo . . . . .	156
8.118.2.8 sinr . . . . .	156
8.119 nas_SLQSSignalStrengthsTlv Struct Reference . . . . .	156
8.119.1 Detailed Description . . . . .	156
8.119.2 Field Documentation . . . . .	156
8.119.2.1 sSLQSSignalStrengthsInfo . . . . .	156
8.119.2.2 TlvPresent . . . . .	156
8.120 nas_SrvStatusInfo Struct Reference . . . . .	156
8.120.1 Detailed Description . . . . .	157
8.120.2 Field Documentation . . . . .	157
8.120.2.1 isPrefDataPath . . . . .	157
8.120.2.2 srvStatus . . . . .	157
8.121 nas_sysInfoCommon Struct Reference . . . . .	157
8.121.1 Detailed Description . . . . .	157
8.121.2 Field Documentation . . . . .	159
8.121.2.1 isSysForbidden . . . . .	159
8.121.2.2 isSysForbiddenValid . . . . .	159
8.121.2.3 roamStatus . . . . .	159

8.121.2.4 roamStatusValid . . . . .	159
8.121.2.5 srvCapability . . . . .	159
8.121.2.6 srvCapabilityValid . . . . .	159
8.121.2.7 srvDomain . . . . .	160
8.121.2.8 srvDomainValid . . . . .	160
8.122nas_TDSCDMAECIOThresh Struct Reference . . . . .	160
8.122.1 Detailed Description . . . . .	160
8.122.2 Field Documentation . . . . .	160
8.122.2.1 pTDSCDMAECIOThreshList . . . . .	160
8.122.2.2 TDSCDMAECIOThreshListLen . . . . .	160
8.123nas_TDSCDMARSCPThresh Struct Reference . . . . .	160
8.123.1 Detailed Description . . . . .	160
8.123.2 Field Documentation . . . . .	161
8.123.2.1 pTDSCDMARSCPThreshList . . . . .	161
8.123.2.2 TDSCDMARSCPThreshListLen . . . . .	161
8.124nas_TDSCDMARSSIThresh Struct Reference . . . . .	161
8.124.1 Detailed Description . . . . .	161
8.124.2 Field Documentation . . . . .	161
8.124.2.1 pTDSCDMARSSIThreshList . . . . .	161
8.124.2.2 TDSCDMARSSIThreshListLen . . . . .	161
8.125nas_TDSCDMASINRThresh Struct Reference . . . . .	161
8.125.1 Detailed Description . . . . .	162
8.125.2 Field Documentation . . . . .	162
8.125.2.1 pTDSCDMASINRThreshList . . . . .	162
8.125.2.2 TDSCDMASINRThreshListLen . . . . .	162
8.126nas_timeInfo Struct Reference . . . . .	162
8.126.1 Detailed Description . . . . .	162
8.126.2 Field Documentation . . . . .	164
8.126.2.1 day . . . . .	164
8.126.2.2 dayLtSavingAdj . . . . .	164

8.126.2.3 dayOfWeek . . . . .	164
8.126.2.4 hour . . . . .	164
8.126.2.5 minute . . . . .	164
8.126.2.6 month . . . . .	164
8.126.2.7 radiolInterface . . . . .	164
8.126.2.8 second . . . . .	164
8.126.2.9 timeZone . . . . .	164
8.126.2.10TlvPresent . . . . .	164
8.126.2.11year . . . . .	164
8.127nas_UMTSInfo Struct Reference . . . . .	164
8.127.1 Detailed Description . . . . .	165
8.127.2 Field Documentation . . . . .	165
8.127.2.1 cellID . . . . .	166
8.127.2.2 ecio . . . . .	166
8.127.2.3 geranInst . . . . .	166
8.127.2.4 GeranInstInfo . . . . .	166
8.127.2.5 lac . . . . .	166
8.127.2.6 plmn . . . . .	166
8.127.2.7 psc . . . . .	166
8.127.2.8 rscp . . . . .	166
8.127.2.9 uarfcn . . . . .	166
8.127.2.10umtsInst . . . . .	166
8.127.2.11UMTSInstInfo . . . . .	166
8.128nas_UMTSinstInfo Struct Reference . . . . .	166
8.128.1 Detailed Description . . . . .	166
8.128.2 Field Documentation . . . . .	167
8.128.2.1 umtsEcio . . . . .	167
8.128.2.2 umtsPsc . . . . .	167
8.128.2.3 umtsRscp . . . . .	167
8.128.2.4 umtsUarfcn . . . . .	167

8.129nas_umtsLTENbrCell Struct Reference . . . . .	167
8.129.1 Detailed Description . . . . .	167
8.129.2 Field Documentation . . . . .	168
8.129.2.1 cellsTDD . . . . .	168
8.129.2.2 earfcn . . . . .	168
8.129.2.3 pci . . . . .	168
8.129.2.4 rsrp . . . . .	168
8.129.2.5 rsrq . . . . .	168
8.129.2.6 srxlev . . . . .	168
8.130nas_UniversalTime Struct Reference . . . . .	168
8.130.1 Detailed Description . . . . .	168
8.130.2 Field Documentation . . . . .	169
8.130.2.1 day . . . . .	169
8.130.2.2 dayOfWeek . . . . .	169
8.130.2.3 hour . . . . .	169
8.130.2.4 minute . . . . .	169
8.130.2.5 month . . . . .	169
8.130.2.6 second . . . . .	169
8.130.2.7 year . . . . .	169
8.131nas_wcdmaCellInfo Struct Reference . . . . .	169
8.131.1 Detailed Description . . . . .	170
8.131.2 Field Documentation . . . . .	170
8.131.2.1 cpich_ecno . . . . .	170
8.131.2.2 cpich_rscp . . . . .	170
8.131.2.3 psc . . . . .	170
8.131.2.4 srxlev . . . . .	170
8.132nas_WCDMAECIOThresh Struct Reference . . . . .	170
8.132.1 Detailed Description . . . . .	170
8.132.2 Field Documentation . . . . .	171
8.132.2.1 pWCDMAECIOThreshList . . . . .	171

8.132.2.2 WCDMAECIOThreshListLen . . . . .	171
8.133nas_WCDMAInfoLTENNeighborCell Struct Reference . . . . .	171
8.133.1 Detailed Description . . . . .	171
8.133.2 Field Documentation . . . . .	172
8.133.2.1 UMTSLTENbrCell . . . . .	172
8.133.2.2 umtsLTENbrCellLen . . . . .	172
8.133.2.3 wcdmaRRCState . . . . .	172
8.134nas_WCDMARSSIThresh Struct Reference . . . . .	172
8.134.1 Detailed Description . . . . .	172
8.134.2 Field Documentation . . . . .	172
8.134.2.1 pWCDMARSSIThreshList . . . . .	172
8.134.2.2 WCDMARSSIThreshListLen . . . . .	172
8.135nas_WCDMASysInfo Struct Reference . . . . .	172
8.135.1 Detailed Description . . . . .	173
8.135.2 Field Documentation . . . . .	175
8.135.2.1 cellId . . . . .	175
8.135.2.2 cellIdValid . . . . .	175
8.135.2.3 hsCallStatus . . . . .	175
8.135.2.4 hsCallStatusValid . . . . .	175
8.135.2.5 hsInd . . . . .	175
8.135.2.6 hsIndValid . . . . .	175
8.135.2.7 lac . . . . .	175
8.135.2.8 lacValid . . . . .	175
8.135.2.9 MCC . . . . .	176
8.135.2.10MNC . . . . .	176
8.135.2.11networkIdValid . . . . .	176
8.135.2.12psc . . . . .	176
8.135.2.13pscValid . . . . .	176
8.135.2.14regRejectInfoValid . . . . .	176
8.135.2.15rejCause . . . . .	176



8.135.2.16rejectSrvDomain . . . . .	176
8.135.2.17sysInfoWCDMA . . . . .	176
8.136NASBandPreferenceTlv Struct Reference . . . . .	176
8.136.1 Field Documentation . . . . .	176
8.136.1.1 band_pref . . . . .	176
8.136.1.2 TlvPresent . . . . .	176
8.137NASEmergencyModeTlv Struct Reference . . . . .	176
8.137.1 Field Documentation . . . . .	177
8.137.1.1 EmerMode . . . . .	177
8.137.1.2 TlvPresent . . . . .	177
8.138NasGetLTCEphyCalInfo Struct Reference . . . . .	177
8.138.1 Field Documentation . . . . .	177
8.138.1.1 PhyCaAggPcellInfo . . . . .	177
8.138.1.2 PhyCaAggScellDIBw . . . . .	177
8.138.1.3 PhyCaAggScellIndex . . . . .	177
8.138.1.4 PhyCaAggScellIndType . . . . .	177
8.138.1.5 PhyCaAggScellInfo . . . . .	177
8.139NASGWAcqOrderPrefTlv Struct Reference . . . . .	177
8.139.1 Field Documentation . . . . .	177
8.139.1.1 GWAcqOrderPref . . . . .	177
8.139.1.2 TlvPresent . . . . .	177
8.140NASLTEBandPreferenceTlv Struct Reference . . . . .	177
8.140.1 Field Documentation . . . . .	178
8.140.1.1 LTEBandPref . . . . .	178
8.140.1.2 TlvPresent . . . . .	178
8.141NASLteNasReleaseInfoTlv Struct Reference . . . . .	178
8.141.1 Field Documentation . . . . .	178
8.141.1.1 nas_major . . . . .	178
8.141.1.2 nas_minor . . . . .	178
8.141.1.3 nas_release . . . . .	178

8.141.1.4 TlvPresent . . . . .	178
8.142NASModePreferenceTlv Struct Reference . . . . .	178
8.142.1 Field Documentation . . . . .	178
8.142.1.1 ModePref . . . . .	178
8.142.1.2 TlvPresent . . . . .	178
8.143NASNetSelPreferenceTlv Struct Reference . . . . .	178
8.143.1 Field Documentation . . . . .	179
8.143.1.1 NetSelPref . . . . .	179
8.143.1.2 TlvPresent . . . . .	179
8.144NASOTAMessageTlv Struct Reference . . . . .	179
8.144.1 Field Documentation . . . . .	179
8.144.1.1 data_buf . . . . .	179
8.144.1.2 data_len . . . . .	179
8.144.1.3 message_type . . . . .	179
8.144.1.4 TlvPresent . . . . .	179
8.145NASPhyCaAggPcellInfo Struct Reference . . . . .	179
8.145.1 Detailed Description . . . . .	180
8.145.2 Field Documentation . . . . .	180
8.145.2.1 dl_bw_value . . . . .	180
8.145.2.2 freq . . . . .	180
8.145.2.3 iLTEbandValue . . . . .	180
8.145.2.4 pci . . . . .	180
8.145.2.5 TlvPresent . . . . .	180
8.146NASPhyCaAggScellIDBw Struct Reference . . . . .	180
8.146.1 Detailed Description . . . . .	180
8.146.2 Field Documentation . . . . .	181
8.146.2.1 dl_bw_value . . . . .	181
8.146.2.2 TlvPresent . . . . .	181
8.147NASPhyCaAggScellIndex Struct Reference . . . . .	181
8.147.1 Detailed Description . . . . .	181

8.147.2 Field Documentation	181
8.147.2.1 scell_idx	181
8.147.2.2 TlvPresent	181
8.148NASPhyCaAggScellIndType Struct Reference	181
8.148.1 Detailed Description	182
8.148.2 Field Documentation	182
8.148.2.1 freq	182
8.148.2.2 pci	182
8.148.2.3 scell_state	182
8.148.2.4 TlvPresent	182
8.149NASPhyCaAggScellInfo Struct Reference	182
8.149.1 Detailed Description	182
8.149.2 Field Documentation	183
8.149.2.1 dl_bw_value	183
8.149.2.2 freq	183
8.149.2.3 iLTEbandValue	183
8.149.2.4 pci	183
8.149.2.5 scell_state	183
8.149.2.6 TlvPresent	183
8.150NASPRLPreferenceTlv Struct Reference	183
8.150.1 Field Documentation	184
8.150.1.1 PRLPref	184
8.150.1.2 TlvPresent	184
8.151NASQmiCbkNasSwtOTAMessageInd Struct Reference	184
8.151.1 Field Documentation	184
8.151.1.1 nasRelInfoTlv	184
8.151.1.2 otaMsgTlv	184
8.151.1.3 timeTlv	184
8.152NASQmiCbkNasSystemSelPrefInd Struct Reference	184
8.152.1 Field Documentation	185

8.152.1.1 BPTlv . . . . .	185
8.152.1.2 EMTlv . . . . .	185
8.152.1.3 GWAOPTlv . . . . .	185
8.152.1.4 LBPTlv . . . . .	185
8.152.1.5 MPTlv . . . . .	185
8.152.1.6 NSPTlv . . . . .	185
8.152.1.7 PRLPTlv . . . . .	185
8.152.1.8 RPTlv . . . . .	185
8.152.1.9 SDPTlv . . . . .	185
8.153NASRoamPreferenceTlv Struct Reference . . . . .	185
8.153.1 Field Documentation . . . . .	185
8.153.1.1 RoamPref . . . . .	185
8.153.1.2 TlvPresent . . . . .	185
8.154NASServDomainPrefTlv Struct Reference . . . . .	185
8.154.1 Field Documentation . . . . .	186
8.154.1.1 SrvDomainPref . . . . .	186
8.154.1.2 TlvPresent . . . . .	186
8.155NASServingSystemInfo Struct Reference . . . . .	186
8.155.1 Detailed Description . . . . .	186
8.155.2 Field Documentation . . . . .	187
8.155.2.1 csAttachState . . . . .	187
8.155.2.2 hdrPersonality . . . . .	187
8.155.2.3 psAttachState . . . . .	187
8.155.2.4 radiolInterfaceList . . . . .	187
8.155.2.5 radiolInterfaceNo . . . . .	187
8.155.2.6 registrationState . . . . .	187
8.155.2.7 selectedNetwork . . . . .	187
8.156NASTimeInfoTlv Struct Reference . . . . .	187
8.156.1 Field Documentation . . . . .	188
8.156.1.1 time . . . . .	188

8.156.1.2 TlvPresent . . . . .	188
8.157newMTMessageTlv Struct Reference . . . . .	188
8.157.1 Detailed Description . . . . .	188
8.157.2 Field Documentation . . . . .	188
8.157.2.1 MTMessageInfo . . . . .	188
8.157.2.2 TlvPresent . . . . .	188
8.158pack_dms_ActivateAutomatic_t Struct Reference . . . . .	188
8.158.1 Detailed Description . . . . .	188
8.158.2 Field Documentation . . . . .	189
8.158.2.1 actCode . . . . .	189
8.159pack_dms_GetCustFeaturesV2_t Struct Reference . . . . .	189
8.159.1 Detailed Description . . . . .	189
8.159.2 Field Documentation . . . . .	189
8.159.2.1 cust_id . . . . .	189
8.159.2.2 list_type . . . . .	189
8.159.2.3 Tlvresult . . . . .	189
8.160pack_dms_ResetToFactoryDefaults_t Struct Reference . . . . .	189
8.160.1 Detailed Description . . . . .	189
8.160.2 Field Documentation . . . . .	190
8.160.2.1 spc . . . . .	190
8.161pack_dms_SetActivationStatusCallback_t Struct Reference . . . . .	190
8.161.1 Detailed Description . . . . .	190
8.161.2 Field Documentation . . . . .	190
8.161.2.1 activationState . . . . .	190
8.162pack_dms_SetCrashAction_t Struct Reference . . . . .	190
8.162.1 Detailed Description . . . . .	190
8.162.2 Field Documentation . . . . .	191
8.162.2.1 crashAction . . . . .	191
8.163pack_dms_SetCustFeature_t Struct Reference . . . . .	191
8.163.1 Field Documentation . . . . .	191

8.163.1.1 DHCPRelayEnabled . . . . .	191
8.163.1.2 DisableIMSI . . . . .	191
8.163.1.3 GpsEnable . . . . .	191
8.163.1.4 GPSLPM . . . . .	191
8.163.1.5 GPSSel . . . . .	191
8.163.1.6 IPFamSupport . . . . .	191
8.163.1.7 IsVoiceEnabled . . . . .	191
8.163.1.8 RMAutoConnect . . . . .	191
8.163.1.9 SMSSupport . . . . .	191
8.164pack_dms_SetCustFeaturesV2_t Struct Reference . . . . .	191
8.164.1 Detailed Description . . . . .	192
8.164.2 Field Documentation . . . . .	192
8.164.2.1 cust_id . . . . .	192
8.164.2.2 cust_value . . . . .	192
8.164.2.3 Tlvresult . . . . .	192
8.164.2.4 value_length . . . . .	192
8.165pack_dms_SetEventReport_t Struct Reference . . . . .	192
8.165.1 Field Documentation . . . . .	192
8.165.1.1 mode . . . . .	192
8.166pack_dms_SetPower_t Struct Reference . . . . .	192
8.166.1 Field Documentation . . . . .	193
8.166.1.1 mode . . . . .	193
8.166.1.2 Tlvresult . . . . .	193
8.167pack_dms_SetUSBComp_t Struct Reference . . . . .	193
8.167.1 Field Documentation . . . . .	193
8.167.1.1 Tlvresult . . . . .	193
8.167.1.2 USBComp . . . . .	193
8.168pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . . .	193
8.168.1 Detailed Description . . . . .	193
8.168.2 Field Documentation . . . . .	193

8.168.2.1 resetInfoInd . . . . .	193
8.169pack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . . .	193
8.169.1 Detailed Description . . . . .	194
8.169.2 Field Documentation . . . . .	194
8.169.2.1 clear . . . . .	194
8.170pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . . .	194
8.170.1 Detailed Description . . . . .	194
8.170.2 Field Documentation . . . . .	194
8.170.2.1 pDestSMSContent . . . . .	194
8.170.2.2 pDestSMSNum . . . . .	194
8.171pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . . .	194
8.171.1 Detailed Description . . . . .	195
8.171.2 Field Documentation . . . . .	195
8.171.2.1 manString . . . . .	195
8.171.2.2 modelString . . . . .	195
8.171.2.3 plasmaIDString . . . . .	195
8.171.2.4 swVerString . . . . .	195
8.172pack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . . .	195
8.172.1 Detailed Description . . . . .	195
8.172.2 Field Documentation . . . . .	196
8.172.2.1 nameString . . . . .	196
8.172.2.2 versionString . . . . .	196
8.173pack_dms_UIMChangePIN_t Struct Reference . . . . .	196
8.173.1 Detailed Description . . . . .	196
8.173.2 Field Documentation . . . . .	196
8.173.2.1 id . . . . .	196
8.173.2.2 newValue . . . . .	196
8.173.2.3 oldValue . . . . .	196
8.174pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . . .	196
8.174.1 Detailed Description . . . . .	197

8.174.2 Field Documentation . . . . .	197
8.174.2.1 facility . . . . .	197
8.175pack_dms_UIMGetICCID_t Struct Reference . . . . .	197
8.175.1 Detailed Description . . . . .	197
8.175.2 Field Documentation . . . . .	197
8.175.2.1 Tlvresult . . . . .	197
8.176pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . . .	197
8.176.1 Detailed Description . . . . .	198
8.176.2 Field Documentation . . . . .	198
8.176.2.1 facility . . . . .	198
8.176.2.2 facilityCk . . . . .	198
8.176.2.3 facilityState . . . . .	198
8.177pack_dms_UIMSetPINProtection_t Struct Reference . . . . .	198
8.177.1 Detailed Description . . . . .	198
8.177.2 Field Documentation . . . . .	199
8.177.2.1 bEnable . . . . .	199
8.177.2.2 id . . . . .	199
8.177.2.3 value . . . . .	199
8.178pack_dms_UIMUnblockControlKey_t Struct Reference . . . . .	199
8.178.1 Detailed Description . . . . .	199
8.178.2 Field Documentation . . . . .	200
8.178.2.1 facility . . . . .	200
8.178.2.2 facilityCk . . . . .	200
8.179pack_dms_UIMUnblockPIN_t Struct Reference . . . . .	200
8.179.1 Detailed Description . . . . .	200
8.179.2 Field Documentation . . . . .	200
8.179.2.1 id . . . . .	200
8.179.2.2 newPin . . . . .	200
8.179.2.3 pukValue . . . . .	200
8.180pack_dms_UIMVerifyPIN_t Struct Reference . . . . .	200



8.180.1 Detailed Description . . . . .	201
8.180.2 Field Documentation . . . . .	201
8.180.2.1 id . . . . .	201
8.180.2.2 value . . . . .	201
8.181pack_fms_GetImagesPreference_t Struct Reference . . . . .	201
8.181.1 Detailed Description . . . . .	201
8.181.2 Field Documentation . . . . .	201
8.181.2.1 Tlvresult . . . . .	201
8.182pack_fms_GetStoredImages_t Struct Reference . . . . .	201
8.182.1 Detailed Description . . . . .	202
8.182.2 Field Documentation . . . . .	202
8.182.2.1 Tlvresult . . . . .	202
8.183pack_fms_SetImagesPreference_t Struct Reference . . . . .	202
8.183.1 Detailed Description . . . . .	202
8.183.2 Field Documentation . . . . .	203
8.183.2.1 bForceDownload . . . . .	203
8.183.2.2 imageListSize . . . . .	203
8.183.2.3 modemindex . . . . .	203
8.183.2.4 pImageList . . . . .	203
8.183.2.5 Tlvresult . . . . .	203
8.184pack_loc_Delete_Assist_Data_t Struct Reference . . . . .	203
8.184.1 Detailed Description . . . . .	203
8.184.2 Field Documentation . . . . .	204
8.184.2.1 pBdsSVInfo . . . . .	204
8.184.2.2 pCellDb . . . . .	204
8.184.2.3 pCikInfo . . . . .	204
8.184.2.4 pGnssData . . . . .	204
8.184.2.5 pSVInfo . . . . .	204
8.184.2.6 Tlvresult . . . . .	204
8.185pack_loc_EventRegister_t Struct Reference . . . . .	204

8.185.1 Detailed Description	204
8.185.2 Field Documentation	206
8.185.2.1 eventRegister	206
8.185.2.2 Tlvresult	206
8.186pack_loc_SetExtPowerState_t Struct Reference	206
8.186.1 Detailed Description	206
8.186.2 Field Documentation	206
8.186.2.1 extPowerState	206
8.186.2.2 Tlvresult	206
8.187pack_loc_SetOperationMode_t Struct Reference	207
8.187.1 Detailed Description	207
8.187.2 Field Documentation	207
8.187.2.1 mode	207
8.187.2.2 Tlvresult	207
8.188pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference	207
8.188.1 Detailed Description	207
8.188.2 Field Documentation	208
8.188.2.1 Tlvresult	208
8.188.2.2 xid	208
8.189pack_loc_SLQSLOCInjectPosition_t Struct Reference	208
8.189.1 Detailed Description	209
8.189.2 Field Documentation	211
8.189.2.1 altitudeSrcInfo	211
8.189.2.2 altitudeWrtEllipsoid	212
8.189.2.3 altitudeWrtMeanSeaLevel	212
8.189.2.4 has_altitudeSrcInfo	212
8.189.2.5 has_altitudeWrtEllipsoid	212
8.189.2.6 has_altitudeWrtMeanSeaLevel	212
8.189.2.7 has_horConfidence	212
8.189.2.8 has_horReliability	212

8.189.2.9 has_horUncCircular . . . . .	212
8.189.2.10 has_latitude . . . . .	212
8.189.2.11 has_longitude . . . . .	212
8.189.2.12 has_positionSrc . . . . .	212
8.189.2.13 has_rawHorConfidence . . . . .	212
8.189.2.14 has_rawHorUncCircular . . . . .	212
8.189.2.15 has_timestampAge . . . . .	212
8.189.2.16 has_timestampUtc . . . . .	212
8.189.2.17 has_vertConfidence . . . . .	212
8.189.2.18 has_vertRelicability . . . . .	212
8.189.2.19 has_vertUnc . . . . .	212
8.189.2.20 horConfidence . . . . .	212
8.189.2.21 horReliability . . . . .	212
8.189.2.22 horUncCircular . . . . .	212
8.189.2.23 attitude . . . . .	212
8.189.2.24 longitude . . . . .	212
8.189.2.25 positionSrc . . . . .	213
8.189.2.26 rawHorConfidence . . . . .	213
8.189.2.27 rawHorUncCircular . . . . .	213
8.189.2.28 timestampAge . . . . .	213
8.189.2.29 timestampUtc . . . . .	213
8.189.2.30 vertConfidence . . . . .	213
8.189.2.31 vertReliability . . . . .	213
8.189.2.32 vertUnc . . . . .	213
8.190 pack_loc_SLQSLOCInjectSensorData_t Struct Reference . . . . .	213
8.190.1 Detailed Description . . . . .	213
8.190.2 Field Documentation . . . . .	214
8.190.2.1 acceleroData . . . . .	214
8.190.2.2 acceleroTimeSrc . . . . .	214
8.190.2.3 accelTemp . . . . .	214

8.190.2.4 gyroData . . . . .	215
8.190.2.5 gyroTemp . . . . .	215
8.190.2.6 gyroTimeSrc . . . . .	215
8.190.2.7 has_acceleroTimeSrc . . . . .	215
8.190.2.8 has_accelTemp . . . . .	215
8.190.2.9 has_accleroData . . . . .	215
8.190.2.10has_gyroData . . . . .	215
8.190.2.11has_gyroTemp . . . . .	215
8.190.2.12has_gyroTimeSrc . . . . .	215
8.190.2.13has_opaqueld . . . . .	215
8.190.2.14opaqueld . . . . .	215
8.191pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . . .	215
8.191.1 Detailed Description . . . . .	215
8.191.2 Field Documentation . . . . .	215
8.191.2.1 timeMsec . . . . .	215
8.191.2.2 timeUncMsec . . . . .	215
8.192pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . . .	216
8.192.1 Detailed Description . . . . .	216
8.192.2 Field Documentation . . . . .	216
8.192.2.1 confidence . . . . .	216
8.192.2.2 has_confidence . . . . .	216
8.192.2.3 state . . . . .	216
8.193pack_loc_Start_t Struct Reference . . . . .	216
8.193.1 Detailed Description . . . . .	217
8.193.2 Field Documentation . . . . .	217
8.193.2.1 pApplicationInfo . . . . .	217
8.193.2.2 pConfigAltitudeAssumed . . . . .	218
8.193.2.3 pHorizontalAccuracyLvl . . . . .	218
8.193.2.4 pIntermediateReportState . . . . .	218
8.193.2.5 pMinIntervalTime . . . . .	218

8.193.2.6 pRecurrenceType . . . . .	218
8.193.2.7 SessionId . . . . .	218
8.193.2.8 Tlvresult . . . . .	218
8.194pack_loc_Stop_t Struct Reference . . . . .	218
8.194.1 Detailed Description . . . . .	218
8.194.2 Field Documentation . . . . .	218
8.194.2.1 SessionId . . . . .	218
8.194.2.2 Tlvresult . . . . .	218
8.195pack_nas_SetACCOLC_t Struct Reference . . . . .	218
8.195.1 Detailed Description . . . . .	219
8.195.2 Field Documentation . . . . .	219
8.195.2.1 accolc . . . . .	219
8.195.2.2 spc . . . . .	219
8.196pack_nas_SetNetworkPreference_t Struct Reference . . . . .	219
8.196.1 Detailed Description . . . . .	219
8.196.2 Field Documentation . . . . .	220
8.196.2.1 Duration . . . . .	220
8.196.2.2 TechnologyPref . . . . .	220
8.196.2.3 Tlvresult . . . . .	220
8.197pack_nas_SLQSGetPLMNName_t Struct Reference . . . . .	220
8.197.1 Detailed Description . . . . .	220
8.197.2 Field Documentation . . . . .	220
8.197.2.1 mcc . . . . .	220
8.197.2.2 mnc . . . . .	220
8.197.2.3 pMncPcsStatus . . . . .	221
8.198pack_nas_SLQSIInitiateNetworkRegistration_t Struct Reference . . . . .	221
8.198.1 Detailed Description . . . . .	221
8.198.2 Field Documentation . . . . .	221
8.198.2.1 pChangeDuration . . . . .	221
8.198.2.2 pMncPcsDigitStatus . . . . .	221

8.198.2.3 pMNRInfo . . . . .	221
8.198.2.4 regAction . . . . .	222
8.199pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . . .	222
8.199.1 Detailed Description . . . . .	222
8.199.2 Field Documentation . . . . .	225
8.199.2.1 pCDMAECIODelta . . . . .	225
8.199.2.2 pCDMAECIOThresh . . . . .	225
8.199.2.3 pCDMARSSIDelta . . . . .	225
8.199.2.4 pCDMARSSIThresh . . . . .	225
8.199.2.5 pGSMRSSIDelta . . . . .	225
8.199.2.6 pGSMRSSIThresh . . . . .	225
8.199.2.7 pHDRECIODelta . . . . .	225
8.199.2.8 pHDRECIOThresh . . . . .	225
8.199.2.9 pHDRIODelta . . . . .	225
8.199.2.10pHDRIOTresh . . . . .	225
8.199.2.11pHDDRSSIDelta . . . . .	225
8.199.2.12pHDDRSSIThresh . . . . .	225
8.199.2.13pHDRSINRDelta . . . . .	225
8.199.2.14pHDRSINRThresh . . . . .	226
8.199.2.15pLTERSRPDelta . . . . .	226
8.199.2.16pLTERSRPThresh . . . . .	226
8.199.2.17pLTERSRQDelta . . . . .	226
8.199.2.18pLTERSRQThresh . . . . .	226
8.199.2.19pLTERSSIDelta . . . . .	226
8.199.2.20pLTERSSIThresh . . . . .	226
8.199.2.21pLTESigRptConfig . . . . .	226
8.199.2.22pLTESNRDelta . . . . .	226
8.199.2.23pLTESNRThresh . . . . .	226
8.199.2.24pTDSCDMAECIODelta . . . . .	226
8.199.2.25pTDSCDMAECIOThresh . . . . .	226

8.199.2.26	pTDSCDMARSCPDelta . . . . .	226
8.199.2.27	pTDSCDMARSCPThresh . . . . .	226
8.199.2.28	pTDSCDMARSSIDelta . . . . .	226
8.199.2.29	pTDSCDMARSSIThresh . . . . .	226
8.199.2.30	pTDSCDMASINRDelta . . . . .	226
8.199.2.31	pTDSCDMASINRThresh . . . . .	226
8.199.2.32	pWCDMAECIODelta . . . . .	226
8.199.2.33	pWCDMAECIOThresh . . . . .	226
8.199.2.34	pWCDMARSSIDelta . . . . .	226
8.199.2.35	pWCDMARSSIThresh . . . . .	226
8.200	pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference . . . . .	226
8.200.1	Detailed Description . . . . .	227
8.200.2	Field Documentation . . . . .	229
8.200.2.1	pDDTMInd . . . . .	229
8.200.2.2	pDualStandByPrefInd . . . . .	229
8.200.2.3	pErrorRateInd . . . . .	229
8.200.2.4	pHDRNewUATIAssInd . . . . .	229
8.200.2.5	pHDRSessionCloseInd . . . . .	229
8.200.2.6	pLTECphyCa . . . . .	229
8.200.2.7	pManagedRoamingInd . . . . .	229
8.200.2.8	pNetworkTimeInd . . . . .	229
8.200.2.9	pServingSystemInd . . . . .	229
8.200.2.10	pSignalStrengthInd . . . . .	229
8.200.2.11	pSubscriptionInfoInd . . . . .	229
8.200.2.12	pSysInfoInd . . . . .	229
8.200.2.13	pSystemSelectionInd . . . . .	229
8.201	pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference . . . . .	229
8.201.1	Detailed Description . . . . .	229
8.201.2	Field Documentation . . . . .	230
8.201.2.1	gsmUmtsDI . . . . .	230

8.201.2.2 gsmUmtsUI . . . . .	230
8.201.2.3 lteEmmDI . . . . .	230
8.201.2.4 lteEmmUI . . . . .	230
8.201.2.5 lteEsmDI . . . . .	230
8.201.2.6 lteEsmUI . . . . .	230
8.201.2.7 pRankIndicatorInd . . . . .	230
8.201.2.8 pTimer . . . . .	231
8.202pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . . .	231
8.202.1 Detailed Description . . . . .	231
8.202.2 Field Documentation . . . . .	231
8.202.2.1 bEnable . . . . .	231
8.202.2.2 pSigIndReq . . . . .	231
8.203pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . . .	231
8.203.1 Detailed Description . . . . .	231
8.203.2 Field Documentation . . . . .	235
8.203.2.1 pAcqOrderPref . . . . .	235
8.203.2.2 pBandPref . . . . .	235
8.203.2.3 pChgDuration . . . . .	235
8.203.2.4 pCSGID . . . . .	235
8.203.2.5 pEmerMode . . . . .	235
8.203.2.6 pGWAcqOrderPref . . . . .	236
8.203.2.7 pLTEBandPref . . . . .	236
8.203.2.8 pMNCIncPCSDigStat . . . . .	236
8.203.2.9 pModePref . . . . .	236
8.203.2.10pNetSelPref . . . . .	236
8.203.2.11pPRLPref . . . . .	236
8.203.2.12pRAT . . . . .	236
8.203.2.13pRoamPref . . . . .	236
8.203.2.14pSrvDomainPref . . . . .	236
8.203.2.15pSrvRegRestriction . . . . .	236



8.203.2.1 <code>pTdsdmaBandPref</code> . . . . .	236
8.204 <code>pack_qmi_t</code> Struct Reference . . . . .	236
8.204.1 Detailed Description . . . . .	236
8.204.2 Field Documentation . . . . .	236
8.204.2.1 <code>msgid</code> . . . . .	236
8.204.2.2 <code>svc</code> . . . . .	237
8.204.2.3 <code>timeout</code> . . . . .	237
8.204.2.4 <code>xid</code> . . . . .	237
8.205 <code>pack_qos_SLQSQoSReadApnExtraParams_t</code> Struct Reference . . . . .	237
8.205.1 Detailed Description . . . . .	237
8.205.2 Field Documentation . . . . .	237
8.205.2.1 <code>apnId</code> . . . . .	237
8.206 <code>pack_qos_SLQSQoSReadDataStats_t</code> Struct Reference . . . . .	237
8.206.1 Detailed Description . . . . .	237
8.206.2 Field Documentation . . . . .	237
8.206.2.1 <code>apnId</code> . . . . .	238
8.207 <code>pack_qos_SLQSSetQoSEventCallback_t</code> Struct Reference . . . . .	238
8.207.1 Detailed Description . . . . .	238
8.207.2 Field Documentation . . . . .	238
8.207.2.1 <code>enable</code> . . . . .	238
8.208 <code>pack_sms_SendSMS_t</code> Struct Reference . . . . .	238
8.208.1 Detailed Description . . . . .	238
8.208.2 Field Documentation . . . . .	239
8.208.2.1 <code>messageFormat</code> . . . . .	239
8.208.2.2 <code>messageSize</code> . . . . .	239
8.208.2.3 <code>pLinktimer</code> . . . . .	239
8.208.2.4 <code>pMessage</code> . . . . .	239
8.209 <code>pack_sms_SetNewSMSCallback_t</code> Struct Reference . . . . .	239
8.209.1 Detailed Description . . . . .	239
8.209.2 Field Documentation . . . . .	239

8.209.2.1 status . . . . .	239
8.210pack_sms_SLQSDelateSMS_t Struct Reference . . . . .	239
8.210.1 Detailed Description . . . . .	239
8.210.2 Field Documentation . . . . .	240
8.210.2.1 pMessageIndex . . . . .	240
8.210.2.2 pMessageMode . . . . .	240
8.210.2.3 pMessageTag . . . . .	240
8.210.2.4 storageType . . . . .	240
8.211pack_sms_SLQSGetSMS_t Struct Reference . . . . .	240
8.211.1 Detailed Description . . . . .	240
8.211.2 Field Documentation . . . . .	241
8.211.2.1 messageIndex . . . . .	241
8.211.2.2 pMessageMode . . . . .	241
8.211.2.3 storageType . . . . .	241
8.212pack_sms_SLQSGetSMSList_t Struct Reference . . . . .	241
8.212.1 Detailed Description . . . . .	241
8.212.2 Field Documentation . . . . .	241
8.212.2.1 pMessageMode . . . . .	241
8.212.2.2 pRequestedTag . . . . .	241
8.212.2.3 storageType . . . . .	241
8.213pack_sms_SLQSModifySMSStatus_t Struct Reference . . . . .	241
8.213.1 Detailed Description . . . . .	242
8.213.2 Field Documentation . . . . .	242
8.213.2.1 messageIndex . . . . .	242
8.213.2.2 messageTag . . . . .	242
8.213.2.3 pMessageMode . . . . .	242
8.213.2.4 storageType . . . . .	242
8.214pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . . .	242
8.214.1 Detailed Description . . . . .	243
8.214.2 Field Documentation . . . . .	244

8.214.2.1 fix_rate . . . . .	244
8.214.2.2 fix_type . . . . .	244
8.214.2.3 function . . . . .	244
8.214.2.4 max_dist . . . . .	244
8.214.2.5 max_time . . . . .	244
8.214.2.6 set_fix_rate . . . . .	244
8.214.2.7 set_fix_type . . . . .	244
8.214.2.8 set_function . . . . .	244
8.214.2.9 set_max_dist . . . . .	244
8.214.2.10set_max_time . . . . .	244
8.215pack_swioama_SLQSOMADMCancelSession_t Struct Reference . . . . .	244
8.215.1 Detailed Description . . . . .	244
8.215.2 Field Documentation . . . . .	244
8.215.2.1 sessionType . . . . .	244
8.216pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference . . . . .	245
8.216.1 Detailed Description . . . . .	245
8.216.2 Field Documentation . . . . .	245
8.216.2.1 SessionType . . . . .	245
8.217pack_swioama_SLQSOMADMSendSelection_t Struct Reference . . . . .	245
8.217.1 Detailed Description . . . . .	245
8.217.2 Field Documentation . . . . .	246
8.217.2.1 pDeferTime . . . . .	246
8.217.2.2 pRejectReason . . . . .	246
8.217.2.3 selection . . . . .	246
8.218pack_swioama_SLQSOMADMSetSettings_t Struct Reference . . . . .	246
8.218.1 Detailed Description . . . . .	246
8.218.2 Field Documentation . . . . .	246
8.218.2.1 FOTAdownload . . . . .	247
8.218.2.2 FOTAUpdate . . . . .	247
8.218.2.3 pAutosdm . . . . .	247

8.218.2.4 pFwAutoCheck . . . . .	247
8.219pack_swima_SLQSOMADMStartSession_t Struct Reference . . . . .	247
8.219.1 Detailed Description . . . . .	247
8.219.2 Field Documentation . . . . .	247
8.219.2.1 sessionType . . . . .	247
8.220pack_uim_ChangePin_t Struct Reference . . . . .	247
8.220.1 Detailed Description . . . . .	247
8.220.2 Field Documentation . . . . .	248
8.220.2.1 changePIN . . . . .	248
8.220.2.2 EncryptedPIN1 . . . . .	248
8.220.2.3 pIndicationToken . . . . .	248
8.220.2.4 pKeyReferenceID . . . . .	248
8.220.2.5 sessionInfo . . . . .	248
8.220.2.6 Tlvresult . . . . .	248
8.221pack_uim_ReadTransparent_t Struct Reference . . . . .	248
8.221.1 Detailed Description . . . . .	249
8.221.2 Field Documentation . . . . .	249
8.221.2.1 fileIndex . . . . .	249
8.221.2.2 pEncryptData . . . . .	249
8.221.2.3 pIndicationToken . . . . .	249
8.221.2.4 readTransparent . . . . .	249
8.221.2.5 sessionInfo . . . . .	249
8.221.2.6 Tlvresult . . . . .	249
8.222pack_uim_SetPinProtection_t Struct Reference . . . . .	249
8.222.1 Detailed Description . . . . .	250
8.222.2 Field Documentation . . . . .	250
8.222.2.1 EncryptedPIN1 . . . . .	250
8.222.2.2 pIndicationToken . . . . .	250
8.222.2.3 pinProtection . . . . .	250
8.222.2.4 pKeyReferenceID . . . . .	250

8.222.2.5 sessionInfo . . . . .	250
8.222.2.6 Tlvresult . . . . .	250
8.223pack_uim_SLQSUIEventRegister_t Struct Reference . . . . .	250
8.223.1 Detailed Description . . . . .	251
8.223.2 Field Documentation . . . . .	251
8.223.2.1 eventMask . . . . .	251
8.224pack_uim_SLQSUIPowerDown_t Struct Reference . . . . .	251
8.224.1 Detailed Description . . . . .	251
8.224.2 Field Documentation . . . . .	251
8.224.2.1 slot . . . . .	251
8.225pack_uim_SLQSUIPowerUp_t Struct Reference . . . . .	251
8.225.1 Detailed Description . . . . .	251
8.225.2 Field Documentation . . . . .	252
8.225.2.1 plgnoreHotSwapSwitch . . . . .	252
8.225.2.2 slot . . . . .	252
8.226pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . . .	252
8.226.1 Detailed Description . . . . .	252
8.226.2 Field Documentation . . . . .	253
8.226.2.1 bLogicalSlot . . . . .	253
8.226.2.2 ulPhysicalSlot . . . . .	253
8.227pack_uim_UnblockPin_t Struct Reference . . . . .	253
8.227.1 Detailed Description . . . . .	253
8.227.2 Field Documentation . . . . .	253
8.227.2.1 EncryptedPIN1 . . . . .	254
8.227.2.2 pIndicationToken . . . . .	254
8.227.2.3 pinProtection . . . . .	254
8.227.2.4 pKeyReferenceID . . . . .	254
8.227.2.5 sessionInfo . . . . .	254
8.227.2.6 Tlvresult . . . . .	254
8.228pack_uim_VerifyPin_t Struct Reference . . . . .	254

8.228.1 Detailed Description . . . . .	254
8.228.2 Field Documentation . . . . .	255
8.228.2.1 pEncryptedPIN1 . . . . .	255
8.228.2.2 pIndicationToken . . . . .	255
8.228.2.3 pKeyReferenceID . . . . .	255
8.228.2.4 sessionInfo . . . . .	255
8.228.2.5 Tlvresult . . . . .	255
8.228.2.6 verifyPIN . . . . .	255
8.229pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . . .	255
8.229.1 Detailed Description . . . . .	255
8.229.2 Field Documentation . . . . .	255
8.229.2.1 pEnableNotification . . . . .	255
8.230pack_wds_GetDefaultProfile_t Struct Reference . . . . .	255
8.230.1 Detailed Description . . . . .	256
8.230.2 Field Documentation . . . . .	256
8.230.2.1 profiletype . . . . .	256
8.231pack_wds_GetDefaultProfileNum_t Struct Reference . . . . .	256
8.231.1 Detailed Description . . . . .	256
8.231.2 Field Documentation . . . . .	256
8.231.2.1 family . . . . .	256
8.231.2.2 type . . . . .	256
8.232pack_wds_GetDormancyState_t Struct Reference . . . . .	256
8.233pack_wds_GetLastMobileIPError_t Struct Reference . . . . .	256
8.234pack_wds_GetMobileIP_t Struct Reference . . . . .	256
8.235pack_wds_GetMobileIPProfile_t Struct Reference . . . . .	256
8.235.1 Detailed Description . . . . .	257
8.235.2 Field Documentation . . . . .	257
8.235.2.1 index . . . . .	257
8.236pack_wds_GetPacketStatistics_t Struct Reference . . . . .	257
8.236.1 Detailed Description . . . . .	257

8.236.2 Field Documentation . . . . .	257
8.236.2.1 pStatMask . . . . .	257
8.237 pack_wds_GetPacketStatus_t Struct Reference . . . . .	257
8.237.1 Detailed Description . . . . .	257
8.237.2 Field Documentation . . . . .	257
8.237.2.1 statmask . . . . .	257
8.238 pack_wds_GetSessionDuration_t Struct Reference . . . . .	258
8.239 pack_wds_RMSetTransferStatistics_t Struct Reference . . . . .	258
8.239.1 Detailed Description . . . . .	258
8.239.2 Field Documentation . . . . .	258
8.239.2.1 RmTrasnferStaticsReq . . . . .	258
8.240 pack_wds_SetAutoconnect_t Struct Reference . . . . .	258
8.240.1 Detailed Description . . . . .	258
8.240.2 Field Documentation . . . . .	258
8.240.2.1 acroamsetting . . . . .	258
8.240.2.2 acsetting . . . . .	258
8.241 pack_wds_SetDefaultProfile_t Struct Reference . . . . .	259
8.241.1 Detailed Description . . . . .	259
8.241.2 Field Documentation . . . . .	259
8.241.2.1 authentication . . . . .	259
8.241.2.2 ipAddress . . . . .	259
8.241.2.3 pApnname . . . . .	259
8.241.2.4 pdpType . . . . .	259
8.241.2.5 pName . . . . .	259
8.241.2.6 pPassword . . . . .	259
8.241.2.7 primaryDNS . . . . .	259
8.241.2.8 profileType . . . . .	259
8.241.2.9 pUsername . . . . .	260
8.241.2.10secondaryDNS . . . . .	260
8.242 pack_wds_SetDefaultProfileNum_t Struct Reference . . . . .	260

8.242.1 Field Documentation . . . . .	260
8.242.1.1 family . . . . .	260
8.242.1.2 index . . . . .	260
8.242.1.3 type . . . . .	260
8.243pack_wds_SetMobileIP_t Struct Reference . . . . .	260
8.243.1 Detailed Description . . . . .	260
8.243.2 Field Documentation . . . . .	260
8.243.2.1 mode . . . . .	260
8.244pack_wds_SetMobileIPParameters_t Struct Reference . . . . .	260
8.244.1 Detailed Description . . . . .	261
8.244.2 Field Documentation . . . . .	261
8.244.2.1 pHA2002bis . . . . .	262
8.244.2.2 pHAAuthenticator . . . . .	262
8.244.2.3 pMode . . . . .	262
8.244.2.4 pReRegPeriod . . . . .	262
8.244.2.5 pReRegTraffic . . . . .	262
8.244.2.6 pRetryInterval . . . . .	262
8.244.2.7 pRetryLimit . . . . .	262
8.244.2.8 pSPC . . . . .	262
8.245pack_wds_SetMobileIPProfile_t Struct Reference . . . . .	262
8.245.1 Detailed Description . . . . .	262
8.245.2 Field Documentation . . . . .	263
8.245.2.1 index . . . . .	263
8.245.2.2 pAAASPI . . . . .	263
8.245.2.3 pAddress . . . . .	263
8.245.2.4 pEnabled . . . . .	263
8.245.2.5 pHASPI . . . . .	263
8.245.2.6 pMNAAA . . . . .	263
8.245.2.7 pMNHA . . . . .	263
8.245.2.8 pNAI . . . . .	263



8.245.2.9 pPrimaryHA . . . . .	263
8.245.2.10 pRevTunneling . . . . .	263
8.245.2.11 pSecondaryHA . . . . .	263
8.245.2.12 spc . . . . .	263
8.246 pack_wds_SLQSCreateProfile_t Struct Reference . . . . .	263
8.246.1 Detailed Description . . . . .	263
8.246.2 Field Documentation . . . . .	264
8.246.2.1 pCurProfile . . . . .	264
8.246.2.2 pProfileId . . . . .	264
8.246.2.3 pProfileType . . . . .	264
8.247 pack_wds_SLQSDeleteProfile_t Struct Reference . . . . .	264
8.247.1 Detailed Description . . . . .	264
8.247.2 Field Documentation . . . . .	264
8.247.2.1 profileIndex . . . . .	264
8.247.2.2 profileType . . . . .	264
8.248 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . . .	264
8.249 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . . .	264
8.250 pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . . .	264
8.250.1 Detailed Description . . . . .	265
8.250.2 Field Documentation . . . . .	265
8.250.2.1 Mask . . . . .	265
8.250.2.2 pReportChannelRate . . . . .	265
8.250.2.3 pReportConnStatus . . . . .	265
8.250.2.4 pReportDataBearerTech . . . . .	265
8.250.2.5 pReportDormStatus . . . . .	265
8.250.2.6 pTransferStatInd . . . . .	265
8.251 pack_wds_SLQSGetProfileSettings_t Struct Reference . . . . .	265
8.251.1 Detailed Description . . . . .	265
8.251.2 Field Documentation . . . . .	266
8.251.2.1 ProfileId . . . . .	266

8.251.2.2 ProfileType . . . . .	266
8.252pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . . .	266
8.252.1 Detailed Description . . . . .	266
8.252.2 Field Documentation . . . . .	267
8.252.2.1 pReqSettings . . . . .	267
8.253pack_wds_SLQSModifyProfile_t Struct Reference . . . . .	267
8.253.1 Detailed Description . . . . .	267
8.253.2 Field Documentation . . . . .	268
8.253.2.1 curProfile . . . . .	268
8.253.2.2 pProfileId . . . . .	268
8.253.2.3 pProfileType . . . . .	268
8.254pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . . .	268
8.254.1 Detailed Description . . . . .	268
8.254.2 Field Documentation . . . . .	269
8.254.2.1 LTEAttachProfileListLen . . . . .	269
8.254.2.2 p3gppRelease . . . . .	270
8.254.2.3 pDefaultPDNEnabled . . . . .	270
8.254.2.4 pLTEAttachProfile . . . . .	270
8.254.2.5 pLTEAttachProfileList . . . . .	270
8.254.2.6 pProfileList . . . . .	270
8.255pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . . .	270
8.255.1 Detailed Description . . . . .	270
8.255.2 Field Documentation . . . . .	270
8.255.2.1 IPFamilyPreference . . . . .	270
8.256pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . . .	270
8.256.1 Detailed Description . . . . .	270
8.256.2 Field Documentation . . . . .	271
8.256.2.1 currentDataBearer . . . . .	271
8.256.2.2 dataBearer . . . . .	271
8.256.2.3 dataSystemStatus . . . . .	271

8.256.2.4 dormancyStatus . . . . .	271
8.256.2.5 interval . . . . .	271
8.256.2.6 mobileIP . . . . .	271
8.256.2.7 transferStats . . . . .	271
8.257pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . . .	271
8.257.1 Detailed Description . . . . .	271
8.257.2 Field Documentation . . . . .	271
8.257.2.1 pProfileId . . . . .	271
8.258pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference . . . . .	271
8.258.1 Detailed Description . . . . .	272
8.258.2 Field Documentation . . . . .	272
8.258.2.1 pHwConfig . . . . .	272
8.258.2.2 pProfileId . . . . .	272
8.258.2.3 pRequestOptionList . . . . .	272
8.259pack_wds_SLQSSSetLoopback_t Struct Reference . . . . .	272
8.259.1 Detailed Description . . . . .	272
8.259.2 Field Documentation . . . . .	273
8.259.2.1 loopbackMode . . . . .	273
8.259.2.2 loopbackMultiplier . . . . .	273
8.260pack_wds_SLQSSStartDataSession_t Struct Reference . . . . .	273
8.260.1 Detailed Description . . . . .	273
8.260.2 Field Documentation . . . . .	274
8.260.2.1 pAuth . . . . .	274
8.260.2.2 pPass . . . . .	274
8.260.2.3 pprofileid3gpp . . . . .	274
8.260.2.4 pprofileid3gpp2 . . . . .	274
8.260.2.5 pTech . . . . .	274
8.260.2.6 pUser . . . . .	274
8.261pack_wds_SLQSSStopDataSession_t Struct Reference . . . . .	274
8.261.1 Detailed Description . . . . .	274

8.261.2 Field Documentation . . . . .	274
8.261.2.1 psid . . . . .	274
8.262pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . . .	274
8.262.1 Detailed Description . . . . .	275
8.262.2 Field Documentation . . . . .	276
8.262.2.1 pCurrChannelRateInd . . . . .	276
8.262.2.2 pCurrDataBearerTechInd . . . . .	276
8.262.2.3 pCurrPrefDataSysInd . . . . .	276
8.262.2.4 pDataBearerTechInd . . . . .	276
8.262.2.5 pDataCallStatusChangeInd . . . . .	276
8.262.2.6 pDataSystemStatusChangeInd . . . . .	276
8.262.2.7 pDormancyStatusInd . . . . .	276
8.262.2.8 pEVDOPageMonPerChangeInd . . . . .	276
8.262.2.9 pMIPStatusInd . . . . .	276
8.262.2.10pTransferStatInd . . . . .	276
8.263pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . . .	276
8.263.1 Detailed Description . . . . .	276
8.263.2 Field Documentation . . . . .	276
8.263.2.1 contextId . . . . .	276
8.263.2.2 contextType . . . . .	276
8.264PackCreateProfileOut Struct Reference . . . . .	277
8.264.1 Field Documentation . . . . .	277
8.264.1.1 ExtErrorCode . . . . .	277
8.264.1.2 ProfileIndex . . . . .	277
8.264.1.3 ProfileType . . . . .	277
8.265packgetDyingGaspCfg Struct Reference . . . . .	277
8.265.1 Detailed Description . . . . .	277
8.265.2 Field Documentation . . . . .	277
8.265.2.1 pDestSMSContent . . . . .	277
8.265.2.2 pDestSMSNum . . . . .	277

8.266packgetDyingGaspStatistics Struct Reference . . . . .	277
8.266.1 Detailed Description . . . . .	278
8.266.2 Field Documentation . . . . .	278
8.266.2.1 pSMSAttemptedFlag . . . . .	278
8.266.2.2 pTimeStamp . . . . .	278
8.267qmiSmsMessageList Struct Reference . . . . .	278
8.267.1 Detailed Description . . . . .	278
8.267.2 Field Documentation . . . . .	278
8.267.2.1 messageIndex . . . . .	278
8.267.2.2 messageTag . . . . .	278
8.268qmiWSDDataBearerTechnology Struct Reference . . . . .	278
8.268.1 Detailed Description . . . . .	279
8.268.2 Field Documentation . . . . .	279
8.268.2.1 currentNetwork . . . . .	279
8.268.2.2 ratMask . . . . .	279
8.268.2.3 soMask . . . . .	279
8.269RFBandInfoElements Struct Reference . . . . .	279
8.269.1 Detailed Description . . . . .	279
8.269.2 Field Documentation . . . . .	279
8.269.2.1 activeBandClass . . . . .	279
8.269.2.2 activeChannel . . . . .	279
8.269.2.3 radiolInterface . . . . .	279
8.270rmTrasnferStaticsReq Struct Reference . . . . .	279
8.270.1 Detailed Description . . . . .	280
8.270.2 Field Documentation . . . . .	280
8.270.2.1 bResetStatistics . . . . .	280
8.270.2.2 ulMask . . . . .	280
8.271sensorData_t Struct Reference . . . . .	280
8.271.1 Detailed Description . . . . .	280
8.271.2 Field Documentation . . . . .	281

8.271.2.1 flags	281
8.271.2.2 sensorDataLen	281
8.271.2.3 timeOfFirstSample	281
8.271.2.4 timeOffset	281
8.271.2.5 xAxis	281
8.271.2.6 yAxis	281
8.271.2.7 zAxis	281
8.272slot_t Struct Reference	281
8.272.1 Detailed Description	281
8.272.2 Field Documentation	282
8.272.2.1 bICCID	282
8.272.2.2 bICCIDLength	282
8.272.2.3 bLogicalSlot	282
8.272.2.4 uPhyCardStatus	282
8.272.2.5 uPhySlotStatus	282
8.273slotInf Struct Reference	282
8.273.1 Detailed Description	283
8.273.2 Field Documentation	284
8.273.2.1 AppStatus	284
8.273.2.2 cardState	284
8.273.2.3 errorState	284
8.273.2.4 numApp	284
8.273.2.5 upinRetries	284
8.273.2.6 upinState	284
8.273.2.7 upukRetries	284
8.274slots_t Struct Reference	284
8.274.1 Field Documentation	284
8.274.1.1 uimSlotStatus	284
8.275sMSCAddress Struct Reference	284
8.275.1 Detailed Description	284

8.275.2 Field Documentation . . . . .	285
8.275.2.1 data . . . . .	285
8.275.2.2 length . . . . .	285
8.276sMSCAddressTlv Struct Reference . . . . .	285
8.276.1 Detailed Description . . . . .	285
8.276.2 Field Documentation . . . . .	285
8.276.2.1 SMSCInfo . . . . .	285
8.276.2.2 TlvPresent . . . . .	285
8.277sMSEtwsMessage Struct Reference . . . . .	285
8.277.1 Detailed Description . . . . .	285
8.277.2 Field Documentation . . . . .	286
8.277.2.1 data . . . . .	286
8.277.2.2 length . . . . .	286
8.277.2.3 notificationType . . . . .	286
8.278sMSEtwsMessageTlv Struct Reference . . . . .	286
8.278.1 Detailed Description . . . . .	286
8.278.2 Field Documentation . . . . .	286
8.278.2.1 EtwsMessageInfo . . . . .	286
8.278.2.2 TlvPresent . . . . .	286
8.279sMSEtwsPlmn Struct Reference . . . . .	286
8.279.1 Detailed Description . . . . .	287
8.279.2 Field Documentation . . . . .	287
8.279.2.1 mobileCountryCode . . . . .	287
8.279.2.2 mobileNetworkCode . . . . .	287
8.280sMSMessageMode Struct Reference . . . . .	287
8.280.1 Detailed Description . . . . .	287
8.280.2 Field Documentation . . . . .	287
8.280.2.1 messageMode . . . . .	287
8.281sMSMTMessage Struct Reference . . . . .	287
8.281.1 Detailed Description . . . . .	288

8.281.2 Field Documentation . . . . .	288
8.281.2.1 messageIndex . . . . .	288
8.281.2.2 storageType . . . . .	288
8.282sMSOnIMS Struct Reference . . . . .	288
8.282.1 Detailed Description . . . . .	288
8.282.2 Field Documentation . . . . .	288
8.282.2.1 smsOnIMS . . . . .	288
8.283sMSOnIMSTlv Struct Reference . . . . .	288
8.283.1 Detailed Description . . . . .	288
8.283.2 Field Documentation . . . . .	289
8.283.2.1 IMSInfo . . . . .	289
8.283.2.2 TlvPresent . . . . .	289
8.284sMSTransferRouteMTMessage Struct Reference . . . . .	289
8.284.1 Detailed Description . . . . .	289
8.284.2 Field Documentation . . . . .	289
8.284.2.1 ackIndicator . . . . .	289
8.284.2.2 data . . . . .	289
8.284.2.3 format . . . . .	289
8.284.2.4 length . . . . .	289
8.284.2.5 transactionID . . . . .	290
8.285tdscdmaSigInfoExt Struct Reference . . . . .	290
8.285.1 Detailed Description . . . . .	290
8.285.2 Field Documentation . . . . .	290
8.285.2.1 ecio . . . . .	290
8.285.2.2 rscp . . . . .	290
8.285.2.3 rssi . . . . .	290
8.285.2.4 sinr . . . . .	290
8.286tempData_t Struct Reference . . . . .	290
8.286.1 Detailed Description . . . . .	290
8.286.2 Field Documentation . . . . .	291



8.286.2.1 temperature	291
8.286.2.2 temperatureDataLen	291
8.286.2.3 timeOfFirstSample	291
8.286.2.4 timeOffset	291
8.286.2.5 timeSource	291
8.287transferRouteMessageTlv Struct Reference	291
8.287.1 Detailed Description	291
8.287.2 Field Documentation	292
8.287.2.1 TlvPresent	292
8.287.2.2 TransferRouteMTMessageInfo	292
8.288transferStatInd Struct Reference	292
8.288.1 Detailed Description	292
8.288.2 Field Documentation	292
8.288.2.1 StatsMask	292
8.288.2.2 StatsPeriod	292
8.289uim_appStatus Struct Reference	292
8.289.1 Detailed Description	293
8.289.2 Field Documentation	295
8.289.2.1 aidLength	295
8.289.2.2 aidVal	295
8.289.2.3 appState	295
8.289.2.4 appType	295
8.289.2.5 persoFeature	295
8.289.2.6 persoRetries	295
8.289.2.7 persoState	295
8.289.2.8 persoUnblockRetries	295
8.289.2.9 pin1Retries	295
8.289.2.10pin1State	295
8.289.2.11pin2Retries	295
8.289.2.12pin2State	295

8.289.2.13 puk1Retries . . . . .	295
8.289.2.14 puk2Retries . . . . .	295
8.289.2.15 univPin . . . . .	295
8.290 uim_cardResult Struct Reference . . . . .	295
8.290.1 Detailed Description . . . . .	296
8.290.2 Field Documentation . . . . .	296
8.290.2.1 sw1 . . . . .	296
8.290.2.2 sw2 . . . . .	296
8.291 uim_cardStatus Struct Reference . . . . .	296
8.291.1 Detailed Description . . . . .	296
8.291.2 Field Documentation . . . . .	297
8.291.2.1 index1xPri . . . . .	297
8.291.2.2 index1xSec . . . . .	297
8.291.2.3 indexGwPri . . . . .	297
8.291.2.4 indexGwSec . . . . .	297
8.291.2.5 numSlot . . . . .	297
8.291.2.6 SlotInfo . . . . .	297
8.292 uim_changeUIMPIN Struct Reference . . . . .	297
8.292.1 Detailed Description . . . . .	297
8.292.2 Field Documentation . . . . .	298
8.292.2.1 oldPINLen . . . . .	298
8.292.2.2 oldPINVal . . . . .	298
8.292.2.3 pinID . . . . .	298
8.292.2.4 pinLen . . . . .	298
8.292.2.5 pinVal . . . . .	298
8.293 uim_encryptedPIN1 Struct Reference . . . . .	298
8.293.1 Detailed Description . . . . .	298
8.293.2 Field Documentation . . . . .	299
8.293.2.1 pin1Len . . . . .	299
8.293.2.2 pin1Val . . . . .	299

8.294uim_fileInfo Struct Reference . . . . .	299
8.294.1 Detailed Description . . . . .	299
8.294.2 Field Documentation . . . . .	299
8.294.2.1 fileID . . . . .	299
8.294.2.2 path . . . . .	300
8.294.2.3 pathLen . . . . .	300
8.295uim_hotSwapStatus Struct Reference . . . . .	300
8.295.1 Detailed Description . . . . .	300
8.295.2 Field Documentation . . . . .	300
8.295.2.1 hotSwap . . . . .	300
8.295.2.2 hotSwapLength . . . . .	300
8.296uim_readResult Struct Reference . . . . .	300
8.296.1 Detailed Description . . . . .	300
8.296.2 Field Documentation . . . . .	301
8.296.2.1 content . . . . .	301
8.296.2.2 contentLen . . . . .	301
8.297uim_readTransparentInfo Struct Reference . . . . .	301
8.297.1 Detailed Description . . . . .	301
8.297.2 Field Documentation . . . . .	301
8.297.2.1 length . . . . .	301
8.297.2.2 offset . . . . .	301
8.298uim_remainingRetries Struct Reference . . . . .	301
8.298.1 Detailed Description . . . . .	302
8.298.2 Field Documentation . . . . .	302
8.298.2.1 unblockLeft . . . . .	302
8.298.2.2 verifyLeft . . . . .	302
8.299uim_sessionInformation Struct Reference . . . . .	302
8.299.1 Detailed Description . . . . .	302
8.299.2 Field Documentation . . . . .	303
8.299.2.1 aid . . . . .	303

8.299.2.2 aidLength . . . . .	303
8.299.2.3 sessionType . . . . .	303
8.300uim_setPINProtection Struct Reference . . . . .	303
8.300.1 Detailed Description . . . . .	303
8.300.2 Field Documentation . . . . .	304
8.300.2.1 pinID . . . . .	304
8.300.2.2 pinLength . . . . .	304
8.300.2.3 pinOperation . . . . .	304
8.300.2.4 pinValue . . . . .	304
8.301uim_slotInfo Struct Reference . . . . .	304
8.301.1 Detailed Description . . . . .	304
8.301.2 Field Documentation . . . . .	305
8.301.2.1 AppStatus . . . . .	305
8.301.2.2 cardState . . . . .	305
8.301.2.3 errorState . . . . .	305
8.301.2.4 numApp . . . . .	305
8.301.2.5 upinRetries . . . . .	305
8.301.2.6 upinState . . . . .	305
8.301.2.7 upukRetries . . . . .	305
8.302uim_UIMSessionInformation Struct Reference . . . . .	305
8.302.1 Detailed Description . . . . .	306
8.302.2 Field Documentation . . . . .	306
8.302.2.1 aid . . . . .	306
8.302.2.2 aidLength . . . . .	306
8.302.2.3 sessionType . . . . .	306
8.303uim_unblockUIMPIN Struct Reference . . . . .	306
8.303.1 Detailed Description . . . . .	306
8.303.2 Field Documentation . . . . .	307
8.303.2.1 newPINLen . . . . .	307
8.303.2.2 newPINVal . . . . .	307

8.303.2.3 pinID . . . . .	307
8.303.2.4 pukLen . . . . .	307
8.303.2.5 pukVal . . . . .	307
8.304uim_verifyUIMPIN Struct Reference . . . . .	307
8.304.1 Detailed Description . . . . .	307
8.304.2 Field Documentation . . . . .	308
8.304.2.1 pinID . . . . .	308
8.304.2.2 pinLen . . . . .	308
8.304.2.3 pinVal . . . . .	308
8.305unpack_dms_GetActivationState_t Struct Reference . . . . .	308
8.305.1 Detailed Description . . . . .	308
8.305.2 Field Documentation . . . . .	309
8.305.2.1 state . . . . .	309
8.306unpack_dms_GetBandCapability_t Struct Reference . . . . .	309
8.306.1 Field Documentation . . . . .	309
8.306.1.1 BandCapability . . . . .	309
8.306.1.2 Tlvresult . . . . .	309
8.307unpack_dms_GetCrashAction_t Struct Reference . . . . .	309
8.307.1 Field Documentation . . . . .	309
8.307.1.1 DevCrashState . . . . .	309
8.307.1.2 Tlvresult . . . . .	309
8.308unpack_dms_GetCustFeature_t Struct Reference . . . . .	309
8.308.1 Field Documentation . . . . .	310
8.308.1.1 DHCPRelayEnabled . . . . .	310
8.308.1.2 DisableIMSI . . . . .	310
8.308.1.3 GpsEnable . . . . .	310
8.308.1.4 GPSLPM . . . . .	310
8.308.1.5 GPSSel . . . . .	310
8.308.1.6 IPFamSupport . . . . .	310
8.308.1.7 IsVoiceEnabled . . . . .	310

8.308.1.8 RMAutoConnect . . . . .	310
8.308.1.9 SMSSupport . . . . .	310
8.308.1.10Tlvresult . . . . .	310
8.309unpack_dms_GetCustFeaturesV2_t Struct Reference . . . . .	310
8.309.1 Detailed Description . . . . .	310
8.309.2 Field Documentation . . . . .	310
8.309.2.1 GetCustomFeatureV2 . . . . .	310
8.309.2.2 Tlvresult . . . . .	310
8.310unpack_dms_GetDeviceCap_t Struct Reference . . . . .	311
8.310.1 Field Documentation . . . . .	311
8.310.1.1 DataServiceCapability . . . . .	311
8.310.1.2 MaxRXChannelRate . . . . .	311
8.310.1.3 MaxTXChannelRate . . . . .	311
8.310.1.4 Radiofaces . . . . .	311
8.310.1.5 RadiofacesSize . . . . .	311
8.310.1.6 SimCapability . . . . .	311
8.310.1.7 Tlvresult . . . . .	311
8.311unpack_dms_GetDeviceCapabilities_t Struct Reference . . . . .	311
8.311.1 Detailed Description . . . . .	311
8.311.2 Field Documentation . . . . .	312
8.311.2.1 dataServiceCaCapability . . . . .	312
8.311.2.2 maxRxChannelRate . . . . .	312
8.311.2.3 maxTxChannelRate . . . . .	312
8.311.2.4 Radiofaces . . . . .	312
8.311.2.5 radiofacesSize . . . . .	312
8.311.2.6 simCapability . . . . .	312
8.312unpack_dms_GetDeviceHardwareRev_t Struct Reference . . . . .	312
8.312.1 Field Documentation . . . . .	312
8.312.1.1 String . . . . .	312
8.312.1.2 stringSize . . . . .	312

8.312.1.3 Tlvresult . . . . .	312
8.313unpack_dms_GetDeviceMfr_t Struct Reference . . . . .	312
8.313.1 Field Documentation . . . . .	313
8.313.1.1 String . . . . .	313
8.313.1.2 stringSize . . . . .	313
8.313.1.3 Tlvresult . . . . .	313
8.314unpack_dms_GetDeviceSerialNumbers_t Struct Reference . . . . .	313
8.314.1 Field Documentation . . . . .	313
8.314.1.1 esnSize . . . . .	313
8.314.1.2 ESNString . . . . .	313
8.314.1.3 imeiSize . . . . .	313
8.314.1.4 IMEIString . . . . .	313
8.314.1.5 imeiSvnSize . . . . .	313
8.314.1.6 ImeiSvnString . . . . .	313
8.314.1.7 meidSize . . . . .	313
8.314.1.8 MEIDString . . . . .	313
8.314.1.9 Tlvresult . . . . .	313
8.315unpack_dms_GetFirmwareInfo_t Struct Reference . . . . .	313
8.315.1 Detailed Description . . . . .	314
8.315.2 Field Documentation . . . . .	314
8.315.2.1 appversion_str . . . . .	314
8.315.2.2 bootversion_str . . . . .	314
8.315.2.3 carrier_str . . . . .	314
8.315.2.4 cur_carr_name . . . . .	314
8.315.2.5 cur_carr_rev . . . . .	314
8.315.2.6 modelid_str . . . . .	314
8.315.2.7 packageid_str . . . . .	314
8.315.2.8 priversion_str . . . . .	314
8.315.2.9 sku_str . . . . .	314
8.315.2.10Tlvresult . . . . .	314

8.316unpack_dms_GetFirmwareRevision_t Struct Reference . . . . .	314
8.316.1 Field Documentation . . . . .	315
8.316.1.1 amssSize . . . . .	315
8.316.1.2 AMSSString . . . . .	315
8.316.1.3 PRIString . . . . .	315
8.316.1.4 Tlvresult . . . . .	315
8.317unpack_dms_GetFirmwareRevisions_t Struct Reference . . . . .	315
8.317.1 Detailed Description . . . . .	315
8.317.2 Field Documentation . . . . .	315
8.317.2.1 amssSize . . . . .	315
8.317.2.2 AMSSString . . . . .	315
8.317.2.3 bootSize . . . . .	315
8.317.2.4 BootString . . . . .	315
8.317.2.5 priSize . . . . .	315
8.317.2.6 PRIString . . . . .	315
8.317.2.7 Tlvresult . . . . .	315
8.318unpack_dms_GetFSN_t Struct Reference . . . . .	316
8.318.1 Field Documentation . . . . .	316
8.318.1.1 String . . . . .	316
8.318.1.2 Tlvresult . . . . .	316
8.319unpack_dms_GetHardwareRevision_t Struct Reference . . . . .	316
8.319.1 Detailed Description . . . . .	316
8.319.2 Field Documentation . . . . .	316
8.319.2.1 hwVer . . . . .	316
8.320unpack_dms_GetIMSI_t Struct Reference . . . . .	316
8.320.1 Field Documentation . . . . .	316
8.320.1.1 imsi . . . . .	316
8.320.1.2 Tlvresult . . . . .	316
8.321unpack_dms_GetManufacturer_t Struct Reference . . . . .	316
8.321.1 Detailed Description . . . . .	317



8.321.2 Field Documentation . . . . .	317
8.321.2.1 manufacturer . . . . .	317
8.321.2.2 Tlvresult . . . . .	317
8.322unpack_dms_GetModelID_t Struct Reference . . . . .	317
8.322.1 Detailed Description . . . . .	317
8.322.2 Field Documentation . . . . .	317
8.322.2.1 modelid . . . . .	317
8.322.2.2 Tlvresult . . . . .	317
8.323unpack_dms_GetNetworkTime_t Struct Reference . . . . .	317
8.323.1 Detailed Description . . . . .	318
8.323.2 Field Documentation . . . . .	318
8.323.2.1 source . . . . .	318
8.323.2.2 timestamp . . . . .	318
8.323.2.3 Tlvresult . . . . .	318
8.324unpack_dms_GetOfflineReason_t Struct Reference . . . . .	318
8.324.1 Detailed Description . . . . .	318
8.324.2 Field Documentation . . . . .	319
8.324.2.1 pbPlatform . . . . .	319
8.324.2.2 pReasonMask . . . . .	319
8.324.2.3 Tlvresult . . . . .	319
8.325unpack_dms_GetPower_t Struct Reference . . . . .	319
8.325.1 Detailed Description . . . . .	319
8.325.2 Field Documentation . . . . .	319
8.325.2.1 HardwareControlledMode . . . . .	320
8.325.2.2 OfflineReason . . . . .	320
8.325.2.3 OperationMode . . . . .	320
8.325.2.4 Tlvresult . . . . .	320
8.326unpack_dms_GetPRLVersion_t Struct Reference . . . . .	320
8.326.1 Field Documentation . . . . .	320
8.326.1.1 Tlvresult . . . . .	320

8.326.1.2 u16PRLVersion . . . . .	320
8.326.1.3 u8PRLPreference . . . . .	320
8.327unpack_dms_GetSerialNumbers_t Struct Reference . . . . .	320
8.327.1 Detailed Description . . . . .	320
8.327.2 Field Documentation . . . . .	320
8.327.2.1 esn . . . . .	320
8.327.2.2 imei_no . . . . .	321
8.327.2.3 imeisv_svn . . . . .	321
8.327.2.4 meid . . . . .	321
8.328unpack_dms_GetUSBComp_t Struct Reference . . . . .	321
8.328.1 Field Documentation . . . . .	321
8.328.1.1 NumSupUSBComps . . . . .	321
8.328.1.2 SupUSBComps . . . . .	321
8.328.1.3 Tlvresult . . . . .	321
8.328.1.4 USBComp . . . . .	321
8.329unpack_dms_GetVoiceNumber_t Struct Reference . . . . .	321
8.329.1 Field Documentation . . . . .	321
8.329.1.1 MIN . . . . .	321
8.329.1.2 minSize . . . . .	321
8.329.1.3 Tlvresult . . . . .	321
8.329.1.4 VoiceNumber . . . . .	321
8.329.1.5 voiceNumberSize . . . . .	321
8.330unpack_dms_ResetToFactoryDefaults_t Struct Reference . . . . .	321
8.330.1 Detailed Description . . . . .	322
8.330.2 Field Documentation . . . . .	322
8.330.2.1 Tlvresult . . . . .	322
8.331unpack_dms_SetActivationStatusCallback_t Struct Reference . . . . .	322
8.331.1 Detailed Description . . . . .	322
8.331.2 Field Documentation . . . . .	322
8.331.2.1 Tlvresult . . . . .	322

8.332unpack_dms_SetCrashAction_t Struct Reference	322
8.332.1 Detailed Description	322
8.332.2 Field Documentation	323
8.332.2.1 notused	323
8.333unpack_dms_SetCustFeature_t Struct Reference	323
8.333.1 Field Documentation	323
8.333.1.1 Tlvresult	323
8.334unpack_dms_SetCustFeaturesV2_t Struct Reference	323
8.334.1 Detailed Description	323
8.334.2 Field Documentation	323
8.334.2.1 Tlvresult	323
8.335unpack_dms_SetEventReport_ind_t Struct Reference	323
8.335.1 Detailed Description	324
8.335.2 Field Documentation	324
8.335.2.1 ActivationStatusTlv	324
8.335.2.2 OperatingModeTlv	324
8.335.2.3 Tlvresult	324
8.336unpack_dms_SetEventReport_t Struct Reference	324
8.336.1 Field Documentation	324
8.336.1.1 Tlvresult	324
8.337unpack_dms_SetFirmwarePreference_t Struct Reference	324
8.337.1 Field Documentation	324
8.337.1.1 Tlvresult	324
8.338unpack_dms_SetPower_t Struct Reference	324
8.338.1 Field Documentation	325
8.338.1.1 Tlvresult	325
8.339unpack_dms_SetUSBComp_t Struct Reference	325
8.339.1 Field Documentation	325
8.339.1.1 Tlvresult	325
8.340unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference	325

8.340.1 Detailed Description . . . . .	325
8.340.2 Field Documentation . . . . .	325
8.340.2.1 source . . . . .	325
8.340.2.2 Tlvresult . . . . .	325
8.340.2.3 type . . . . .	325
8.341unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . . .	325
8.341.1 Detailed Description . . . . .	326
8.341.2 Field Documentation . . . . .	326
8.341.2.1 source . . . . .	326
8.341.2.2 Tlvresult . . . . .	326
8.341.2.3 type . . . . .	326
8.342unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . . .	326
8.342.1 Detailed Description . . . . .	326
8.342.2 Field Documentation . . . . .	326
8.342.2.1 Tlvresult . . . . .	326
8.343unpack_dms_SLQSGetBandCapability_t Struct Reference . . . . .	326
8.343.1 Detailed Description . . . . .	327
8.343.2 Field Documentation . . . . .	329
8.343.2.1 bandCapability . . . . .	329
8.343.2.2 is_LteBandCapability_Available . . . . .	329
8.343.2.3 is_TdsBandCapability_Available . . . . .	329
8.343.2.4 LteBandCapability . . . . .	329
8.343.2.5 TdsBandCapability . . . . .	329
8.344unpack_dms_SLQSGetERIFile_t Struct Reference . . . . .	329
8.344.1 Detailed Description . . . . .	329
8.344.2 Field Documentation . . . . .	329
8.344.2.1 eriFile . . . . .	329
8.344.2.2 Tlvresult . . . . .	329
8.345unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference . . . . .	330
8.345.1 Detailed Description . . . . .	330

8.345.2 Field Documentation . . . . .	330
8.345.2.1 Tlvresult . . . . .	330
8.346unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . . .	330
8.346.1 Detailed Description . . . . .	330
8.346.2 Field Documentation . . . . .	330
8.346.2.1 crashInfoParam . . . . .	330
8.346.2.2 Tlvresult . . . . .	330
8.347unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference . . . . .	331
8.347.1 Detailed Description . . . . .	331
8.347.2 Field Documentation . . . . .	331
8.347.2.1 pGetDyingGaspCfg . . . . .	331
8.347.2.2 Tlvresult . . . . .	331
8.348unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference . . . . .	331
8.348.1 Detailed Description . . . . .	331
8.348.2 Field Documentation . . . . .	331
8.348.2.1 pGetDyingGaspStatistics . . . . .	331
8.348.2.2 Tlvresult . . . . .	332
8.349unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference . . . . .	332
8.349.1 Detailed Description . . . . .	332
8.349.2 Field Documentation . . . . .	332
8.349.2.1 carrier . . . . .	332
8.349.2.2 fwvers . . . . .	332
8.349.2.3 numEntries . . . . .	332
8.349.2.4 pCurrImgInfo . . . . .	332
8.349.2.5 pkgver . . . . .	332
8.349.2.6 priver . . . . .	333
8.350unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference . . . . .	333
8.350.1 Detailed Description . . . . .	333
8.350.2 Field Documentation . . . . .	334
8.350.2.1 imgType . . . . .	334

8.350.2.2 logString . . . . .	334
8.350.2.3 refData . . . . .	334
8.350.2.4 refString . . . . .	334
8.350.2.5 ResCode . . . . .	334
8.350.2.6 Tlvresult . . . . .	334
8.351unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference . . . . .	334
8.351.1 Detailed Description . . . . .	334
8.351.2 Field Documentation . . . . .	335
8.351.2.1 manString . . . . .	335
8.351.2.2 modelString . . . . .	335
8.351.2.3 plasmaIDString . . . . .	335
8.351.2.4 swVerString . . . . .	335
8.351.2.5 Tlvresult . . . . .	335
8.352unpack_dms_SLQSSwiGetOSInfo_t Struct Reference . . . . .	335
8.352.1 Detailed Description . . . . .	335
8.352.2 Field Documentation . . . . .	335
8.352.2.1 nameString . . . . .	335
8.352.2.2 Tlvresult . . . . .	335
8.352.2.3 versionString . . . . .	335
8.353unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference . . . . .	335
8.353.1 Detailed Description . . . . .	336
8.353.2 Field Documentation . . . . .	336
8.353.2.1 meidString . . . . .	336
8.353.2.2 Tlvresult . . . . .	336
8.354unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . . .	336
8.354.1 Detailed Description . . . . .	336
8.354.2 Field Documentation . . . . .	336
8.354.2.1 Tlvresult . . . . .	336
8.355unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . . .	336
8.355.1 Detailed Description . . . . .	337

8.355.2 Field Documentation . . . . .	337
8.355.2.1 Tlvresult . . . . .	337
8.356unpack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . . .	337
8.356.1 Detailed Description . . . . .	337
8.356.2 Field Documentation . . . . .	337
8.356.2.1 Tlvresult . . . . .	337
8.357unpack_dms_SLQSUIMGetState_t Struct Reference . . . . .	337
8.357.1 Detailed Description . . . . .	338
8.357.2 Field Documentation . . . . .	338
8.357.2.1 state . . . . .	338
8.357.2.2 Tlvresult . . . . .	338
8.358unpack_dms_UIMGetControlKeyStatus_t Struct Reference . . . . .	338
8.358.1 Detailed Description . . . . .	338
8.358.2 Field Documentation . . . . .	339
8.358.2.1 facilityState . . . . .	339
8.358.2.2 Tlvresult . . . . .	339
8.358.2.3 unblockRetriesLeft . . . . .	339
8.358.2.4 verifyRetriesLeft . . . . .	339
8.359unpack_dms_UIMGetICCID_t Struct Reference . . . . .	339
8.359.1 Detailed Description . . . . .	339
8.359.2 Field Documentation . . . . .	339
8.359.2.1 String . . . . .	339
8.359.2.2 stringSize . . . . .	339
8.359.2.3 Tlvresult . . . . .	339
8.360unpack_dms_UIMGetPINStatus_t Struct Reference . . . . .	339
8.360.1 Detailed Description . . . . .	340
8.360.2 Field Documentation . . . . .	341
8.360.2.1 p1Status . . . . .	341
8.360.2.2 p1UnblockRetriesLeft . . . . .	341
8.360.2.3 p1VerifyRetriesLeft . . . . .	341

8.360.2.4 p2Status . . . . .	341
8.360.2.5 p2UnblockRetriesLeft . . . . .	341
8.360.2.6 p2VerifyRetriesLeft . . . . .	341
8.360.2.7 Tlvresult . . . . .	341
8.361unpack_dms_UIMSetControlKeyProtection_t Struct Reference . . . . .	341
8.361.1 Detailed Description . . . . .	341
8.361.2 Field Documentation . . . . .	342
8.361.2.1 Tlvresult . . . . .	342
8.361.2.2 verifyRetriesLeft . . . . .	342
8.362unpack_dms_UIMSetPINProtection_t Struct Reference . . . . .	342
8.362.1 Detailed Description . . . . .	342
8.362.2 Field Documentation . . . . .	342
8.362.2.1 Tlvresult . . . . .	342
8.362.2.2 unblockRetriesLeft . . . . .	342
8.362.2.3 verifyRetriesLeft . . . . .	342
8.363unpack_dms_UIMUnblockControlKey_t Struct Reference . . . . .	342
8.363.1 Detailed Description . . . . .	343
8.363.2 Field Documentation . . . . .	343
8.363.2.1 Tlvresult . . . . .	343
8.363.2.2 unblockRetriesLeft . . . . .	343
8.364unpack_fms_GetImagesPreference_t Struct Reference . . . . .	343
8.364.1 Detailed Description . . . . .	343
8.364.2 Field Documentation . . . . .	343
8.364.2.1 ImageListSize . . . . .	344
8.364.2.2 pImageList . . . . .	344
8.364.2.3 Tlvresult . . . . .	344
8.365unpack_fms_GetStoredImages_t Struct Reference . . . . .	344
8.365.1 Detailed Description . . . . .	344
8.365.2 Field Documentation . . . . .	344
8.365.2.1 imageList . . . . .	344



8.365.2.2 imagelistSize . . . . .	344
8.365.2.3 Tlvresult . . . . .	344
8.366unpack_fms_SetImagesPreference_t Struct Reference . . . . .	344
8.366.1 Detailed Description . . . . .	345
8.366.2 Field Documentation . . . . .	345
8.366.2.1 ImageTypes . . . . .	345
8.366.2.2 ImageTypesSize . . . . .	345
8.366.2.3 Tlvresult . . . . .	345
8.367unpack_loc_BestAvailPos_Ind_t Struct Reference . . . . .	345
8.367.1 Detailed Description . . . . .	346
8.367.2 Field Documentation . . . . .	349
8.367.2.1 pAltitudeWrtEllipsoid . . . . .	349
8.367.2.2 pAltitudeWrtMeanSeaLevel . . . . .	350
8.367.2.3 pGpsTime . . . . .	350
8.367.2.4 pHeading . . . . .	350
8.367.2.5 pHeadingUnc . . . . .	350
8.367.2.6 pHorCirConf . . . . .	350
8.367.2.7 pHorEllpConf . . . . .	350
8.367.2.8 pHorReliability . . . . .	350
8.367.2.9 pHorUncCircular . . . . .	350
8.367.2.10pHorUncEllipseOrientAzimuth . . . . .	350
8.367.2.11pHorUncEllipseSemiMajor . . . . .	350
8.367.2.12pHorUncEllipseSemiMinor . . . . .	350
8.367.2.13pLatitude . . . . .	350
8.367.2.14pLongitude . . . . .	350
8.367.2.15pMagneticDeviation . . . . .	350
8.367.2.16pPrecisionDilution . . . . .	350
8.367.2.17pSensorDataUsage . . . . .	350
8.367.2.18pSpeedHorizontal . . . . .	350
8.367.2.19pSpeedUnc . . . . .	350

8.367.2.20pSpeedVertical . . . . .	350
8.367.2.21pSpeedVerticalUnc . . . . .	350
8.367.2.22pSvUsedforFix . . . . .	350
8.367.2.23pTechnologyMask . . . . .	350
8.367.2.24pTimeSrc . . . . .	350
8.367.2.25pTimestampUtc . . . . .	351
8.367.2.26pTimeUnc . . . . .	351
8.367.2.27pVertConfidence . . . . .	351
8.367.2.28pVertReliability . . . . .	351
8.367.2.29pVertUnc . . . . .	351
8.367.2.30pXid . . . . .	351
8.367.2.31status . . . . .	351
8.367.2.32Tlvresult . . . . .	351
8.368unpack_loc_Delete_Assist_Data_t Struct Reference . . . . .	351
8.368.1 Detailed Description . . . . .	351
8.368.2 Field Documentation . . . . .	351
8.368.2.1 Tlvresult . . . . .	351
8.369unpack_loc_DeleteAssistData_Ind_t Struct Reference . . . . .	351
8.369.1 Detailed Description . . . . .	352
8.369.2 Field Documentation . . . . .	352
8.369.2.1 status . . . . .	352
8.369.2.2 Tlvresult . . . . .	352
8.370unpack_loc_EngineState_Ind_t Struct Reference . . . . .	352
8.370.1 Detailed Description . . . . .	352
8.370.2 Field Documentation . . . . .	353
8.370.2.1 engineState . . . . .	353
8.370.2.2 Tlvresult . . . . .	353
8.371unpack_loc_EventRegister_t Struct Reference . . . . .	353
8.371.1 Detailed Description . . . . .	353
8.371.2 Field Documentation . . . . .	353

8.371.2.1 Tlvresult . . . . .	353
8.372unpack_loc_GnssSvInfo_Ind_t Struct Reference . . . . .	353
8.372.1 Detailed Description . . . . .	354
8.372.2 Field Documentation . . . . .	354
8.372.2.1 altitudeAssumed . . . . .	354
8.372.2.2 pSatelliteInfo . . . . .	354
8.372.2.3 Tlvresult . . . . .	354
8.373unpack_loc_PositionRpt_Ind_t Struct Reference . . . . .	354
8.373.1 Detailed Description . . . . .	355
8.373.2 Field Documentation . . . . .	359
8.373.2.1 pAltitudeAssumed . . . . .	359
8.373.2.2 pAltitudeWrtEllipsoid . . . . .	359
8.373.2.3 pAltitudeWrtMeanSeaLevel . . . . .	359
8.373.2.4 pFixId . . . . .	359
8.373.2.5 pGpsTime . . . . .	359
8.373.2.6 pHeading . . . . .	359
8.373.2.7 pHeadingUnc . . . . .	359
8.373.2.8 pHorConfidence . . . . .	359
8.373.2.9 pHorReliability . . . . .	359
8.373.2.10pHorUncCircular . . . . .	359
8.373.2.11pHorUncEllipseOrientAzimuth . . . . .	359
8.373.2.12pHorUncEllipseSemiMajor . . . . .	359
8.373.2.13pHorUncEllipseSemiMinor . . . . .	359
8.373.2.14pLatitude . . . . .	359
8.373.2.15pLeapSeconds . . . . .	359
8.373.2.16pLongitude . . . . .	359
8.373.2.17pMagneticDeviation . . . . .	359
8.373.2.18pPrecisionDilution . . . . .	359
8.373.2.19pSensorDataUsage . . . . .	359
8.373.2.20pSpeedHorizontal . . . . .	360

8.373.2.21pSpeedUnc . . . . .	360
8.373.2.22pSpeedVertical . . . . .	360
8.373.2.23pSvUsedforFix . . . . .	360
8.373.2.24pTechnologyMask . . . . .	360
8.373.2.25pTimeSrc . . . . .	360
8.373.2.26pTimestampUtc . . . . .	360
8.373.2.27pTimeUnc . . . . .	360
8.373.2.28pVertConfidence . . . . .	360
8.373.2.29pVertReliability . . . . .	360
8.373.2.30pVertUnc . . . . .	360
8.373.2.31sessionId . . . . .	360
8.373.2.32sessionStatus . . . . .	360
8.373.2.33Tlvresult . . . . .	360
8.374unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . . .	360
8.374.1 Detailed Description . . . . .	361
8.374.2 Field Documentation . . . . .	361
8.374.2.1 status . . . . .	361
8.374.2.2 Tlvresult . . . . .	361
8.375unpack_loc_SetExtPowerState_t Struct Reference . . . . .	361
8.375.1 Detailed Description . . . . .	361
8.375.2 Field Documentation . . . . .	362
8.375.2.1 Tlvresult . . . . .	362
8.376unpack_loc_SetOperationMode_Ind_t Struct Reference . . . . .	362
8.376.1 Detailed Description . . . . .	362
8.376.2 Field Documentation . . . . .	362
8.376.2.1 status . . . . .	362
8.376.2.2 Tlvresult . . . . .	362
8.377unpack_loc_SetOperationMode_t Struct Reference . . . . .	362
8.377.1 Detailed Description . . . . .	363
8.377.2 Field Documentation . . . . .	363

8.377.2.1 Tlvresult . . . . .	363
8.378unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . . .	363
8.378.1 Detailed Description . . . . .	363
8.378.2 Field Documentation . . . . .	363
8.378.2.1 Tlvresult . . . . .	363
8.379unpack_loc_Start_t Struct Reference . . . . .	363
8.379.1 Detailed Description . . . . .	363
8.379.2 Field Documentation . . . . .	364
8.379.2.1 Tlvresult . . . . .	364
8.380unpack_loc_Stop_t Struct Reference . . . . .	364
8.380.1 Detailed Description . . . . .	364
8.380.2 Field Documentation . . . . .	364
8.380.2.1 Tlvresult . . . . .	364
8.381unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . . .	364
8.381.1 Detailed Description . . . . .	365
8.381.2 Field Documentation . . . . .	365
8.381.2.1 Application . . . . .	365
8.381.2.2 Broadcast . . . . .	365
8.381.2.3 CustomSCP . . . . .	365
8.381.2.4 ForceRev0 . . . . .	365
8.381.2.5 Protocol . . . . .	365
8.381.2.6 RegForeignNID . . . . .	365
8.381.2.7 RegForeignSID . . . . .	365
8.381.2.8 RegHomeSID . . . . .	365
8.381.2.9 Roaming . . . . .	365
8.381.2.10SCI . . . . .	365
8.381.2.11SCM . . . . .	365
8.382unpack_nas_GetHomeNetwork_t Struct Reference . . . . .	365
8.382.1 Detailed Description . . . . .	365
8.382.2 Field Documentation . . . . .	366

8.382.2.1 mcc . . . . .	366
8.382.2.2 mnc . . . . .	366
8.382.2.3 name . . . . .	366
8.382.2.4 nid . . . . .	366
8.382.2.5 sid . . . . .	366
8.383unpack_nas_GetNetworkPreference_t Struct Reference . . . . .	366
8.383.1 Detailed Description . . . . .	366
8.383.2 Field Documentation . . . . .	367
8.383.2.1 ActiveTechPref . . . . .	367
8.383.2.2 Duration . . . . .	367
8.383.2.3 PersistentTechPref . . . . .	367
8.383.2.4 Tlvresult . . . . .	367
8.384unpack_nas_GetRFInfo_t Struct Reference . . . . .	367
8.384.1 Detailed Description . . . . .	367
8.384.2 Field Documentation . . . . .	367
8.384.2.1 instancesSize . . . . .	367
8.384.2.2 RFBandInfoElements . . . . .	368
8.385unpack_nas_GetServingNetwork_t Struct Reference . . . . .	368
8.385.1 Detailed Description . . . . .	368
8.385.2 Field Documentation . . . . .	368
8.385.2.1 CSDomain . . . . .	368
8.385.2.2 DataCaps . . . . .	368
8.385.2.3 DataCapsLen . . . . .	368
8.385.2.4 MCC . . . . .	368
8.385.2.5 MNC . . . . .	369
8.385.2.6 Name . . . . .	369
8.385.2.7 nameSize . . . . .	369
8.385.2.8 PSDomain . . . . .	369
8.385.2.9 Radiolfaces . . . . .	369
8.385.2.10RadiolfacesSize . . . . .	369

8.385.2.11	RAN	369
8.385.2.12	RegistrationState	369
8.385.2.13	Roaming	369
8.386	unpack_nas_GetServingNetworkCapabilities_t Struct Reference	369
8.386.1	Detailed Description	369
8.386.2	Field Documentation	369
8.386.2.1	DataCaps	369
8.386.2.2	DataCapsLen	369
8.387	unpack_nas_GetSignalStrengths_t Struct Reference	369
8.387.1	Detailed Description	370
8.387.2	Field Documentation	370
8.387.2.1	len	370
8.387.2.2	radio	370
8.387.2.3	rsqi	370
8.388	unpack_nas_PerformNetworkScan_t Struct Reference	370
8.388.1	Detailed Description	370
8.388.2	Field Documentation	370
8.388.2.1	p3GppNetworkInfoInstances	370
8.388.2.2	p3GppNetworkInstanceSize	370
8.388.2.3	pPCSIInstance	370
8.388.2.4	pPCSIInstanceSize	370
8.388.2.5	pRATINInstance	370
8.388.2.6	pRATInstanceSize	371
8.388.2.7	pScanResult	371
8.389	unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	371
8.389.1	Detailed Description	371
8.389.2	Field Documentation	371
8.389.2.1	dataCaps	371
8.389.2.2	dataCapsSize	371
8.390	unpack_nas_SetEventReportInd_t Struct Reference	371

8.390.1 Detailed Description . . . . .	371
8.390.2 Field Documentation . . . . .	371
8.390.2.1 RFTlv . . . . .	371
8.390.2.2 RRTlv . . . . .	372
8.390.2.3 SLQSSSTlv . . . . .	372
8.390.2.4 SSTlv . . . . .	372
8.391unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . . .	372
8.391.1 Detailed Description . . . . .	372
8.391.2 Field Documentation . . . . .	372
8.391.2.1 sPhyCaAggPcellInfo . . . . .	372
8.391.2.2 sPhyCaAggScellIDBw . . . . .	372
8.391.2.3 sPhyCaAggScellIndex . . . . .	372
8.391.2.4 sPhyCaAggScellIndType . . . . .	372
8.391.2.5 sPhyCaAggScellInfo . . . . .	373
8.392unpack_nas_SetNetworkPreference_t Struct Reference . . . . .	373
8.392.1 Detailed Description . . . . .	373
8.392.2 Field Documentation . . . . .	373
8.392.2.1 Tlvresult . . . . .	373
8.393unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . . .	373
8.393.1 Detailed Description . . . . .	374
8.393.2 Field Documentation . . . . .	374
8.393.2.1 roaming . . . . .	374
8.394unpack_nas_SetServingSystemCallback_ind_t Struct Reference . . . . .	374
8.394.1 Detailed Description . . . . .	374
8.394.2 Field Documentation . . . . .	374
8.394.2.1 SSInfo . . . . .	374
8.394.2.2 Tlvresult . . . . .	374
8.395unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . . .	374
8.395.1 Detailed Description . . . . .	375
8.395.2 Field Documentation . . . . .	375



8.395.2.1 LTECphyCalInfo . . . . .	375
8.395.2.2 Tlvresult . . . . .	375
8.396unpack_nas_SLQSGetNetworkTime_t Struct Reference . . . . .	375
8.396.1 Detailed Description . . . . .	375
8.396.2 Field Documentation . . . . .	375
8.396.2.1 p3GPP2TimeInfo . . . . .	375
8.396.2.2 p3GPPTimeInfo . . . . .	376
8.397unpack_nas_SLQSGetPLMNName_t Struct Reference . . . . .	376
8.397.1 Field Documentation . . . . .	376
8.397.1.1 longName . . . . .	376
8.397.1.2 longNameCI . . . . .	376
8.397.1.3 longNameEn . . . . .	376
8.397.1.4 longNameLen . . . . .	376
8.397.1.5 longNameSB . . . . .	376
8.397.1.6 shortName . . . . .	376
8.397.1.7 shortNameCI . . . . .	376
8.397.1.8 shortNameEn . . . . .	376
8.397.1.9 shortNameLen . . . . .	376
8.397.1.10shortNameSB . . . . .	376
8.397.1.11spn . . . . .	376
8.397.1.12spnEncoding . . . . .	376
8.397.1.13spnLength . . . . .	376
8.398unpack_nas_SLQSGetServingSystem_t Struct Reference . . . . .	376
8.398.1 Detailed Description . . . . .	377
8.398.2 Field Documentation . . . . .	378
8.398.2.1 BasestationID . . . . .	378
8.398.2.2 BasestationLatitude . . . . .	378
8.398.2.3 BasestationLongitude . . . . .	378
8.398.2.4 CallBarStatus . . . . .	378
8.398.2.5 CDMA_P_Rev . . . . .	378

8.398.2.6 CDMASystemInfoExt . . . . .	378
8.398.2.7 CellID . . . . .	378
8.398.2.8 ConcSvcInfo . . . . .	378
8.398.2.9 CurrentPLMN . . . . .	378
8.398.2.10DataSrvCapabilities . . . . .	378
8.398.2.11DefaultRoamInd . . . . .	378
8.398.2.12DetailedSvcInfo . . . . .	378
8.398.2.13DTMInd . . . . .	378
8.398.2.14Gpp2TimeZone . . . . .	378
8.398.2.15GppNetworkDSTAdjustment . . . . .	378
8.398.2.16GppTimeZone . . . . .	378
8.398.2.17HdrPersonality . . . . .	378
8.398.2.18Lac . . . . .	378
8.398.2.19NetworkID . . . . .	379
8.398.2.20PRLInd . . . . .	379
8.398.2.21RoamIndicatorVal . . . . .	379
8.398.2.22RoamingIndicatorList . . . . .	379
8.398.2.23ServingSystem . . . . .	379
8.398.2.24SystemID . . . . .	379
8.398.2.25TrackAreaCode . . . . .	379
8.399unpack_nas_SLQSGetSignalStrength_t Struct Reference . . . . .	379
8.399.1 Detailed Description . . . . .	379
8.399.2 Field Documentation . . . . .	380
8.399.2.1 ecioList . . . . .	380
8.399.2.2 ecioListLen . . . . .	380
8.399.2.3 errorRateList . . . . .	380
8.399.2.4 errorRateListLen . . . . .	380
8.399.2.5 lo . . . . .	380
8.399.2.6 ltersrp . . . . .	380
8.399.2.7 ltesnr . . . . .	380

8.399.2.8 rsrqInfo . . . . .	380
8.399.2.9 rxSignalStrengthList . . . . .	380
8.399.2.10rxSignalStrengthListLen . . . . .	380
8.399.2.11signalStrengthReqMask . . . . .	380
8.399.2.12sinr . . . . .	380
8.400unpack_nas_SLQSGetSysInfo_t Struct Reference . . . . .	380
8.400.1 Detailed Description . . . . .	380
8.400.2 Field Documentation . . . . .	382
8.400.2.1 pAddCDMASysInfo . . . . .	382
8.400.2.2 pAddGSMSysInfo . . . . .	382
8.400.2.3 pAddHDRSysInfo . . . . .	382
8.400.2.4 pAddLTESysInfo . . . . .	382
8.400.2.5 pAddWCDMASysInfo . . . . .	382
8.400.2.6 pCDMASrvStatusInfo . . . . .	382
8.400.2.7 pCDMASysInfo . . . . .	382
8.400.2.8 pGSMCallBarringSysInfo . . . . .	382
8.400.2.9 pGSMCipherDomainSysInfo . . . . .	382
8.400.2.10pGSMSrvStatusInfo . . . . .	382
8.400.2.11pGSMSysInfo . . . . .	382
8.400.2.12pHDRSrvStatusInfo . . . . .	383
8.400.2.13pHDRSysInfo . . . . .	383
8.400.2.14pLTESrvStatusInfo . . . . .	383
8.400.2.15pLTESysInfo . . . . .	383
8.400.2.16pLTEVoiceSupportSysInfo . . . . .	383
8.400.2.17pWCDMACallBarringSysInfo . . . . .	383
8.400.2.18pWCDMACipherDomainSysInfo . . . . .	383
8.400.2.19pWCDMASrvStatusInfo . . . . .	383
8.400.2.20pWCDMASysInfo . . . . .	383
8.401unpack_nas_SLQSGetSysSelectionPref_t Struct Reference . . . . .	383
8.401.1 Detailed Description . . . . .	383

8.401.2 Field Documentation	386
8.401.2.1 pBandPref	386
8.401.2.2 pEmerMode	386
8.401.2.3 pGWAcqOrderPref	386
8.401.2.4 pLTETBandPref	386
8.401.2.5 pModePref	386
8.401.2.6 pNetSelPref	386
8.401.2.7 pPRLPref	386
8.401.2.8 pRoamPref	386
8.401.2.9 pSrvDomainPref	386
8.402unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference	386
8.402.1 Detailed Description	387
8.402.2 Field Documentation	387
8.402.2.1 pCDMAInfo	387
8.402.2.2 pGERANInfo	387
8.402.2.3 pLTEInfoInterfreq	388
8.402.2.4 pLTEInfoIntrafreq	388
8.402.2.5 pLTEInfoNeighboringGSM	388
8.402.2.6 pLTEInfoNeighboringWCDMA	388
8.402.2.7 pUMTSCellID	388
8.402.2.8 pUMTSInfo	388
8.402.2.9 pWCDMAInfoLTENeighborCell	388
8.403unpack_nas_SLQSNasGetSigInfo_t Struct Reference	388
8.403.1 Detailed Description	388
8.403.2 Field Documentation	388
8.403.2.1 CDMASSInfo	388
8.403.2.2 GSMSSInfo	388
8.403.2.3 HDRSSInfo	388
8.403.2.4 LTESInfo	388
8.403.2.5 WCDMASSInfo	388

8.404unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . . .	389
8.404.1 Detailed Description . . . . .	389
8.404.2 Field Documentation . . . . .	389
8.404.2.1 pDayltSavAdj . . . . .	389
8.404.2.2 pRadioInterface . . . . .	389
8.404.2.3 pTimeZone . . . . .	389
8.404.2.4 universalTime . . . . .	389
8.405unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . . .	390
8.405.1 Detailed Description . . . . .	390
8.405.2 Field Documentation . . . . .	390
8.405.2.1 pCDMASigInfo . . . . .	390
8.405.2.2 pGSMSigInfo . . . . .	390
8.405.2.3 pHDRSigInfo . . . . .	390
8.405.2.4 pLTESigInfo . . . . .	390
8.405.2.5 pRscp . . . . .	390
8.405.2.6 pTDSCDMASigInfoExt . . . . .	390
8.405.2.7 pWCDMASigInfo . . . . .	390
8.406unpack_nas_SLQSNasSwiModemStatus_t Struct Reference . . . . .	390
8.406.1 Detailed Description . . . . .	391
8.406.2 Field Documentation . . . . .	391
8.406.2.1 commonInfo . . . . .	391
8.406.2.2 pLTEInfo . . . . .	391
8.407unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . . .	391
8.407.1 Detailed Description . . . . .	391
8.407.2 Field Documentation . . . . .	391
8.407.2.1 Info . . . . .	391
8.407.2.2 Tlvresult . . . . .	391
8.408unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . . .	391
8.408.1 Detailed Description . . . . .	392
8.408.2 Field Documentation . . . . .	392

8.408.2.1 t3396_apn . . . . .	392
8.408.2.2 t3396_plmn_id . . . . .	392
8.408.2.3 t3396_val . . . . .	392
8.409unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . . .	392
8.409.1 Detailed Description . . . . .	392
8.409.2 Field Documentation . . . . .	393
8.409.2.1 Info . . . . .	393
8.409.2.2 Tlvresult . . . . .	393
8.410unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . . .	393
8.410.1 Detailed Description . . . . .	393
8.410.2 Field Documentation . . . . .	393
8.410.2.1 CQIValueCW0 . . . . .	393
8.410.2.2 CQIValueCW1 . . . . .	393
8.410.2.3 ValidityCW0 . . . . .	393
8.410.2.4 ValidityCW1 . . . . .	393
8.411unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference . . . . .	394
8.411.1 Detailed Description . . . . .	394
8.411.2 Field Documentation . . . . .	394
8.411.2.1 pSccRxInfo . . . . .	394
8.412unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . . .	394
8.412.1 Detailed Description . . . . .	395
8.412.2 Field Documentation . . . . .	396
8.412.2.1 pAddCDMASysInfo . . . . .	396
8.412.2.2 pAddGSMSysInfo . . . . .	396
8.412.2.3 pAddHDRSysInfo . . . . .	396
8.412.2.4 pAddLTESysInfo . . . . .	396
8.412.2.5 pAddWCDMASysInfo . . . . .	396
8.412.2.6 pCDMASrvStatusInfo . . . . .	396
8.412.2.7 pCDMASysInfo . . . . .	396
8.412.2.8 pGSMCallBarringSysInfo . . . . .	396

8.412.2.9 pGSMCipherDomainSysInfo . . . . .	396
8.412.2.10pGSMSrvStatusInfo . . . . .	397
8.412.2.11pGSMSysInfo . . . . .	397
8.412.2.12pHDRSrvStatusInfo . . . . .	397
8.412.2.13pHDRSysInfo . . . . .	397
8.412.2.14pLTERsrvStatusInfo . . . . .	397
8.412.2.15pLTERsysInfo . . . . .	397
8.412.2.16pLTEVoiceSupportSysInfo . . . . .	397
8.412.2.17pSysInfoNoChange . . . . .	397
8.412.2.18pWCDMACallBarringSysInfo . . . . .	397
8.412.2.19pWCDMACipherDomainSysInfo . . . . .	397
8.412.2.20pWCDMASrvStatusInfo . . . . .	397
8.412.2.21pWCDMASysInfo . . . . .	397
8.413unpack_omaDmConfigTlv_t Struct Reference . . . . .	397
8.413.1 Detailed Description . . . . .	397
8.413.2 Field Documentation . . . . .	398
8.413.2.1 alertmsg . . . . .	398
8.413.2.2 alertmsglength . . . . .	398
8.413.2.3 state . . . . .	398
8.413.2.4 userInputReq . . . . .	398
8.413.2.5 userInputTimeout . . . . .	398
8.414unpack_omaDmFotaTlv_t Struct Reference . . . . .	398
8.414.1 Detailed Description . . . . .	399
8.414.2 Field Documentation . . . . .	400
8.414.2.1 description . . . . .	400
8.414.2.2 descriptionlength . . . . .	400
8.414.2.3 fwdloadsize . . . . .	400
8.414.2.4 fwloadComplete . . . . .	400
8.414.2.5 namelength . . . . .	400
8.414.2.6 package_name . . . . .	400

8.414.2.7 sessionType . . . . .	400
8.414.2.8 severity . . . . .	400
8.414.2.9 state . . . . .	400
8.414.2.10updateCompleteStatus . . . . .	400
8.414.2.11userInputReq . . . . .	400
8.414.2.12userInputTimeout . . . . .	400
8.414.2.13version . . . . .	400
8.414.2.14versionlength . . . . .	400
8.415unpack_omaDmNotificationsTlv_t Struct Reference . . . . .	400
8.415.1 Field Documentation . . . . .	401
8.415.1.1 notification . . . . .	401
8.415.1.2 sessionStatus . . . . .	401
8.416unpack_qmi_t Struct Reference . . . . .	401
8.416.1 Detailed Description . . . . .	401
8.416.2 Field Documentation . . . . .	401
8.416.2.1 msgid . . . . .	401
8.416.2.2 type . . . . .	401
8.416.2.3 xid . . . . .	401
8.417unpack_qos_dataRate_t Struct Reference . . . . .	401
8.417.1 Detailed Description . . . . .	401
8.417.2 Field Documentation . . . . .	402
8.417.2.1 dataRateMax . . . . .	402
8.417.2.2 guaranteedRate . . . . .	402
8.418unpack_qos_IPv4Addr_t Struct Reference . . . . .	402
8.418.1 Detailed Description . . . . .	402
8.418.2 Field Documentation . . . . .	402
8.418.2.1 addr . . . . .	402
8.418.2.2 subnetMask . . . . .	402
8.419unpack_qos_IPv6Addr_t Struct Reference . . . . .	402
8.419.1 Detailed Description . . . . .	403



8.419.2 Field Documentation . . . . .	403
8.419.2.1 addr . . . . .	403
8.419.2.2 prefixLen . . . . .	403
8.420unpack_qos_IPv6TrafCls_t Struct Reference . . . . .	403
8.420.1 Detailed Description . . . . .	403
8.420.2 Field Documentation . . . . .	403
8.420.2.1 mask . . . . .	403
8.420.2.2 val . . . . .	403
8.421unpack_qos_pktErrRate_t Struct Reference . . . . .	404
8.421.1 Detailed Description . . . . .	404
8.421.2 Field Documentation . . . . .	404
8.421.2.1 exponent . . . . .	404
8.421.2.2 multiplier . . . . .	404
8.422unpack_qos_Port_t Struct Reference . . . . .	404
8.422.1 Detailed Description . . . . .	404
8.422.2 Field Documentation . . . . .	404
8.422.2.1 port . . . . .	404
8.422.2.2 range . . . . .	404
8.423unpack_qos_QosFlowInfo_t Struct Reference . . . . .	405
8.423.1 Detailed Description . . . . .	405
8.423.2 Field Documentation . . . . .	406
8.423.2.1 BearerID . . . . .	406
8.423.2.2 is_RxQFlowGranted_Available . . . . .	406
8.423.2.3 is_TxQFlowGranted_Available . . . . .	406
8.423.2.4 NumRxFilters . . . . .	406
8.423.2.5 NumTxFilters . . . . .	406
8.423.2.6 QFlowState . . . . .	406
8.423.2.7 RxQFilter . . . . .	406
8.423.2.8 RxQFlowGranted . . . . .	406
8.423.2.9 TxQFilter . . . . .	406

8.423.2.10TxQFlowGranted . . . . .	406
8.424unpack_qos_QosFlowInfoState_t Struct Reference . . . . .	406
8.424.1 Detailed Description . . . . .	407
8.424.2 Field Documentation . . . . .	407
8.424.2.1 id . . . . .	407
8.424.2.2 isNewFlow . . . . .	407
8.424.2.3 state . . . . .	407
8.425unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . . .	407
8.425.1 Detailed Description . . . . .	407
8.425.2 Field Documentation . . . . .	407
8.425.2.1 NWQoSStatus . . . . .	408
8.426unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . . .	408
8.426.1 Detailed Description . . . . .	408
8.426.2 Field Documentation . . . . .	409
8.426.2.1 ambr_dl . . . . .	409
8.426.2.2 ambr_dl_ext . . . . .	409
8.426.2.3 ambr_dl_ext2 . . . . .	409
8.426.2.4 ambr_ul . . . . .	409
8.426.2.5 ambr_ul_ext . . . . .	409
8.426.2.6 ambr_ul_ext2 . . . . .	409
8.426.2.7 apnId . . . . .	409
8.427unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . . .	409
8.427.1 Detailed Description . . . . .	409
8.427.2 Field Documentation . . . . .	410
8.427.2.1 apnId . . . . .	410
8.427.2.2 numQosFlow . . . . .	410
8.427.2.3 qosFlow . . . . .	410
8.427.2.4 total_rx_bytes . . . . .	410
8.427.2.5 total_rx_pkt . . . . .	410
8.427.2.6 total_tx_bytes . . . . .	410

8.427.2.7 total_tx_bytes_drp . . . . .	410
8.427.2.8 total_tx_pkt . . . . .	410
8.427.2.9 total_tx_pkt_drp . . . . .	410
8.428unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . . .	410
8.428.1 Detailed Description . . . . .	410
8.428.2 Field Documentation . . . . .	411
8.428.2.1 NumFlows . . . . .	411
8.428.2.2 QosFlowInfo . . . . .	411
8.429unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . . .	411
8.429.1 Detailed Description . . . . .	411
8.429.2 Field Documentation . . . . .	411
8.429.2.1 status . . . . .	411
8.430unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . . .	411
8.430.1 Detailed Description . . . . .	412
8.430.2 Field Documentation . . . . .	412
8.430.2.1 event . . . . .	412
8.431unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . . .	412
8.431.1 Detailed Description . . . . .	412
8.431.2 Field Documentation . . . . .	413
8.431.2.1 event . . . . .	413
8.431.2.2 id . . . . .	413
8.431.2.3 reason . . . . .	413
8.431.2.4 status . . . . .	413
8.432unpack_qos_swiQosFilter_t Struct Reference . . . . .	413
8.432.1 Detailed Description . . . . .	414
8.432.2 Field Documentation . . . . .	415
8.432.2.1 EspSpi . . . . .	415
8.432.2.2 Id . . . . .	416
8.432.2.3 index . . . . .	416
8.432.2.4 IPv4DstAddr . . . . .	416

8.432.2.5 IPv4SrcAddr . . . . .	416
8.432.2.6 IPv4Tos . . . . .	416
8.432.2.7 IPv6DstAddr . . . . .	416
8.432.2.8 IPv6Label . . . . .	416
8.432.2.9 IPv6SrcAddr . . . . .	416
8.432.2.10IPv6TrafCls . . . . .	416
8.432.2.11is_EspSpi_Available . . . . .	416
8.432.2.12s_Id_Available . . . . .	416
8.432.2.13s_IPv4DstAddr_Available . . . . .	416
8.432.2.14s_IPv4SrcAddr_Available . . . . .	416
8.432.2.15s_IPv4Tos_Available . . . . .	416
8.432.2.16s_IPv6DstAddr_Available . . . . .	416
8.432.2.17s_IPv6Label_Available . . . . .	416
8.432.2.18s_IPv6SrcAddr_Available . . . . .	416
8.432.2.19s_IPv6TrafCls_Available . . . . .	416
8.432.2.20s_NxtHdrProto_Available . . . . .	416
8.432.2.21is_Precedence_Available . . . . .	416
8.432.2.22s_TCPDstPort_Available . . . . .	416
8.432.2.23s_TCPSrcPort_Available . . . . .	416
8.432.2.24s_TranDstPort_Available . . . . .	416
8.432.2.25s_TranSrcPort_Available . . . . .	417
8.432.2.26s_UDPDstPort_Available . . . . .	417
8.432.2.27s_UDPSrcPort_Available . . . . .	417
8.432.2.28NxtHdrProto . . . . .	417
8.432.2.29Precedence . . . . .	417
8.432.2.30TCPDstPort . . . . .	417
8.432.2.31TCPSrcPort . . . . .	417
8.432.2.32TranDstPort . . . . .	417
8.432.2.33TranSrcPort . . . . .	417
8.432.2.34UDPDstPort . . . . .	417

8.432.2.35UDPSrcPort . . . . .	417
8.432.2.36version . . . . .	417
8.433unpack_qos_swiQosFlow_t Struct Reference . . . . .	417
8.433.1 Detailed Description . . . . .	418
8.433.2 Field Documentation . . . . .	420
8.433.2.1 DataRate . . . . .	420
8.433.2.2 index . . . . .	420
8.433.2.3 is_DataRate_Available . . . . .	420
8.433.2.4 is_Jitter_Available . . . . .	420
8.433.2.5 is_Latency_Available . . . . .	420
8.433.2.6 is_LteQci_Available . . . . .	420
8.433.2.7 is_MaxAllowedPktSz_Available . . . . .	420
8.433.2.8 is_MinPolicedPktSz_Available . . . . .	420
8.433.2.9 is_PktErrRate_Available . . . . .	420
8.433.2.10s_ProfileId3GPP2_Available . . . . .	420
8.433.2.11is_TokenBucket_Available . . . . .	420
8.433.2.12s_TrafficClass_Available . . . . .	420
8.433.2.13s_val_3GPP2Pri_Available . . . . .	420
8.433.2.14s_val_3GPPImCn_Available . . . . .	420
8.433.2.15s_val_3GPPResResidualBER_Available . . . . .	420
8.433.2.16s_val_3GPPSigInd_Available . . . . .	420
8.433.2.17s_val_3GPPTraHdlPri_Available . . . . .	420
8.433.2.18Jitter . . . . .	421
8.433.2.19Latency . . . . .	421
8.433.2.20LteQci . . . . .	421
8.433.2.21MaxAllowedPktSz . . . . .	421
8.433.2.22MinPolicedPktSz . . . . .	421
8.433.2.23PktErrRate . . . . .	421
8.433.2.24ProfileId3GPP2 . . . . .	421
8.433.2.25TokenBucket . . . . .	421

8.433.2.26TrafficClass . . . . .	421
8.433.2.27val_3GPP2Pri . . . . .	421
8.433.2.28val_3GPPImCn . . . . .	421
8.433.2.29val_3GPPResResidualBER . . . . .	421
8.433.2.30val_3GPPSigInd . . . . .	421
8.433.2.31val_3GPPTraHdlPri . . . . .	421
8.434unpack_qos_tokenBucket_t Struct Reference . . . . .	421
8.434.1 Detailed Description . . . . .	421
8.434.2 Field Documentation . . . . .	422
8.434.2.1 bucketSz . . . . .	422
8.434.2.2 peakRate . . . . .	422
8.434.2.3 tokenRate . . . . .	422
8.435unpack_qos_Tos_t Struct Reference . . . . .	422
8.435.1 Detailed Description . . . . .	422
8.435.2 Field Documentation . . . . .	422
8.435.2.1 mask . . . . .	422
8.435.2.2 val . . . . .	422
8.436unpack_QosFlowStat_t Struct Reference . . . . .	422
8.436.1 Detailed Description . . . . .	423
8.436.2 Field Documentation . . . . .	423
8.436.2.1 bearerId . . . . .	423
8.436.2.2 tx_bytes . . . . .	423
8.436.2.3 tx_bytes_drp . . . . .	423
8.436.2.4 tx_pkt . . . . .	423
8.436.2.5 tx_pkt_drp . . . . .	423
8.437unpack_RMTransferStatistics_ind_t Struct Reference . . . . .	423
8.437.1 Detailed Description . . . . .	423
8.437.2 Field Documentation . . . . .	424
8.437.2.1 RxDropConutTlv . . . . .	424
8.437.2.2 RxOkByteCountTlv . . . . .	424

8.437.2.3 RxOkConutTlv . . . . .	424
8.437.2.4 TxDropConutTlv . . . . .	424
8.437.2.5 TxOkByteCountTlv . . . . .	424
8.437.2.6 TxOkConutTlv . . . . .	424
8.438unpack_sms_SendSMS_t Struct Reference . . . . .	424
8.438.1 Detailed Description . . . . .	424
8.438.2 Field Documentation . . . . .	425
8.438.2.1 messageFailureCode . . . . .	425
8.438.2.2 messageID . . . . .	425
8.439unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . . .	425
8.439.1 Detailed Description . . . . .	425
8.439.2 Field Documentation . . . . .	426
8.439.2.1 ETWSPLMNTlv . . . . .	426
8.439.2.2 ETWSTlv . . . . .	426
8.439.2.3 IMSTlv . . . . .	426
8.439.2.4 MMTlv . . . . .	426
8.439.2.5 NewMMTlv . . . . .	426
8.439.2.6 SMSCTlv . . . . .	426
8.439.2.7 TRMessageTlv . . . . .	426
8.440unpack_sms_SetNewSMSCallback_t Struct Reference . . . . .	426
8.441unpack_sms_SLQSDeleteSMS_t Struct Reference . . . . .	426
8.442unpack_sms_SLQSGetSMS_t Struct Reference . . . . .	426
8.442.1 Detailed Description . . . . .	426
8.442.2 Field Documentation . . . . .	427
8.442.2.1 message . . . . .	427
8.442.2.2 messageFormat . . . . .	427
8.442.2.3 messageSize . . . . .	427
8.442.2.4 messageTag . . . . .	427
8.443unpack_sms_SLQSGetSMSList_t Struct Reference . . . . .	427
8.443.1 Detailed Description . . . . .	427

8.443.2 Field Documentation . . . . .	427
8.443.2.1 messageList . . . . .	427
8.443.2.2 messageListSize . . . . .	427
8.444unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . . .	427
8.445unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . . .	428
8.445.1 Detailed Description . . . . .	428
8.445.2 Field Documentation . . . . .	428
8.445.2.1 messageMode . . . . .	428
8.445.2.2 storageType . . . . .	428
8.446unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . . .	428
8.446.1 Detailed Description . . . . .	428
8.446.2 Field Documentation . . . . .	429
8.446.2.1 fix_rate . . . . .	429
8.446.2.2 fix_rate_reported . . . . .	429
8.446.2.3 fix_type . . . . .	430
8.446.2.4 fix_type_reported . . . . .	430
8.446.2.5 function . . . . .	430
8.446.2.6 function_reported . . . . .	430
8.446.2.7 max_dist . . . . .	430
8.446.2.8 max_dist_reported . . . . .	430
8.446.2.9 max_time . . . . .	430
8.446.2.10max_time_reported . . . . .	430
8.447unpack_swioma_SLQSOMADMAAlertCallback_ind_t Struct Reference . . . . .	430
8.447.1 Detailed Description . . . . .	430
8.447.2 Field Documentation . . . . .	431
8.447.2.1 eventType . . . . .	431
8.447.2.2 SessionInfoConfig . . . . .	431
8.447.2.3 SessionInfoFota . . . . .	431
8.447.2.4 SessionInfoNotification . . . . .	431
8.448unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . . .	431



8.448.1 Detailed Description . . . . .	431
8.448.2 Field Documentation . . . . .	433
8.448.2.1 Date . . . . .	433
8.448.2.2 DateLength . . . . .	433
8.448.2.3 PkgDescLength . . . . .	433
8.448.2.4 PkgDescription . . . . .	433
8.448.2.5 PkgName . . . . .	433
8.448.2.6 PkgNameLength . . . . .	433
8.448.2.7 RetryCount . . . . .	433
8.448.2.8 SessionState . . . . .	433
8.448.2.9 SessionType . . . . .	433
8.448.2.10Severity . . . . .	433
8.448.2.11Source . . . . .	433
8.448.2.12SourceLength . . . . .	434
8.448.2.13Status . . . . .	434
8.448.2.14Time . . . . .	434
8.448.2.15TimeLength . . . . .	434
8.448.2.16UpdateCompleteStatus . . . . .	434
8.449unpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . . .	434
8.449.1 Detailed Description . . . . .	434
8.449.2 Field Documentation . . . . .	435
8.449.2.1 Autosdm . . . . .	435
8.449.2.2 FOTAdownload . . . . .	435
8.449.2.3 FOTAUpdate . . . . .	435
8.449.2.4 FwAutoCheck . . . . .	435
8.449.2.5 OMADMEEnabled . . . . .	435
8.450unpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . . .	435
8.450.1 Detailed Description . . . . .	435
8.450.2 Field Documentation . . . . .	436
8.450.2.1 FwAvailability . . . . .	436

8.451unpack_uim_ChangePin_t Struct Reference . . . . .	436
8.451.1 Detailed Description . . . . .	436
8.451.2 Field Documentation . . . . .	436
8.451.2.1 pEncryptedPIN1 . . . . .	436
8.451.2.2 pIndicationToken . . . . .	437
8.451.2.3 pRemainingRetries . . . . .	437
8.451.2.4 Tlvresult . . . . .	437
8.452unpack_uim_GetCardStatus_t Struct Reference . . . . .	437
8.452.1 Detailed Description . . . . .	437
8.452.2 Field Documentation . . . . .	437
8.452.2.1 pCardStatus . . . . .	437
8.452.2.2 pHotSwapStatus . . . . .	437
8.452.2.3 Tlvresult . . . . .	437
8.453unpack_uim_ReadTransparent_t Struct Reference . . . . .	437
8.453.1 Detailed Description . . . . .	438
8.453.2 Field Documentation . . . . .	438
8.453.2.1 pCardResult . . . . .	438
8.453.2.2 pEncryptedData . . . . .	438
8.453.2.3 pIndicationToken . . . . .	438
8.453.2.4 pReadResult . . . . .	438
8.453.2.5 Tlvresult . . . . .	438
8.454unpack_uim_SetPinProtection_t Struct Reference . . . . .	438
8.454.1 Detailed Description . . . . .	438
8.454.2 Field Documentation . . . . .	439
8.454.2.1 pEncryptedPIN1 . . . . .	439
8.454.2.2 pIndicationToken . . . . .	439
8.454.2.3 pRemainingRetries . . . . .	439
8.454.2.4 Tlvresult . . . . .	439
8.455unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . . .	439
8.455.1 Detailed Description . . . . .	439

8.455.2 Field Documentation . . . . .	440
8.455.2.1 bNumberOfPhySlots . . . . .	440
8.455.2.2 slotsstatusChange . . . . .	440
8.456unpack_uim_SLQSUIEventRegister_t Struct Reference . . . . .	440
8.456.1 Detailed Description . . . . .	440
8.456.2 Field Documentation . . . . .	440
8.456.2.1 eventMask . . . . .	440
8.457unpack_uim_SLQSUIGetSlotsStatus_t Struct Reference . . . . .	440
8.457.1 Detailed Description . . . . .	440
8.457.2 Field Documentation . . . . .	441
8.457.2.1 pNumberOfPhySlot . . . . .	441
8.457.2.2 pUimSlotsStatus . . . . .	441
8.458unpack_uim_SLQSUISetStatusChangeCallBack_ind_t Struct Reference . . . . .	441
8.458.1 Detailed Description . . . . .	441
8.458.2 Field Documentation . . . . .	441
8.458.2.1 pCardStatus . . . . .	441
8.459unpack_uim_UnblockPin_t Struct Reference . . . . .	441
8.459.1 Detailed Description . . . . .	441
8.459.2 Field Documentation . . . . .	442
8.459.2.1 pEncryptedPIN1 . . . . .	442
8.459.2.2 pIndicationToken . . . . .	442
8.459.2.3 pRemainingRetries . . . . .	442
8.459.2.4 Tlvresult . . . . .	442
8.460unpack_uim_VerifyPin_t Struct Reference . . . . .	442
8.460.1 Detailed Description . . . . .	442
8.460.2 Field Documentation . . . . .	443
8.460.2.1 pEncryptedPIN1 . . . . .	443
8.460.2.2 pIndicationToken . . . . .	443
8.460.2.3 pRemainingRetries . . . . .	443
8.460.2.4 Tlvresult . . . . .	443

8.461unpack_wds_DHCPv4ClientLease_ind_t Struct Reference . . . . .	443
8.461.1 Field Documentation . . . . .	443
8.461.1.1 DHCPv4LeaseOptTlv . . . . .	443
8.461.1.2 DHCPv4LeaseStateTlv . . . . .	443
8.461.1.3 IPv4AddrTlv . . . . .	443
8.461.1.4 ProfileIdTlv . . . . .	443
8.462unpack_wds_GetAutoconnect_t Struct Reference . . . . .	443
8.462.1 Detailed Description . . . . .	443
8.462.2 Field Documentation . . . . .	444
8.462.2.1 psetting . . . . .	444
8.463unpack_wds_GetByteTotals_t Struct Reference . . . . .	444
8.463.1 Detailed Description . . . . .	444
8.463.2 Field Documentation . . . . .	444
8.463.2.1 pRXTotalBytes . . . . .	444
8.463.2.2 pTXTotalBytes . . . . .	444
8.464unpack_wds_GetConnectionRate_t Struct Reference . . . . .	444
8.464.1 Detailed Description . . . . .	444
8.464.2 Field Documentation . . . . .	445
8.464.2.1 currentChannelRXRate . . . . .	445
8.464.2.2 currentChannelTXRate . . . . .	445
8.464.2.3 maxChannelRXRate . . . . .	445
8.464.2.4 maxChannelTXRate . . . . .	445
8.465unpack_wds_GetDataBearerTechnology_t Struct Reference . . . . .	445
8.465.1 Detailed Description . . . . .	445
8.465.2 Field Documentation . . . . .	446
8.465.2.1 pDataBearer . . . . .	446
8.466unpack_wds_GetDefaultProfile_t Struct Reference . . . . .	446
8.466.1 Detailed Description . . . . .	446
8.466.2 Field Documentation . . . . .	446
8.466.2.1 apnname . . . . .	446

8.466.2.2 apnsize . . . . .	446
8.466.2.3 auth . . . . .	446
8.466.2.4 ipaddr . . . . .	447
8.466.2.5 ipaddrv6 . . . . .	447
8.466.2.6 name . . . . .	447
8.466.2.7 namesize . . . . .	447
8.466.2.8 pdptype . . . . .	447
8.466.2.9 pridns . . . . .	447
8.466.2.10pridnsv6 . . . . .	447
8.466.2.11secdns . . . . .	447
8.466.2.12secdnsv6 . . . . .	447
8.466.2.13username . . . . .	447
8.466.2.14usersize . . . . .	447
8.467unpack_wds_GetDefaultProfileNum_t Struct Reference . . . . .	447
8.467.1 Detailed Description . . . . .	447
8.467.2 Field Documentation . . . . .	447
8.467.2.1 index . . . . .	447
8.468unpack_wds_GetDormancyState_t Struct Reference . . . . .	447
8.468.1 Detailed Description . . . . .	448
8.468.2 Field Documentation . . . . .	448
8.468.2.1 dormancyState . . . . .	448
8.469unpack_wds_GetLastMobileIPError_t Struct Reference . . . . .	448
8.469.1 Detailed Description . . . . .	448
8.469.2 Field Documentation . . . . .	448
8.469.2.1 error . . . . .	448
8.470unpack_wds_GetMobileIP_t Struct Reference . . . . .	448
8.470.1 Detailed Description . . . . .	448
8.470.2 Field Documentation . . . . .	448
8.470.2.1 mipMode . . . . .	448
8.471unpack_wds_GetMobileIPProfile_t Struct Reference . . . . .	448

8.471.1 Detailed Description . . . . .	449
8.471.2 Field Documentation . . . . .	449
8.471.2.1 AAASPI . . . . .	449
8.471.2.2 AAASState . . . . .	449
8.471.2.3 address . . . . .	449
8.471.2.4 enabled . . . . .	449
8.471.2.5 HASPI . . . . .	449
8.471.2.6 HASState . . . . .	449
8.471.2.7 NAI . . . . .	449
8.471.2.8 naiSize . . . . .	449
8.471.2.9 primaryHA . . . . .	449
8.471.2.10 revTunneling . . . . .	450
8.471.2.11 secondaryHA . . . . .	450
8.472unpack_wds_GetPacketStatistics_t Struct Reference . . . . .	450
8.472.1 Detailed Description . . . . .	450
8.472.2 Field Documentation . . . . .	451
8.472.2.1 pRXDroppedCount . . . . .	451
8.472.2.2 pRXOkBytesCount . . . . .	451
8.472.2.3 pRXOKBytesLastCall . . . . .	451
8.472.2.4 pRXPacketErrors . . . . .	451
8.472.2.5 pRXPacketOverflows . . . . .	451
8.472.2.6 pRXPacketSuccesses . . . . .	451
8.472.2.7 pTXDroppedCount . . . . .	451
8.472.2.8 pTXOkBytesCount . . . . .	451
8.472.2.9 pTXOKBytesLastCall . . . . .	451
8.472.2.10 pTXPacketErrors . . . . .	451
8.472.2.11 pTXPacketOverflows . . . . .	451
8.472.2.12 pTXPacketSuccesses . . . . .	451
8.473unpack_wds_GetPacketStatus_t Struct Reference . . . . .	451
8.473.1 Detailed Description . . . . .	452

8.473.2 Field Documentation . . . . .	452
8.473.2.1 rXDroppedCount . . . . .	452
8.473.2.2 rXOkBytesCount . . . . .	452
8.473.2.3 rXOKBytesLastCall . . . . .	452
8.473.2.4 rXPacketErrors . . . . .	452
8.473.2.5 rXPacketOverflows . . . . .	452
8.473.2.6 rXPacketSuccesses . . . . .	452
8.473.2.7 tXDroppedCount . . . . .	452
8.473.2.8 tXOkBytesCount . . . . .	453
8.473.2.9 tXOKBytesLastCall . . . . .	453
8.473.2.10 tXPacketErrors . . . . .	453
8.473.2.11 tXPacketOverflows . . . . .	453
8.473.2.12 tXPacketSuccesses . . . . .	453
8.474unpack_wds_GetSessionDuration_t Struct Reference . . . . .	453
8.474.1 Detailed Description . . . . .	453
8.474.2 Field Documentation . . . . .	453
8.474.2.1 callDuration . . . . .	453
8.475unpack_wds_GetSessionState_t Struct Reference . . . . .	453
8.475.1 Detailed Description . . . . .	453
8.475.2 Field Documentation . . . . .	453
8.475.2.1 connectionStatus . . . . .	453
8.476unpack_wds_RMSetTransferStatistics_t Struct Reference . . . . .	454
8.477unpack_wds_SetMobileIPProfile_t Struct Reference . . . . .	454
8.478unpack_wds_SLQSCreateProfile_t Struct Reference . . . . .	454
8.478.1 Detailed Description . . . . .	454
8.478.2 Field Documentation . . . . .	454
8.478.2.1 pCreateProfileOut . . . . .	454
8.478.2.2 pProfileID . . . . .	454
8.478.2.3 Tlvresult . . . . .	454
8.479unpack_wds_SLQSDeleteProfile_t Struct Reference . . . . .	454

8.479.1 Detailed Description . . . . .	454
8.479.2 Field Documentation . . . . .	454
8.479.2.1 extendedErrorCode . . . . .	454
8.480unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . . .	455
8.480.1 Detailed Description . . . . .	455
8.480.2 Field Documentation . . . . .	456
8.480.2.1 _3gppRelease . . . . .	456
8.480.2.2 defaultPDNEnabled . . . . .	456
8.480.2.3 LTEAttachProfile . . . . .	456
8.480.2.4 LTEAttachProfileList . . . . .	456
8.480.2.5 LTEAttachProfileListLen . . . . .	456
8.480.2.6 profileList . . . . .	456
8.481unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . . .	456
8.481.1 Detailed Description . . . . .	456
8.481.2 Field Documentation . . . . .	456
8.481.2.1 currNetworkInfo . . . . .	456
8.481.2.2 networkInfoLen . . . . .	456
8.481.2.3 prefNetwork . . . . .	456
8.482unpack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . . .	457
8.482.1 Detailed Description . . . . .	457
8.482.2 Field Documentation . . . . .	457
8.482.2.1 current_channel_rx_rate . . . . .	457
8.482.2.2 current_channel_tx_rate . . . . .	457
8.482.2.3 max_channel_rx_rate . . . . .	457
8.482.2.4 max_channel_tx_rate . . . . .	457
8.483unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . . .	458
8.483.1 Detailed Description . . . . .	458
8.483.2 Field Documentation . . . . .	458
8.483.2.1 curDataBearerTechnology . . . . .	458
8.483.2.2 dataBearerMask . . . . .	458



8.483.2.3 lastCallDataBearerTechnology . . . . .	458
8.484unpack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . . .	458
8.484.1 Detailed Description . . . . .	458
8.484.2 Field Documentation . . . . .	459
8.484.2.1 callEndReason . . . . .	459
8.484.2.2 channelRate . . . . .	459
8.484.2.3 connectionStatus . . . . .	459
8.484.2.4 dataBearerTech . . . . .	459
8.484.2.5 dormancyStatus . . . . .	459
8.484.2.6 lastCallDataBearerTech . . . . .	459
8.484.2.7 lastCallRXOKBytesCnt . . . . .	459
8.484.2.8 lastCallTXOKBytesCnt . . . . .	459
8.484.2.9 mdmCallDurationActive . . . . .	459
8.484.2.10rxOKBytesCount . . . . .	459
8.484.2.11txOKBytesCount . . . . .	459
8.485unpack_wds_SLQSGetProfileSettings_t Struct Reference . . . . .	459
8.485.1 Field Documentation . . . . .	460
8.485.1.1 pProfileSettings . . . . .	460
8.485.1.2 ProfileType . . . . .	460
8.485.1.3 Tlvresult . . . . .	460
8.486unpack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . . .	460
8.486.1 Detailed Description . . . . .	460
8.486.2 Field Documentation . . . . .	461
8.486.2.1 APNName . . . . .	461
8.486.2.2 Authentication . . . . .	461
8.486.2.3 DomainList . . . . .	461
8.486.2.4 GPRSGrantedQoS . . . . .	461
8.486.2.5 GWAddressV4 . . . . .	461
8.486.2.6 IMCNflag . . . . .	461
8.486.2.7 IPFamilyPreference . . . . .	461

8.486.2.8 IPv4 . . . . .	461
8.486.2.9 IPV6AddrInfo . . . . .	461
8.486.2.10 IPV6GWAddrInfo . . . . .	461
8.486.2.11 Mtu . . . . .	461
8.486.2.12 PCSCFAddrPCO . . . . .	461
8.486.2.13 PCSCFFQDNAddrList . . . . .	461
8.486.2.14 PDType . . . . .	462
8.486.2.15 PrimaryDNSV4 . . . . .	462
8.486.2.16 PrimaryDNSV6 . . . . .	462
8.486.2.17 ProfileID . . . . .	462
8.486.2.18 ProfileName . . . . .	462
8.486.2.19 SecondaryDNSV4 . . . . .	462
8.486.2.20 SecondaryDNSV6 . . . . .	462
8.486.2.21 ServerAddrList . . . . .	462
8.486.2.22 SubnetMaskV4 . . . . .	462
8.486.2.23 Technology . . . . .	462
8.486.2.24 UMTSGrantedQoS . . . . .	462
8.486.2.25 Username . . . . .	462
8.487unpack_wds_SLQSMModifyProfile_t Struct Reference . . . . .	462
8.487.1 Detailed Description . . . . .	462
8.487.2 Field Documentation . . . . .	462
8.487.2.1 pExtErrorCode . . . . .	462
8.488unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . . .	462
8.488.1 Detailed Description . . . . .	463
8.488.2 Field Documentation . . . . .	463
8.488.2.1 Tlvresult . . . . .	463
8.489unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . . .	463
8.489.1 Detailed Description . . . . .	463
8.489.2 Field Documentation . . . . .	463
8.489.2.1 bearerID . . . . .	463

8.489.2.2 conn_status . . . . .	463
8.489.2.3 ipFamily . . . . .	464
8.489.2.4 reconfigReqd . . . . .	464
8.489.2.5 sessionEndReason . . . . .	464
8.489.2.6 techName . . . . .	464
8.489.2.7 verboseSessnEndReason . . . . .	464
8.489.2.8 verboseSessnEndReasonType . . . . .	464
8.490unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . . .	464
8.490.1 Detailed Description . . . . .	464
8.490.2 Field Documentation . . . . .	465
8.490.2.1 currDBTechAvail . . . . .	465
8.490.2.2 currNWInfo . . . . .	465
8.490.2.3 dataSysStatAvail . . . . .	465
8.490.2.4 dBTechAvail . . . . .	465
8.490.2.5 dBTechnology . . . . .	465
8.490.2.6 dormancyStatAvail . . . . .	465
8.490.2.7 dormancyStatus . . . . .	465
8.490.2.8 mipstatAvail . . . . .	465
8.490.2.9 mipStatus . . . . .	465
8.490.2.10netInfoLen . . . . .	465
8.490.2.11prefNetwork . . . . .	465
8.490.2.12ratMask . . . . .	465
8.490.2.13rx_bytes . . . . .	465
8.490.2.14rx_pkts . . . . .	465
8.490.2.15soMask . . . . .	465
8.490.2.16tx_bytes . . . . .	465
8.490.2.17tx_pkts . . . . .	465
8.490.2.18ferStatAvail . . . . .	465
8.491unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . . .	465
8.491.1 Detailed Description . . . . .	465

8.491.2 Field Documentation . . . . .	466
8.491.2.1 pHwConfig . . . . .	466
8.491.2.2 pRequestOptionList . . . . .	466
8.492unpack_wds_SLQSSGetLoopback_t Struct Reference . . . . .	466
8.492.1 Detailed Description . . . . .	466
8.492.2 Field Documentation . . . . .	466
8.492.2.1 ByteLoopbackMode . . . . .	466
8.492.2.2 ByteLoopbackMultiplier . . . . .	466
8.493unpack_wds_SLQSSStartDataSession_t Struct Reference . . . . .	466
8.493.1 Detailed Description . . . . .	467
8.493.2 Field Documentation . . . . .	467
8.493.2.1 pFailureReason . . . . .	467
8.493.2.2 psid . . . . .	467
8.493.2.3 pVerboseFailReasonType . . . . .	467
8.493.2.4 pVerboseFailureReason . . . . .	467
8.494unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . . .	467
8.494.1 Detailed Description . . . . .	467
8.494.2 Field Documentation . . . . .	468
8.494.2.1 apnName . . . . .	468
8.494.2.2 bearerId . . . . .	468
8.494.2.3 contextId . . . . .	468
8.494.2.4 ipv4Address . . . . .	468
8.494.2.5 ipv4GWAddress . . . . .	468
8.494.2.6 ipv6Address . . . . .	468
8.494.2.7 ipv6GWAddress . . . . .	468
8.494.2.8 prDNSIPv4Address . . . . .	468
8.494.2.9 prDNSIPv6Address . . . . .	468
8.494.2.10prPCSCFIPv4Address . . . . .	468
8.494.2.11prPCSCFIPv6Address . . . . .	468
8.494.2.12seDNSIPv4Address . . . . .	469

8.494.2.13seDNSIPv6Address . . . . .	469
8.494.2.14sePCSCFIPv4Address . . . . .	469
8.494.2.15sePCSCFIPv6Address . . . . .	469
8.495UnPackGetProfileSettingOut Struct Reference . . . . .	469
8.495.1 Field Documentation . . . . .	469
8.495.1.1 curProfile . . . . .	469
8.495.1.2 pExtErrCode . . . . .	469
8.496unpackWdsProfileParam Union Reference . . . . .	469
8.496.1 Field Documentation . . . . .	469
8.496.1.1 SlqsProfile3GPP . . . . .	469
8.496.1.2 SlqsProfile3GPP2 . . . . .	469
8.497wds_currNetworkInfo Struct Reference . . . . .	469
8.497.1 Detailed Description . . . . .	469
8.497.2 Field Documentation . . . . .	471
8.497.2.1 NetworkType . . . . .	471
8.497.2.2 RATMask . . . . .	471
8.497.2.3 SOMask . . . . .	471
8.498wds_DataULongLongTlv Struct Reference . . . . .	471
8.498.1 Field Documentation . . . . .	471
8.498.1.1 TlvPresent . . . . .	471
8.498.1.2 ullData . . . . .	471
8.499wds_DataULongTlv Struct Reference . . . . .	471
8.499.1 Field Documentation . . . . .	472
8.499.1.1 TlvPresent . . . . .	472
8.499.1.2 ulData . . . . .	472
8.500wds_DHCPLeaseOptTlv Struct Reference . . . . .	472
8.500.1 Field Documentation . . . . .	472
8.500.1.1 numOpt . . . . .	472
8.500.1.2 optList . . . . .	472
8.500.1.3 optListData . . . . .	472

8.500.1.4 TlvPresent . . . . .	472
8.501wds_DHCPLeaseStateTlv Struct Reference . . . . .	472
8.501.1 Field Documentation . . . . .	472
8.501.1.1 leaseState . . . . .	472
8.501.1.2 TlvPresent . . . . .	472
8.502wds_DHCPOpt Struct Reference . . . . .	472
8.502.1 Field Documentation . . . . .	473
8.502.1.1 optCode . . . . .	473
8.502.1.2 optValLen . . . . .	473
8.502.1.3 pOptVal . . . . .	473
8.503wds_DHCPProfileIdTlv Struct Reference . . . . .	473
8.503.1 Field Documentation . . . . .	473
8.503.1.1 profileId . . . . .	473
8.503.1.2 profileType . . . . .	473
8.503.1.3 TlvPresent . . . . .	473
8.504wds_DHCPv4HWConfig Struct Reference . . . . .	473
8.504.1 Detailed Description . . . . .	473
8.504.2 Field Documentation . . . . .	474
8.504.2.1 chaddr . . . . .	474
8.504.2.2 chaddrLen . . . . .	474
8.504.2.3 hwType . . . . .	474
8.505wds_DHCPv4Option Struct Reference . . . . .	474
8.505.1 Detailed Description . . . . .	474
8.505.2 Field Documentation . . . . .	474
8.505.2.1 optCode . . . . .	474
8.505.2.2 optVal . . . . .	474
8.505.2.3 optValLen . . . . .	474
8.506wds_DHCPv4OptionList Struct Reference . . . . .	474
8.506.1 Detailed Description . . . . .	475
8.506.2 Field Documentation . . . . .	475

8.506.2.1 numOpt . . . . .	475
8.506.2.2 pOptList . . . . .	475
8.507wds_DHCPv4ProfileId Struct Reference . . . . .	475
8.507.1 Detailed Description . . . . .	475
8.507.2 Field Documentation . . . . .	475
8.507.2.1 profileId . . . . .	475
8.507.2.2 profileType . . . . .	475
8.508wds_Domain Struct Reference . . . . .	476
8.508.1 Detailed Description . . . . .	476
8.508.2 Field Documentation . . . . .	476
8.508.2.1 domainLen . . . . .	476
8.508.2.2 domainName . . . . .	476
8.509wds_DomainNameList Struct Reference . . . . .	476
8.509.1 Detailed Description . . . . .	476
8.509.2 Field Documentation . . . . .	477
8.509.2.1 domain . . . . .	477
8.509.2.2 numInstances . . . . .	477
8.510wds_GPRSQoS Struct Reference . . . . .	477
8.510.1 Detailed Description . . . . .	477
8.510.2 Field Documentation . . . . .	477
8.510.2.1 delayClass . . . . .	477
8.510.2.2 meanThroughputClass . . . . .	477
8.510.2.3 peakThroughputClass . . . . .	477
8.510.2.4 precedenceClass . . . . .	477
8.510.2.5 reliabilityClass . . . . .	478
8.511wds_IPv4AdTlv Struct Reference . . . . .	478
8.511.1 Field Documentation . . . . .	478
8.511.1.1 IPv4Addr . . . . .	478
8.511.1.2 TlvPresent . . . . .	478
8.512wds_IPV6AddressInfo Struct Reference . . . . .	478

8.512.1 Detailed Description . . . . .	478
8.512.2 Field Documentation . . . . .	478
8.512.2.1 IPAddressV6 . . . . .	478
8.512.2.2 IPV6PrefixLen . . . . .	478
8.513wds_IPV6GWAddressInfo Struct Reference . . . . .	478
8.513.1 Detailed Description . . . . .	479
8.513.2 Field Documentation . . . . .	479
8.513.2.1 gwAddressV6 . . . . .	479
8.513.2.2 gwV6PrefixLen . . . . .	479
8.514wds_PCSCFFQDNAddress Struct Reference . . . . .	479
8.514.1 Detailed Description . . . . .	479
8.514.2 Field Documentation . . . . .	479
8.514.2.1 fqdnAddr . . . . .	480
8.514.2.2 fqdnLen . . . . .	480
8.515wds_PCSCFFQDNAddressList Struct Reference . . . . .	480
8.515.1 Detailed Description . . . . .	480
8.515.2 Field Documentation . . . . .	480
8.515.2.1 numInstances . . . . .	480
8.515.2.2 pcsfFQDNAddress . . . . .	480
8.516wds_PCSCFIPv4ServerAddressList Struct Reference . . . . .	480
8.516.1 Detailed Description . . . . .	480
8.516.2 Field Documentation . . . . .	481
8.516.2.1 numInstances . . . . .	481
8.516.2.2 pcsfIPv4Addr . . . . .	481
8.517wds_ProfileIdentifier Struct Reference . . . . .	481
8.517.1 Detailed Description . . . . .	481
8.517.2 Field Documentation . . . . .	481
8.517.2.1 profileIndex . . . . .	481
8.517.2.2 profileType . . . . .	481
8.518wds_profileInfo Union Reference . . . . .	481



8.518.1 Detailed Description . . . . .	481
8.518.2 Field Documentation . . . . .	482
8.518.2.1 SlqsProfile3GPP . . . . .	482
8.518.2.2 SlqsProfile3GPP2 . . . . .	482
8.519wds_TrStatInd Struct Reference . . . . .	482
8.519.1 Detailed Description . . . . .	482
8.519.2 Field Documentation . . . . .	482
8.519.2.1 statsMask . . . . .	482
8.519.2.2 statsPeriod . . . . .	482
8.520wds_UMTSMInQoS Struct Reference . . . . .	482
8.520.1 Detailed Description . . . . .	483
8.520.2 Field Documentation . . . . .	484
8.520.2.1 deliveryErrSDU . . . . .	484
8.520.2.2 grntDownlinkBitrate . . . . .	485
8.520.2.3 grntUplinkBitrate . . . . .	485
8.520.2.4 maxDownlinkBitrate . . . . .	485
8.520.2.5 maxSDUSize . . . . .	485
8.520.2.6 maxUplinkBitrate . . . . .	485
8.520.2.7 qosDeliveryOrder . . . . .	485
8.520.2.8 resBerRatio . . . . .	485
8.520.2.9 sduErrorRatio . . . . .	485
8.520.2.10trafficClass . . . . .	485
8.520.2.11trafficPriority . . . . .	485
8.520.2.12transferDelay . . . . .	485
8.521wdsDhcpv4HwConfig Struct Reference . . . . .	485
8.521.1 Detailed Description . . . . .	485
8.521.2 Field Documentation . . . . .	485
8.521.2.1 chaddr . . . . .	485
8.521.2.2 chaddrLen . . . . .	486
8.521.2.3 hwType . . . . .	486

8.522wdsDhcpv4Option Struct Reference . . . . .	486
8.522.1 Detailed Description . . . . .	486
8.522.2 Field Documentation . . . . .	486
8.522.2.1 optCode . . . . .	486
8.522.2.2 optVal . . . . .	486
8.522.2.3 optValLen . . . . .	486
8.523wdsDhcpv4OptionList Struct Reference . . . . .	486
8.523.1 Detailed Description . . . . .	486
8.523.2 Field Documentation . . . . .	487
8.523.2.1 numOpt . . . . .	487
8.523.2.2 pOptList . . . . .	487
8.524wdsDhcpv4ProfileId Struct Reference . . . . .	487
8.524.1 Detailed Description . . . . .	487
8.524.2 Field Documentation . . . . .	487
8.524.2.1 profileId . . . . .	487
8.524.2.2 profileType . . . . .	487
<b>9 File Documentation . . . . .</b>	<b>489</b>
9.1 apdoxypages.c File Reference . . . . .	489
9.1.1 Detailed Description . . . . .	489
9.2 common.h File Reference . . . . .	489
9.2.1 Macro Definition Documentation . . . . .	491
9.2.1.1 DEault_LOC_TIMEOUT_IN_SEC . . . . .	491
9.2.1.2 MINREQBKLEN . . . . .	491
9.2.1.3 MSGID_AND_LEN . . . . .	491
9.2.1.4 MSGID_DONT_CARE . . . . .	491
9.2.1.5 SDK_VALIDATE_INPUT_PACK_PARAM . . . . .	491
9.2.1.6 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID . . . . .	491
9.2.1.7 SDU_HDR_LEN . . . . .	491
9.2.1.8 UNUSEDPARAM . . . . .	491
9.2.2 Typedef Documentation . . . . .	491

9.2.2.1	logger	491
9.2.3	Enumeration Type Documentation	491
9.2.3.1	eLOG_LEVEL	491
9.2.3.2	eQMI_SVC	492
9.2.3.3	eTimeout	492
9.2.3.4	msgtype	492
9.2.4	Function Documentation	493
9.2.4.1	fill_pack_ctx(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t svc, int timeout)	493
9.2.4.2	fill_sdu_hdr(pack_qmi_t *pCtx, uint8_t *pReqBuf)	493
9.2.4.3	get_version()	493
9.2.4.4	helper_get_resp_ctx(uint8_t svc, uint8_t *pbuf, uint32_t len, unpack_qmi_t *pCtx)	493
9.2.4.5	helper_get_xid(uint8_t *qmi_resp)	493
9.2.4.6	helper_set_log_func(logger func)	493
9.2.4.7	helper_set_log_lvl(uint8_t lvl)	493
9.2.4.8	libpack_GetVersion()	493
9.2.4.9	libpack_log(uint8_t lvl, const char *fmt,...)	494
9.2.4.10	unpack_result_code_only(uint8_t *pMdmResp)	494
9.2.5	Variable Documentation	494
9.2.5.1	glog	494
9.2.5.2	gloglvl	494
9.3	dms.h File Reference	494
9.3.1	Macro Definition Documentation	500
9.3.1.1	ACT_CODE_MAX_SIZE	500
9.3.1.2	CK_MAX_SIZE	500
9.3.1.3	DMS_IMGDETAILS_LEN	500
9.3.1.4	DMS_MAX_CUST_ID_LEN	500
9.3.1.5	DMS_MAX_CUST_VALUE_LEN	500
9.3.1.6	DMS_MAX_FWUPDATE_LOG_STR_SZ	501
9.3.1.7	DMS_MAX_FWUPDATE_REF_STR_SZ	501
9.3.1.8	DMS_PM_FACTORY	501

9.3.1.9	DMS_PM_LOW	501
9.3.1.10	DMS_PM_OFFLINE	501
9.3.1.11	DMS_PM_ONLINE	501
9.3.1.12	DMS_PM_PERSISTENT_LOW	501
9.3.1.13	DMS_PM_RESET	501
9.3.1.14	DMS_PM_SHUT_DOWN	501
9.3.1.15	DMS_SET_REPORT_DISABLE	501
9.3.1.16	DMS_SET_REPORT_ENABLE	501
9.3.1.17	DMS_SLQSFWINFO_APPVERSION_SZ	501
9.3.1.18	DMS_SLQSFWINFO_BOOTVERSION_SZ	501
9.3.1.19	DMS_SLQSFWINFO_CARRIER_SZ	501
9.3.1.20	DMS_SLQSFWINFO_CUR_CARR_NAME	501
9.3.1.21	DMS_SLQSFWINFO_CUR_CARR_REV	501
9.3.1.22	DMS_SLQSFWINFO_MODELID_SZ	501
9.3.1.23	DMS_SLQSFWINFO_PACKAGEID_SZ	501
9.3.1.24	DMS_SLQSFWINFO_PRIVERSION_SZ	501
9.3.1.25	DMS_SLQSFWINFO_SKU_SZ	501
9.3.1.26	DMS_SWI_SET_IND_DISABLE	501
9.3.1.27	DMS_SWI_SET_IND_ENABLE	501
9.3.1.28	DMS_UINT8_MAX_STRING_SZ	501
9.3.1.29	ERI_DATA_MAX_SIZE	502
9.3.1.30	MAX_BUILD_ID_LEN	502
9.3.1.31	MEID_MAX_SIZE	502
9.3.1.32	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	502
9.3.1.33	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	502
9.3.1.34	SPC_SIZE	502
9.3.1.35	UNIQUE_ID_LEN	502
9.3.2	Function Documentation	502
9.3.2.1	pack_dms_ActivateAutomatic(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_ActivateAutomatic_t *pReq)	502

9.3.2.2	<a href="#">pack_dms_GetActivationState(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</a>	502
9.3.2.3	<a href="#">pack_dms_GetBandCapability(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	503
9.3.2.4	<a href="#">pack_dms_GetCrashAction(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	503
9.3.2.5	<a href="#">pack_dms_GetCustFeature(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	504
9.3.2.6	<a href="#">pack_dms_GetCustFeaturesV2(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_GetCustFeaturesV2_t *reqArg)</a>	504
9.3.2.7	<a href="#">pack_dms_GetDeviceCap(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	504
9.3.2.8	<a href="#">pack_dms_GetDeviceCapabilities(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	504
9.3.2.9	<a href="#">pack_dms_GetDeviceHardwareRev(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	505
9.3.2.10	<a href="#">pack_dms_GetDeviceMfr(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	505
9.3.2.11	<a href="#">pack_dms_GetDeviceSerialNumbers(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	506
9.3.2.12	<a href="#">pack_dms_GetFirmwareInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	506
9.3.2.13	<a href="#">pack_dms_GetFirmwareRevision(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	506
9.3.2.14	<a href="#">pack_dms_GetFirmwareRevisions(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	507
9.3.2.15	<a href="#">pack_dms_GetFSN(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	507
9.3.2.16	<a href="#">pack_dms_GetHardwareRevision(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	508
9.3.2.17	<a href="#">pack_dms_GetIMSI(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	508
9.3.2.18	<a href="#">pack_dms_GetManufacturer(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</a>	508
9.3.2.19	<a href="#">pack_dms_GetModelID(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	509
9.3.2.20	<a href="#">pack_dms_GetNetworkTime(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</a>	509
9.3.2.21	<a href="#">pack_dms_GetOfflineReason(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</a>	510

9.3.2.22	<code>pack_dms_GetPower(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	510
9.3.2.23	<code>pack_dms_GetPRLVersion(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	510
9.3.2.24	<code>pack_dms_GetSerialNumbers(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	511
9.3.2.25	<code>pack_dms_GetUSBComp(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	511
9.3.2.26	<code>pack_dms_GetVoiceNumber(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	512
9.3.2.27	<code>pack_dms_ResetToFactoryDefaults(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_ResetToFactoryDefaults_t *pReq)</code>	512
9.3.2.28	<code>pack_dms_SetActivationStatusCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetActivationStatusCallback_t *reqArg)</code>	512
9.3.2.29	<code>pack_dms_SetCrashAction(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetCrashAction_t reqArg)</code>	513
9.3.2.30	<code>pack_dms_SetCustFeature(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetCustFeature_t *reqArg)</code>	514
9.3.2.31	<code>pack_dms_SetCustFeaturesV2(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetCustFeaturesV2_t *reqArg)</code>	514
9.3.2.32	<code>pack_dms_SetEventReport(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetEventReport_t *reqArg)</code>	514
9.3.2.33	<code>pack_dms_SetFirmwarePreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	515
9.3.2.34	<code>pack_dms_SetPower(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetPower_t *reqArg)</code>	515
9.3.2.35	<code>pack_dms_SetUSBComp(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetUSBComp_t *reqArg)</code>	516
9.3.2.36	<code>pack_dms_SLQSDmsSwiGetResetInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	516
9.3.2.37	<code>pack_dms_SLQSDmsSwiIndicationRegister(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSDmsSwiIndicationRegister_t *reqArg)</code>	516
9.3.2.38	<code>pack_dms_SLQSGetBandCapability(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)</code>	517
9.3.2.39	<code>pack_dms_SLQSGetERIFFile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	517
9.3.2.40	<code>pack_dms_SLQSSwiClearDyingGaspStatistics(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	518
9.3.2.41	<code>pack_dms_SLQSSwiGetCrashInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiGetCrashInfo_t *pReq)</code>	518

9.3.2.42	<code>pack_dms_SLQSSwiGetDyingGaspCfg(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	519
9.3.2.43	<code>pack_dms_SLQSSwiGetDyingGaspStatistics(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	519
9.3.2.44	<code>pack_dms_SLQSSwiGetFirmwareCurr(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	519
9.3.2.45	<code>pack_dms_SLQSSwiGetFwUpdateStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	520
9.3.2.46	<code>pack_dms_SLQSSwiGetHostDevInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	520
9.3.2.47	<code>pack_dms_SLQSSwiGetOSInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	520
9.3.2.48	<code>pack_dms_SLQSSwiGetSerialNoExt(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	521
9.3.2.49	<code>pack_dms_SLQSSwiSetDyingGaspCfg(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiSetDyingGaspCfg_t *reqArg)</code>	521
9.3.2.50	<code>pack_dms_SLQSSwiSetHostDevInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiSetHostDevInfo_t *pReq)</code>	522
9.3.2.51	<code>pack_dms_SLQSSwiSetOSInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiSetOSInfo_t *pReq)</code>	522
9.3.2.52	<code>pack_dms_SLQSUIMGetState(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	522
9.3.2.53	<code>pack_dms_UIMChangePIN(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMChangePIN_t *pReq)</code>	523
9.3.2.54	<code>pack_dms_UIMGetControlKeyStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMGetControlKeyStatus_t *pReq)</code>	523
9.3.2.55	<code>pack_dms_UIMGetICCID(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMGetICCID_t *reqArg)</code>	524
9.3.2.56	<code>pack_dms_UIMGetPINStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	524
9.3.2.57	<code>pack_dms_UIMSetControlKeyProtection(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMSetControlKeyProtection_t *pReq)</code>	525
9.3.2.58	<code>pack_dms_UIMSetPINProtection(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMSetPINProtection_t *pReq)</code>	525
9.3.2.59	<code>pack_dms_UIMUnblockControlKey(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMUnblockControlKey_t *pReq)</code>	525
9.3.2.60	<code>pack_dms_UIMUnblockPIN(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMUnblockPIN_t *pReq)</code>	526
9.3.2.61	<code>pack_dms_UIMVerifyPIN(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMVerifyPIN_t *pReq)</code>	526

9.3.2.62	pack_dms_ValidateSPC(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_ResetToFactoryDefaults_t *pReq) . . . . .	527
9.3.2.63	unpack_dms_ActivateAutomatic(uint8_t *pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t *pOutput) . . . . .	527
9.3.2.64	unpack_dms_GetActivationState(uint8_t *pResp, uint16_t respLen, unpack_dms_GetActivationState_t *pOutput) . . . . .	527
9.3.2.65	unpack_dms_GetBandCapability(uint8_t *pResp, uint16_t respLen, unpack_dms_GetBandCapability_t *pOutput) . . . . .	528
9.3.2.66	unpack_dms_GetCrashAction(uint8_t *pResp, uint16_t respLen, unpack_dms_GetCrashAction_t *pOutput) . . . . .	528
9.3.2.67	unpack_dms_GetCustFeature(uint8_t *pResp, uint16_t respLen, unpack_dms_GetCustFeature_t *pOutput) . . . . .	529
9.3.2.68	unpack_dms_GetCustFeaturesV2(uint8_t *pResp, uint16_t respLen, unpack_dms_GetCustFeaturesV2_t *pOutput) . . . . .	529
9.3.2.69	unpack_dms_GetDeviceCap(uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceCap_t *pOutput) . . . . .	529
9.3.2.70	unpack_dms_GetDeviceCapabilities(uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceCapabilities_t *pOutput) . . . . .	529
9.3.2.71	unpack_dms_GetDeviceHardwareRev(uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceHardwareRev_t *pOutput) . . . . .	530
9.3.2.72	unpack_dms_GetDeviceMfr(uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceMfr_t *pOutput) . . . . .	530
9.3.2.73	unpack_dms_GetDeviceSerialNumbers(uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceSerialNumbers_t *pOutput) . . . . .	531
9.3.2.74	unpack_dms_GetFirmwareInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareInfo_t *pOutput) . . . . .	531
9.3.2.75	unpack_dms_GetFirmwareRevision(uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareRevision_t *pOutput) . . . . .	531
9.3.2.76	unpack_dms_GetFirmwareRevisions(uint8_t *pResp, uint16_t respLen, unpack_dms_GetFirmwareRevisions_t *pOutput) . . . . .	532
9.3.2.77	unpack_dms_GetFSN(uint8_t *pResp, uint16_t respLen, unpack_dms_GetFSN_t *pOutput) . . . . .	532
9.3.2.78	unpack_dms_GetHardwareRevision(uint8_t *pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t *pOutput) . . . . .	533
9.3.2.79	unpack_dms_GetIMSI(uint8_t *pResp, uint16_t respLen, unpack_dms_GetIMSI_t *pOutput) . . . . .	533
9.3.2.80	unpack_dms_GetManufacturer(uint8_t *pResp, uint16_t respLen, unpack_dms_GetManufacturer_t *pOutput) . . . . .	533
9.3.2.81	unpack_dms_GetModelID(uint8_t *pResp, uint16_t respLen, unpack_dms_GetModelID_t *pOutput) . . . . .	534



9.3.2.82	<a href="#">unpack_dms_GetNetworkTime(uint8_t *pResp, uint16_t respLen, unpack_dms_GetNetworkTime_t *pOutput)</a>	534
9.3.2.83	<a href="#">unpack_dms_GetOfflineReason(uint8_t *pResp, uint16_t respLen, unpack_dms_GetOfflineReason_t *pOutput)</a>	535
9.3.2.84	<a href="#">unpack_dms_GetPower(uint8_t *pResp, uint16_t respLen, unpack_dms_GetPower_t *pOutput)</a>	535
9.3.2.85	<a href="#">unpack_dms_GetPRLVersion(uint8_t *pResp, uint16_t respLen, unpack_dms_GetPRLVersion_t *pOutput)</a>	535
9.3.2.86	<a href="#">unpack_dms_GetSerialNumbers(uint8_t *pResp, uint16_t respLen, unpack_dms_GetSerialNumbers_t *pOutput)</a>	536
9.3.2.87	<a href="#">unpack_dms_GetUSBComp(uint8_t *pResp, uint16_t respLen, unpack_dms_GetUSBComp_t *pOutput)</a>	536
9.3.2.88	<a href="#">unpack_dms_GetVoiceNumber(uint8_t *pResp, uint16_t respLen, unpack_dms_GetVoiceNumber_t *pOutput)</a>	537
9.3.2.89	<a href="#">unpack_dms_ResetToFactoryDefaults(uint8_t *pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t *pOutput)</a>	537
9.3.2.90	<a href="#">unpack_dms_SetActivationStatusCallback(uint8_t *pResp, uint16_t respLen, unpack_dms_SetActivationStatusCallback_t *pOutput)</a>	537
9.3.2.91	<a href="#">unpack_dms_SetCrashAction(uint8_t *pResp, uint16_t respLen, unpack_dms_SetCrashAction_t *pOutput)</a>	538
9.3.2.92	<a href="#">unpack_dms_SetCustFeature(uint8_t *pResp, uint16_t respLen, unpack_dms_SetCustFeature_t *pOutput)</a>	538
9.3.2.93	<a href="#">unpack_dms_SetCustFeaturesV2(uint8_t *pResp, uint16_t respLen, unpack_dms_SetCustFeaturesV2_t *pOutput)</a>	539
9.3.2.94	<a href="#">unpack_dms_SetEventReport(uint8_t *pResp, uint16_t respLen, unpack_dms_SetEventReport_t *pOutput)</a>	539
9.3.2.95	<a href="#">unpack_dms_SetEventReport_ind(uint8_t *pResp, uint16_t respLen, unpack_dms_SetEventReport_ind_t *pOutput)</a>	539
9.3.2.96	<a href="#">unpack_dms_SetFirmwarePreference(uint8_t *pResp, uint16_t respLen, unpack_dms_SetFirmwarePreference_t *pOutput)</a>	540
9.3.2.97	<a href="#">unpack_dms_SetPower(uint8_t *pResp, uint16_t respLen, unpack_dms_SetPower_t *pOutput)</a>	540
9.3.2.98	<a href="#">unpack_dms_SetUSBComp(uint8_t *pResp, uint16_t respLen, unpack_dms_SetUSBComp_t *pOutput)</a>	541
9.3.2.99	<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_t *pOutput)</a>	541
9.3.2.100	<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_Ind(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t *pOutput)</a>	541
9.3.2.101	<a href="#">unpack_dms_SLQSDmsSwiIndicationRegister(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiIndicationRegister_t *pOutput)</a>	542

9.3.2.102 unpack_dms_SLQSGetBandCapability(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetBandCapability_t *pOutput) . . . . .	542
9.3.2.103 unpack_dms_SLQSGetERIFile(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetERIFile_t *pOutput) . . . . .	543
9.3.2.104 unpack_dms_SLQSSwiClearDyingGaspStatistics(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiClearDyingGaspStatistics_t *pOutput) . . . . .	543
9.3.2.105 unpack_dms_SLQSSwiGetCrashInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetCrashInfo_t *pOutput) . . . . .	544
9.3.2.106 unpack_dms_SLQSSwiGetDyingGaspCfg(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspCfg_t *pOutput) . . . . .	544
9.3.2.107 unpack_dms_SLQSSwiGetDyingGaspStatistics(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspStatistics_t *pOutput) . . . . .	544
9.3.2.108 unpack_dms_SLQSSwiGetFirmwareCurr(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetFirmwareCurr_t *pOutput) . . . . .	545
9.3.2.109 unpack_dms_SLQSSwiGetFwUpdateStatus(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetFwUpdateStatus_t *pOutput) . . . . .	545
9.3.2.110 unpack_dms_SLQSSwiGetHostDevInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetHostDevInfo_t *pOutput) . . . . .	546
9.3.2.111 unpack_dms_SLQSSwiGetOSInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetOSInfo_t *pOutput) . . . . .	546
9.3.2.112 unpack_dms_SLQSSwiGetSerialNoExt(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetSerialNoExt_t *pOutput) . . . . .	546
9.3.2.113 unpack_dms_SLQSSwiSetDyingGaspCfg(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiSetDyingGaspCfg_t *pOutput) . . . . .	547
9.3.2.114 unpack_dms_SLQSSwiSetHostDevInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiSetHostDevInfo_t *pOutput) . . . . .	547
9.3.2.115 unpack_dms_SLQSSwiSetOSInfo(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiSetOSInfo_t *pOutput) . . . . .	548
9.3.2.116 unpack_dms_SLQSUIGetState(uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSUIGetState_t *pOutput) . . . . .	548
9.3.2.117 unpack_dms_UIMChangePIN(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput) . . . . .	548
9.3.2.118 unpack_dms_UIMGetControlKeyStatus(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMGetControlKeyStatus_t *pOutput) . . . . .	549
9.3.2.119 unpack_dms_UIMGetICCID(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMGetICCID_t *pOutput) . . . . .	549
9.3.2.120 unpack_dms_UIMGetPINStatus(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMGetPINStatus_t *pOutput) . . . . .	550
9.3.2.121 unpack_dms_UIMSetControlKeyProtection(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetControlKeyProtection_t *pOutput) . . . . .	550

9.3.2.122	unpack_dms_UIMSetPINProtection(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)	550
9.3.2.123	unpack_dms_UIMUnblockControlKey(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMUnblockControlKey_t *pOutput)	551
9.3.2.124	unpack_dms_UIMUnblockPIN(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)	551
9.3.2.125	unpack_dms_UIMVerifyPIN(uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)	552
9.3.2.126	unpack_dms_ValidateSPC(uint8_t *pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t *pOutput)	552
9.4	fms.h File Reference	552
9.4.1	Macro Definition Documentation	553
9.4.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	553
9.4.1.2	FMS_GOBI_LISTENTRIES_MAX	553
9.4.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	553
9.4.1.4	FMS_GOBI_MBN_IMG_ID_STR_LEN	554
9.4.1.5	FMS_IMAGE_ID_BUILD_ID_LEN	554
9.4.1.6	FMS_IMAGE_ID_IMG_ID_LEN	554
9.4.1.7	FMS_IMAGE_ID_MAX_ENTRIES	554
9.4.1.8	FMS_IMAGE_ID_PRI_IMGTYPE	554
9.4.1.9	FMS_MAX_IMAGE_ID_ELEMENT	554
9.4.1.10	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	554
9.4.2	Function Documentation	554
9.4.2.1	GetValidFwPriCombinations(FMSImageList *pStoredImageList, uint32_t *pValidCombinationSize, CarrierImage_t *pValidCombinations)	554
9.4.2.2	pack_fms_GetImagesPreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_fms_GetImagesPreference_t *reqArg)	554
9.4.2.3	pack_fms_GetStoredImages(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_fms_GetStoredImages_t *reqArg)	555
9.4.2.4	pack_fms_SetImagesPreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_fms_SetImagesPreference_t *reqArg)	555
9.4.2.5	unpack_fms_GetImagesPreference(uint8_t *pResp, uint16_t respLen, unpack_fms_GetImagesPreference_t *pOutput)	555
9.4.2.6	unpack_fms_GetStoredImages(uint8_t *pResp, uint16_t respLen, unpack_fms_GetStoredImages_t *pOutput)	555

9.4.2.7	unpack_fms_SetImagesPreference(uint8_t *pResp, uint16_t respLen, unpack_fms_SetImagesPreference_t *pOutput)	556
9.5	libsdh.h File Reference	556
9.5.1	Detailed Description	557
9.5.2	Macro Definition Documentation	557
9.5.2.1	FIRMWARE_INFO_STRING_SIZE	557
9.5.2.2	IMG_MASK_CLEAR	557
9.5.2.3	IMG_MASK_GENERIC	557
9.5.2.4	IMG_MASK_MDM	557
9.5.2.5	IMG_MASK_PRI	557
9.5.2.6	LIBSDP_CARRIER_PACKAGE_SKU	557
9.5.2.7	LIBSDP_SKU_STRING_LENGTH	557
9.5.3	Typedef Documentation	557
9.5.3.1	libSDP_FirmwareInfo	557
9.5.3.2	libsdplogger	558
9.5.4	Enumeration Type Documentation	558
9.5.4.1	libSDP_Fw_Type	558
9.5.4.2	libSDP_fwdwl_error_codes	559
9.5.4.3	libSDP_Models	559
9.5.5	Function Documentation	559
9.5.5.1	libSDP_BuildImagesPreferenceRequest(libSDP_FirmwareInfo info, pack_fms_SetImagesPreference_t *pack)	559
9.5.5.2	libSDP_CalculateImageMask(unpack_fms_SetImagesPreference_t SetPrefRspFromModem)	560
9.5.5.3	libSDP_CheckValidFirmwareInfo(libSDP_FirmwareInfo info)	560
9.5.5.4	libSDP_DownloadFW(char *pImagePath, char *szTTYPath, int iFWImageType, int image_mask, int iModelFamily)	560
9.5.5.5	libSDP_ExtractFirmwareParametersByPath(char *pImagePath, libSDP_FirmwareInfo *info)	561
9.5.5.6	libSDP_getFileType(char *szPath)	561
9.5.5.7	libSDP_GetModelFamily(char *pModelString)	562
9.5.5.8	libSDP_GetVersion()	562
9.5.5.9	libsdh_set_log_func(libsdplogger func)	562

9.5.5.10	libsdp_SetReadBlockSize(unsigned long IBlockSize)	563
9.6	loc.h File Reference	563
9.6.1	Macro Definition Documentation	566
9.6.1.1	LOC_UINT8_MAX_STRING_SZ	566
9.6.1.2	LOCEVENTMASKBATCHFULLNOTIFICATION	566
9.6.1.3	LOCEVENTMASKENGINESTATE	566
9.6.1.4	LOCEVENTMASKFIXSESSIONSTATE	566
9.6.1.5	LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION	566
9.6.1.6	LOCEVENTMASKGEOFENCEBREACHNOTIFICATION	566
9.6.1.7	LOCEVENTMASKGEOFENCEGENALERT	566
9.6.1.8	LOCEVENTMASKGNSSMEASUREMENTREPORT	566
9.6.1.9	LOCEVENTMASKGNSSSVINFO	566
9.6.1.10	LOCEVENTMASKINJECTPOSITIONREQ	567
9.6.1.11	LOCEVENTMASKINJECTPREDICTEDORBITSREQ	567
9.6.1.12	LOCEVENTMASKINJECTTIMERREQ	567
9.6.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	567
9.6.1.14	LOCEVENTMASKINVALIDVALUE	567
9.6.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	567
9.6.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	567
9.6.1.17	LOCEVENTMASKMOTIONDATACONTROL	567
9.6.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	567
9.6.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	568
9.6.1.20	LOCEVENTMASKNMEA	568
9.6.1.21	LOCEVENTMASKPEDOMETERCONTROL	568
9.6.1.22	LOCEVENTMASKPOSITIONREPORT	568
9.6.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	568
9.6.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	568
9.6.1.25	LOCEVENTMASKTIMESYNCREQ	568
9.6.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	568
9.6.1.27	LOCEVENTMASKWIFIREQ	568

9.6.1.28	MAX_SENSOR_DATA_LEN . . . . .	569
9.6.1.29	MAX_TEMP_DATA_LEN . . . . .	569
9.6.2	Enumeration Type Documentation . . . . .	569
9.6.2.1	anonymous enum . . . . .	569
9.6.3	Function Documentation . . . . .	569
9.6.3.1	pack_loc_DeleteAssistData(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Delete_Assist_Data_t *reqArg) . . . . .	569
9.6.3.2	pack_loc_EventRegister(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_EventRegister_t *reqArg) . . . . .	569
9.6.3.3	pack_loc_SetExtPowerState(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SetExtPowerState_t *reqArg) . . . . .	570
9.6.3.4	pack_loc_SetOperationMode(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SetOperationMode_t *reqArg) . . . . .	570
9.6.3.5	pack_loc_SLQSLOCGetBestAvailPos(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCGetBestAvailPos_t *reqArg) . . . . .	571
9.6.3.6	pack_loc_SLQSLOCInjectPosition(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectPosition_t *reqArg) . . . . .	571
9.6.3.7	pack_loc_SLQSLOCInjectSensorData(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectSensorData_t *reqArg) . . . . .	571
9.6.3.8	pack_loc_SLQSLOCInjectUTCTime(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectUTCTime_t *reqArg) . . . . .	572
9.6.3.9	pack_loc_SLQSLOCSetCradleMountConfig(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCSetCradleMountConfig_t *reqArg) . . . . .	572
9.6.3.10	pack_loc_Start(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Start_t *reqArg) . . . . .	572
9.6.3.11	pack_loc_Stop(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Stop_t *reqArg) . . . . .	573
9.6.3.12	unpack_loc_BestAvailPos_Ind(uint8_t *pResp, uint16_t respLen, unpack_loc_BestAvailPos_Ind_t *pOutput) . . . . .	573
9.6.3.13	unpack_loc_DeleteAssistData(uint8_t *pResp, uint16_t respLen, unpack_loc_Delete_Assist_Data_t *pOutput) . . . . .	574
9.6.3.14	unpack_loc_DeleteAssistData_Ind(uint8_t *pResp, uint16_t respLen, unpack_loc_DeleteAssistData_Ind_t *pOutput) . . . . .	574
9.6.3.15	unpack_loc_EngineState_Ind(uint8_t *pResp, uint16_t respLen, unpack_loc_EngineState_Ind_t *pOutput) . . . . .	574
9.6.3.16	unpack_loc_EventRegister(uint8_t *pResp, uint16_t respLen, unpack_loc_EventRegister_t *pOutput) . . . . .	575

9.6.3.17	<a href="#">unpack_loc_GnssSvInfo_Ind(uint8_t *pResp, uint16_t respLen, unpack_loc_↔ GnssSvInfo_Ind_t *pOutput)</a>	575
9.6.3.18	<a href="#">unpack_loc_PositionRpt_Ind(uint8_t *pResp, uint16_t respLen, unpack_loc_↔ PositionRpt_Ind_t *pOutput)</a>	576
9.6.3.19	<a href="#">unpack_loc_SetExtPowerConfig_Ind(uint8_t *pResp, uint16_t respLen, unpack_↔ _loc_SetExtPowerConfig_Ind_t *pOutput)</a>	576
9.6.3.20	<a href="#">unpack_loc_SetExtPowerState(uint8_t *pResp, uint16_t respLen, unpack_loc_↔ SetExtPowerState_t *pOutput)</a>	576
9.6.3.21	<a href="#">unpack_loc_SetOperationMode(uint8_t *pResp, uint16_t respLen, unpack_loc_↔ _SetOperationMode_t *pOutput)</a>	577
9.6.3.22	<a href="#">unpack_loc_SetOperationMode_Ind(uint8_t *pResp, uint16_t respLen, unpack_↔ _loc_SetOperationMode_Ind_t *pOutput)</a>	577
9.6.3.23	<a href="#">unpack_loc_SLQSLOCGetBestAvailPos(uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCGetBestAvailPos_t *pOutput)</a>	578
9.6.3.24	<a href="#">unpack_loc_SLQSLOCInjectPosition(uint8_t *pResp, uint16_t respLen)</a>	578
9.6.3.25	<a href="#">unpack_loc_SLQSLOCInjectSensorData(uint8_t *pResp, uint16_t respLen)</a>	578
9.6.3.26	<a href="#">unpack_loc_SLQSLOCInjectUTCTime(uint8_t *pResp, uint16_t respLen)</a>	579
9.6.3.27	<a href="#">unpack_loc_SLQSLOCSetCradleMountConfig(uint8_t *pResp, uint16_t respLen)</a>	579
9.6.3.28	<a href="#">unpack_loc_Start(uint8_t *pResp, uint16_t respLen, unpack_loc_Start_t *pOutput)</a>	579
9.6.3.29	<a href="#">unpack_loc_Stop(uint8_t *pResp, uint16_t respLen, unpack_loc_Stop_t *pOutput)</a>	580
9.7	<a href="#">nas.h File Reference</a>	580
9.7.1	<a href="#">Macro Definition Documentation</a>	586
9.7.1.1	<a href="#">NAS_MAX_DESCRIPTION_LENGTH</a>	586
9.7.1.2	<a href="#">NAS_MAX_NUM_NETWORKS</a>	586
9.7.1.3	<a href="#">NAS_MAX_SCC_RX_INFO_INSTANCES</a>	586
9.7.1.4	<a href="#">NAS_OTA_MESSAGE_MAX_BUF_SIZE</a>	586
9.7.1.5	<a href="#">NAS_PLMN_LENGTH</a>	586
9.7.1.6	<a href="#">NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST</a>	586
9.7.2	<a href="#">Enumeration Type Documentation</a>	586
9.7.2.1	<a href="#">LIBPACK_NAS_LTE_CPHY_CA_BW_NRB</a>	586
9.7.2.2	<a href="#">LIBPACK_NAS_LTE_CPHY_SCELL_STATE</a>	586
9.7.2.3	<a href="#">NAS_LTE_CPHY_CA_BW_NRB_LITE</a>	587
9.7.2.4	<a href="#">NAS_LTE_CPHY_SCELL_STATE_LITE</a>	587

9.7.3	Function Documentation	587
9.7.3.1	pack_nas_GetACCOLC(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	587
9.7.3.2	pack_nas_GetANAAAuthenticationStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	587
9.7.3.3	pack_nas_GetCDMANetworkParameters(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	588
9.7.3.4	pack_nas_GetHomeNetwork(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	588
9.7.3.5	pack_nas_GetNetworkPreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	588
9.7.3.6	pack_nas_GetRFInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	589
9.7.3.7	pack_nas_GetServingNetwork(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	589
9.7.3.8	pack_nas_GetServingNetworkCapabilities(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	589
9.7.3.9	pack_nas_GetSignalStrengths(pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen)	590
9.7.3.10	pack_nas_PerformNetworkScan(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	590
9.7.3.11	pack_nas_SetACCOLC(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetACCOLC_t reqParam)	590
9.7.3.12	pack_nas_SetLURRejectCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)	591
9.7.3.13	pack_nas_SetNetworkPreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetNetworkPreference_t *reqArg)	591
9.7.3.14	pack_nas_SetRFInfoCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)	591
9.7.3.15	pack_nas_SLqsGetLTECphyCAInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	592
9.7.3.16	pack_nas_SLQSGetNetworkTime(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	592
9.7.3.17	pack_nas_SLQSGetPLMNName(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSGetPLMNName_t *reqArg)	592
9.7.3.18	pack_nas_SLQSGetServingSystem(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	592
9.7.3.19	pack_nas_SLQSGetSignalStrength(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint16_t reqMask)	593
9.7.3.20	pack_nas_SLQSGetSysInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	593
9.7.3.21	pack_nas_SLQSGetSysSelectionPref(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	593



9.7.3.22	pack_nas_SLQSIInitiateNetworkRegistration(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSIInitiateNetworkRegistration_t *pReqParam)	594
9.7.3.23	pack_nas_SLQSNasConfigSigInfo2(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasConfigSigInfo2_t *pReqParam)	594
9.7.3.24	pack_nas_SLQSNasGetCellLocationInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	594
9.7.3.25	pack_nas_SLQSNasGetSigInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	595
9.7.3.26	pack_nas_SLQSNasIndicationRegisterExt(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasIndicationRegisterExt_t *pReqParam)	595
9.7.3.27	pack_nas_SLQSNasSwiIndicationRegister(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasSwiIndicationRegister_t *pReqParam)	595
9.7.3.28	pack_nas_SLQSNasSwiModemStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	596
9.7.3.29	pack_nas_SLQSSetBandPreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint64_t bandPref)	596
9.7.3.30	pack_nas_SLQSSetSignalStrengthsCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSignalStrengthsCallback_t *pReqParam)	597
9.7.3.31	pack_nas_SLQSSetSysSelectionPref(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSysSelectionPref_t *pReqParam)	597
9.7.3.32	pack_nas_SLQSSwiGetLteCQI(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	597
9.7.3.33	pack_nas_SLQSSwiGetLteSccRxInfo(pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen)	598
9.7.3.34	unpack_nas_GetACCOLC(uint8_t *pResp, uint16_t respLen, uint8_t *pAccolc)	598
9.7.3.35	unpack_nas_GetANAAAAAuthenticationStatus(uint8_t *pResp, uint16_t respLen, uint32_t *pAuthStatus)	598
9.7.3.36	unpack_nas_GetCDMANetworkParameters(uint8_t *pResp, uint16_t respLen, unpack_nas_GetCDMANetworkParameters_t *pOutput)	599
9.7.3.37	unpack_nas_GetHomeNetwork(uint8_t *pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t *pOutput)	599
9.7.3.38	unpack_nas_GetNetworkPreference(uint8_t *pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t *pOutput)	599
9.7.3.39	unpack_nas_GetRFInfo(uint8_t *pResp, uint16_t respLen, unpack_nas_GetRFInfo_t *pOutput)	599
9.7.3.40	unpack_nas_GetServingNetwork(uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t *pOutput)	600
9.7.3.41	unpack_nas_GetServingNetworkCapabilities(uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t *pOutput)	600

9.7.3.42	unpack_nas_GetSignalStrengths(uint8_t *pResp, uint16_t respLen, unpack_nas_GetSignalStrengths_t *pOutput) . . . . .	601
9.7.3.43	unpack_nas_PerformNetworkScan(uint8_t *pResp, uint16_t respLen, unpack_nas_PerformNetworkScan_t *pOutput) . . . . .	601
9.7.3.44	unpack_nas_SetACCOLC(uint8_t *pResp, uint16_t respLen) . . . . .	601
9.7.3.45	unpack_nas_SetDataCapabilitiesCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SetDataCapabilitiesCallback_ind_t *pOutput) . . . . .	602
9.7.3.46	unpack_nas_SetEventReportInd(uint8_t *pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *pOutput) . . . . .	602
9.7.3.47	unpack_nas_SetLURejectCallback(uint8_t *pResp, uint16_t respLen) . . . . .	602
9.7.3.48	unpack_nas_SetNasLTECphyCalIndCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCalIndCallback_ind_t *pOutput) . . . . .	602
9.7.3.49	unpack_nas_SetNetworkPreference(uint8_t *pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t *pOutput) . . . . .	603
9.7.3.50	unpack_nas_SetRFInfoCallback(uint8_t *pResp, uint16_t respLen) . . . . .	603
9.7.3.51	unpack_nas_SetRoamingIndicatorCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t *pOutput) . . . . .	603
9.7.3.52	unpack_nas_SetServingSystemCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t *pOutput) . . . . .	603
9.7.3.53	unpack_nas_SLqsGetLTECphyCAInfo(uint8_t *pResp, uint16_t respLen, unpack_nas_SLqsGetLTECphyCAInfo_t *pOutput) . . . . .	604
9.7.3.54	unpack_nas_SLQSGetNetworkTime(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t *pOutput) . . . . .	604
9.7.3.55	unpack_nas_SLQSGetPLMNName(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t *pOutput) . . . . .	604
9.7.3.56	unpack_nas_SLQSGetServingSystem(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetServingSystem_t *pOutput) . . . . .	604
9.7.3.57	unpack_nas_SLQSGetSignalStrength(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t *pOutput) . . . . .	605
9.7.3.58	unpack_nas_SLQSGetSysInfo(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t *pOutput) . . . . .	605
9.7.3.59	unpack_nas_SLQSGetSysSelectionPref(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSysSelectionPref_t *pOutput) . . . . .	605
9.7.3.60	unpack_nas_SLQSInitiateNetworkRegistration(uint8_t *pResp, uint16_t respLen) . . . . .	606
9.7.3.61	unpack_nas_SLQSNasConfigSigInfo2(uint8_t *pResp, uint16_t respLen) . . . . .	606
9.7.3.62	unpack_nas_SLQSNasGetCellLocationInfo(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetCellLocationInfo_t *pOutput) . . . . .	606

9.7.3.63	unpack_nas_SLQSNasGetSigInfo(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetSigInfo_t *pOutput) . . . . .	607
9.7.3.64	unpack_nas_SLQSNasIndicationRegisterExt(uint8_t *pResp, uint16_t respLen) . . . . .	607
9.7.3.65	unpack_nas_SLQSNasNetworkTimeCallBack_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t *pOutput) . . . . .	607
9.7.3.66	unpack_nas_SLQSNasSigInfoCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSigInfoCallback_ind_t *pOutput) . . . . .	608
9.7.3.67	unpack_nas_SLQSNasSwiIndicationRegister(uint8_t *pResp, uint16_t respLen) . . . . .	608
9.7.3.68	unpack_nas_SLQSNasSwiModemStatus(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiModemStatus_t *pOutput) . . . . .	608
9.7.3.69	unpack_nas_SLQSNasSwiOTAMessageCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t *pOutput) . . . . .	609
9.7.3.70	unpack_nas_SLQSNasSysInfoCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSysInfoCallback_ind_t *pOutput) . . . . .	609
9.7.3.71	unpack_nas_SLQSNasTimerCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasTimerCallback_ind_t *pOutput) . . . . .	609
9.7.3.72	unpack_nas_SLQSSetBandPreference(uint8_t *pResp, uint16_t respLen) . . . . .	610
9.7.3.73	unpack_nas_SLQSSetSignalStrengthsCallback(uint8_t *pResp, uint16_t respLen) . . . . .	610
9.7.3.74	unpack_nas_SLQSSetSysSelectionPref(uint8_t *pResp, uint16_t respLen) . . . . .	610
9.7.3.75	unpack_nas_SLQSSetSysSelectionPrefCallBack_ind(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t *pOutput) . . . . .	611
9.7.3.76	unpack_nas_SLQSSwiGetLteCQI(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteCQI_t *pOutput) . . . . .	611
9.7.3.77	unpack_nas_SLQSSwiGetLteScCRxInfo(uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteScCRxInfo_t *pOutput) . . . . .	611
9.8	qaGobiApiTableBandClasses.h File Reference . . . . .	612
9.8.1	Detailed Description . . . . .	612
9.8.2	Band Classes (Value - Description) . . . . .	612
9.8.2.1	LTE Bands . . . . .	614
9.9	qaGobiApiTableCallControlReturnReasons.h File Reference . . . . .	615
9.9.1	Detailed Description . . . . .	615
9.9.2	Coding Group Bits 7..4(0000) . . . . .	615
9.10	qaGobiApiTableCallEndReasons.h File Reference . . . . .	616
9.10.1	Detailed Description . . . . .	616
9.10.2	Call end reason codes (Code - Reason) . . . . .	616

9.10.2.1	Technology-agnostic call end reasons . . . . .	616
9.10.2.2	CDMA . . . . .	617
9.10.2.3	WCDMA/GSM call end reasons . . . . .	617
9.10.2.4	CDMA . . . . .	619
9.10.2.5	call end reason type . . . . .	620
9.10.2.6	Mobile IP call end reasons (Type=1) . . . . .	620
9.10.2.7	Internal call end reasons (Type=2) . . . . .	622
9.10.2.8	Call Manager defined call end reasons (Type=3) . . . . .	623
9.10.2.9	3GPP specification defined call end reasons (Type=6) . . . . .	628
9.10.2.10	PPP call end reasons (Type=7) . . . . .	630
9.10.2.11	EHRPD call end reasons (Type=8) . . . . .	630
9.10.2.12	IPv6 call end reasons (Type=9) . . . . .	631
9.11	qaGobiApiTableCarrierCodes.h File Reference . . . . .	631
9.11.1	Detailed Description . . . . .	631
9.11.2	Carrier Codes (Number - Carrier) . . . . .	632
9.12	qaGobiApiTableCodingScheme.h File Reference . . . . .	633
9.12.1	Detailed Description . . . . .	633
9.12.2	Coding Group Bits 7..4(0000) . . . . .	634
9.12.2.1	Use of bits 3..0\n\n . . . . .	634
9.12.3	Coding Group Bits 7..4(0001) . . . . .	634
9.12.3.1	use of bits 3..0 . . . . .	634
9.12.4	Coding Group Bits 7..4(0010) . . . . .	634
9.12.4.1	use of bits 3..0 . . . . .	634
9.12.5	Coding Group Bits 7..4(0011) . . . . .	635
9.12.5.1	use of bits 3..0 . . . . .	635
9.12.6	Coding Group Bits 7..4(01xx) . . . . .	635
9.12.6.1	use of bits 3..0 . . . . .	635
9.12.7	Coding Group Bits 7..4(1001) . . . . .	635
9.12.7.1	Reserved coding groups . . . . .	635
9.12.8	Coding Group Bits 7..4(1010..1101) . . . . .	636

9.12.8.1	Reserved coding groups . . . . .	636
9.12.9	Coding Group Bits 7..4(1110) . . . . .	636
9.12.9.1	Defined by the WAP Forum . . . . .	636
9.12.10	Coding Group Bits 7..4 (1111) . . . . .	636
9.12.10.1	Data coding / message handling . . . . .	636
9.12.11	Macro Definition Documentation . . . . .	636
9.12.11.1	__GOBI_API_CODING_SCHEME_H__ . . . . .	636
9.13	qaGobiApiTableGpsCapabilityCodes.h File Reference . . . . .	636
9.13.1	Detailed Description . . . . .	636
9.13.2	GPS capability (Value - Capability) . . . . .	636
9.14	qaGobiApiTablePowerModes.h File Reference . . . . .	637
9.14.1	Detailed Description . . . . .	637
9.14.2	Power Modes (Value - Description) . . . . .	637
9.15	qaGobiApiTableRadioInterfaces.h File Reference . . . . .	637
9.15.1	Detailed Description . . . . .	637
9.15.2	Radio interface . . . . .	638
9.15.2.1	Technology (Value - Radio Interface Technology) . . . . .	638
9.16	qaGobiApiTableRegionCodes.h File Reference . . . . .	638
9.16.1	Detailed Description . . . . .	638
9.16.2	Region Codes (Code - Region) . . . . .	638
9.17	qaGobiApiTableServiceOptions.h File Reference . . . . .	638
9.17.1	Detailed Description . . . . .	639
9.17.2	Service Option codes (Code - Reason) . . . . .	639
9.17.2.1	Description . . . . .	639
9.18	qaGobiApiTableSupServiceInfoClasses.h File Reference . . . . .	641
9.18.1	Detailed Description . . . . .	641
9.18.2	Supplementary Service Information Classes (Value - Service Class) . . . . .	641
9.19	qaGobiApiTableSwiAudio.h File Reference . . . . .	641
9.19.1	Detailed Description . . . . .	641
9.19.2	ACDB Device (Device ID - description) . . . . .	642

9.19.3	Physical Interface (Device ID - description - Interface parameters)	642
9.20	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	642
9.20.1	Detailed Description	642
9.20.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	642
9.21	qaGobiApiTableVoiceCallEndReasons.h File Reference	643
9.21.1	Detailed Description	644
9.21.2	Voice Call and supplementary services end reason codes (Code - Reason)	644
9.21.2.1	General	644
9.21.2.2	service Errors	645
9.21.2.3	control cause values	646
9.21.2.4	reject causes	648
9.21.2.5	reject causes	649
9.21.2.6	reject causes	649
9.21.2.7	stratum reject causes	649
9.21.2.8	reject causes	649
9.21.2.9	IP end reasons	649
9.22	qmerrno.h File Reference	650
9.22.1	Enumeration Type Documentation	652
9.22.1.1	eQCWWANError	652
9.22.1.2	qm_wds_ds_profile_extended_err_codes	657
9.23	qos.h File Reference	657
9.23.1	Macro Definition Documentation	659
9.23.1.1	LIBPACK_MAX_QOS_FILTERS	659
9.23.1.2	LIBPACK_MAX_QOS_FLOW_PER_APN_STATS	659
9.23.1.3	LIBPACK_MAX_QOS_FLOWS	659
9.23.2	Function Documentation	659
9.23.2.1	pack_qos_SLQSQosGetNetworkStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	659
9.23.2.2	pack_qos_SLQSQosSwiReadApnExtraParams(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_qos_SLQSQosSwiReadApnExtraParams_t reqParam)	659

9.23.2.3	<a href="#">pack_qos_SLQSQosSwiReadDataStats(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_qos_SLQSQosSwiReadDataStats_t reqParam)</a>	660
9.23.2.4	<a href="#">pack_qos_SLQSSetQosEventCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_qos_SLQSSetQosEventCallback_t reqParam)</a>	661
9.23.2.5	<a href="#">unpack_qos_SLQSQosGetNetworkStatus(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSQosGetNetworkStatus_t *pOutput)</a>	662
9.23.2.6	<a href="#">unpack_qos_SLQSQosSwiReadApnExtraParams(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSQosSwiReadApnExtraParams_t *pOutput)</a>	662
9.23.2.7	<a href="#">unpack_qos_SLQSQosSwiReadDataStats(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSQosSwiReadDataStats_t *pOutput)</a>	663
9.23.2.8	<a href="#">unpack_qos_SLQSSetQosEventCallback(uint8_t *pResp, uint16_t respLen)</a>	663
9.23.2.9	<a href="#">unpack_qos_SLQSSetQosEventCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSSetQosEventCallback_ind_t *pOutput)</a>	663
9.23.2.10	<a href="#">unpack_qos_SLQSSetQosNWStatusCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSSetQosNWStatusCallback_ind_t *pOutput)</a>	664
9.23.2.11	<a href="#">unpack_qos_SLQSSetQosPriEventCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSSetQosPriEventCallback_ind_t *pOutput)</a>	665
9.23.2.12	<a href="#">unpack_qos_SLQSSetQosStatusCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSSetQosStatusCallback_ind_t *pOutput)</a>	665
9.24	<a href="#">sms.h File Reference</a>	666
9.24.1	<a href="#">Macro Definition Documentation</a>	668
9.24.1.1	<a href="#">MAX_CDMA_ENC_MO_TXT_MSG_SIZE</a>	668
9.24.1.2	<a href="#">MAX_MS_TRANSFER_ROUTE_MSG</a>	668
9.24.1.3	<a href="#">MAX_MSC_ADDRESS_SIZE</a>	668
9.24.1.4	<a href="#">MAX_MSE_TWS_MSG</a>	668
9.24.1.5	<a href="#">MAX_SMS_LIST_SIZE</a>	668
9.24.1.6	<a href="#">MAX_SMS_MESSAGE_SIZE</a>	668
9.24.2	<a href="#">Typedef Documentation</a>	668
9.24.2.1	<a href="#">sMSCAddressInfo</a>	668
9.24.2.2	<a href="#">sMSEtwSMessageInfo</a>	668
9.24.2.3	<a href="#">sMSEtwSPImnInfo</a>	669
9.24.2.4	<a href="#">sSMSMessageModelInfo</a>	669
9.24.2.5	<a href="#">sMSMTMessageInfo</a>	669
9.24.2.6	<a href="#">sMSOnIMSInfo</a>	669

9.24.2.7	smSTransferRouteMTMessageInfo . . . . .	669
9.24.3	Enumeration Type Documentation . . . . .	670
9.24.3.1	eqmiCbkSetStatus . . . . .	670
9.24.4	Function Documentation . . . . .	670
9.24.4.1	pack_sms_SendSMS(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SendSMS_t *reqParam) . . . . .	670
9.24.4.2	pack_sms_SetNewSMSCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SetNewSMSCallback_t reqParam) . . . . .	671
9.24.4.3	pack_sms_SLQSDDeleteSMS(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *p↔ Len, pack_sms_SLQSDDeleteSMS_t *reqParam) . . . . .	671
9.24.4.4	pack_sms_SLQSGetSMS(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSMS_t *reqParam) . . . . .	671
9.24.4.5	pack_sms_SLQSGetSMSList(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t↔ t *pLen, pack_sms_SLQSGetSMSList_t *reqParam) . . . . .	672
9.24.4.6	pack_sms_SLQSModifySMSSStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSModifySMSSStatus_t *reqParam) . . . . .	672
9.24.4.7	unpack_sms_SendSMS(uint8_t *pResp, uint16_t respLen, unpack_sms_Send↔ SMS_t *pOutput) . . . . .	673
9.24.4.8	unpack_sms_SetNewSMSCallback(uint8_t *pResp, uint16_t respLen, unpack↔ _sms_SetNewSMSCallback_t *pOutput) . . . . .	673
9.24.4.9	unpack_sms_SetNewSMSCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_ind_t *pOutput) . . . . .	673
9.24.4.10	unpack_sms_SLQSDDeleteSMS(uint8_t *pResp, uint16_t respLen, unpack_↔ sms_SLQSDDeleteSMS_t *pOutput) . . . . .	674
9.24.4.11	unpack_sms_SLQSGetSMS(uint8_t *pResp, uint16_t respLen, unpack_sms_↔ SLQSGetSMS_t *pOutput) . . . . .	674
9.24.4.12	unpack_sms_SLQSGetSMSList(uint8_t *pResp, uint16_t respLen, unpack_↔ sms_SLQSGetSMSList_t *pOutput) . . . . .	674
9.24.4.13	unpack_sms_SLQSModifySMSSStatus(uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSModifySMSSStatus_t *pOutput) . . . . .	675
9.24.4.14	unpack_sms_SLQSWmsMemoryFullCallBack_ind(uint8_t *pResp, uint16_t↔ t respLen, unpack_sms_SLQSWmsMemoryFullCallBack_ind_t *pOutput) . . . .	675
9.25	SwiDataTypes.h File Reference . . . . .	676
9.25.1	Detailed Description . . . . .	676
9.25.2	Macro Definition Documentation . . . . .	676
9.25.2.1	QMI_NO_LTE_FW_SUPPORT . . . . .	676
9.25.2.2	QMI_TLV_PLACEHOLDER . . . . .	676



9.25.2.3	SWI_API	676
9.25.2.4	UNUSEDPARAM	676
9.25.3	Typedef Documentation	677
9.25.3.1	BOOL	677
9.25.3.2	BYTE	677
9.25.3.3	CHAR	677
9.25.3.4	FLOAT	677
9.25.3.5	INT32	677
9.25.3.6	INT8	677
9.25.3.7	LPCSTR	677
9.25.3.8	SHORT	677
9.25.3.9	ULONG	677
9.25.3.10	ULONGLONG	677
9.25.3.11	USHORT	677
9.25.3.12	WORD	677
9.26	swiloc.h File Reference	677
9.26.1	Function Documentation	677
9.26.1.1	pack_swiloc_SwiLocGetAutoStart(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	677
9.26.1.2	pack_swiloc_SwiLocSetAutoStart(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiloc_SwiLocSetAutoStart_t *reqArg)	678
9.26.1.3	unpack_swiloc_SwiLocGetAutoStart(uint8_t *pResp, uint16_t respLen, unpack_swiloc_SwiLocGetAutoStart_t *pOutput)	678
9.26.1.4	unpack_swiloc_SwiLocSetAutoStart(uint8_t *pResp, uint16_t respLen)	679
9.27	swioma.h File Reference	679
9.27.1	Macro Definition Documentation	680
9.27.1.1	LIBPACK_MAX_SWIOMA_STR_LEN	680
9.27.2	Function Documentation	680
9.27.2.1	pack_swioma_SLQSOMADMAAlertCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	680
9.27.2.2	pack_swioma_SLQSOMADMCancelSession(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swioma_SLQSOMADMCancelSession_t reqParam)	681

9.27.2.3	pack_swima_SLQSOMADMGetSessionInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swima_SLQSOMADMGetSessionInfo_t reqParam)	682
9.27.2.4	pack_swima_SLQSOMADMGetSettings(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	682
9.27.2.5	pack_swima_SLQSOMADMSendSelection(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swima_SLQSOMADMSendSelection_t reqParam)	683
9.27.2.6	pack_swima_SLQSOMADMSetSettings(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swima_SLQSOMADMSetSettings_t reqParam)	683
9.27.2.7	pack_swima_SLQSOMADMStartSession(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swima_SLQSOMADMStartSession_t reqParam)	684
9.27.2.8	unpack_swima_SLQSOMADMAAlertCallback(uint8_t *pResp, uint16_t respLen)	685
9.27.2.9	unpack_swima_SLQSOMADMAAlertCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_swima_SLQSOMADMAAlertCallback_ind_t *pOutput)	685
9.27.2.10	unpack_swima_SLQSOMADMCancelSession(uint8_t *pResp, uint16_t respLen)	686
9.27.2.11	unpack_swima_SLQSOMADMGetSessionInfo(uint8_t *pResp, uint16_t respLen, unpack_swima_SLQSOMADMGetSessionInfo_t *pOutput)	686
9.27.2.12	unpack_swima_SLQSOMADMGetSettings(uint8_t *pResp, uint16_t respLen, unpack_swima_SLQSOMADMGetSettings_t *pOutput)	687
9.27.2.13	unpack_swima_SLQSOMADMSendSelection(uint8_t *pResp, uint16_t respLen)	687
9.27.2.14	unpack_swima_SLQSOMADMSetSettings(uint8_t *pResp, uint16_t respLen)	688
9.27.2.15	unpack_swima_SLQSOMADMStartSession(uint8_t *pResp, uint16_t respLen, unpack_swima_SLQSOMADMStartSession_t *pOutput)	688
9.28	SWIWWANCMAPI.h File Reference	689
9.29	uim.h File Reference	689
9.29.1	Macro Definition Documentation	690
9.29.1.1	MAX_DESCRIPTION_LENGTH	690
9.29.1.2	MAX_ICCID_LENGTH	690
9.29.1.3	MAX_NO_OF_APPLICATIONS	690
9.29.1.4	MAX_NO_OF_SLOTS	690
9.29.1.5	MAX_SLOTS_STATUS	691
9.29.1.6	UIM_MAX_DESCRIPTION_LENGTH	691
9.29.1.7	UIM_MAX_NO_OF_APPLICATIONS	691
9.29.1.8	UIM_MAX_NO_OF_SLOTS	691
9.29.1.9	UIM_UINT8_MAX_STRING_SZ	691

9.29.2	Function Documentation	691
9.29.2.1	pack_uim_ChangePin(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_ChangePin_t *reqArg)	691
9.29.2.2	pack_uim_GetCardStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)	691
9.29.2.3	pack_uim_ReadTransparent(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_ReadTransparent_t *reqArg)	692
9.29.2.4	pack_uim_SetPinProtection(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SetPinProtection_t *reqArg)	692
9.29.2.5	pack_uim_SLQSUIEventRegister(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIEventRegister_t *reqArg)	692
9.29.2.6	pack_uim_SLQSUIGetSlotsStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	693
9.29.2.7	pack_uim_SLQSUIPowerDown(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIPowerDown_t *reqArg)	693
9.29.2.8	pack_uim_SLQSUIPowerUp(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIPowerUp_t *reqArg)	694
9.29.2.9	pack_uim_SLQSUISwitchSlot(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUISwitchSlot_t *reqArg)	694
9.29.2.10	pack_uim_UnblockPin(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_UnblockPin_t *reqArg)	694
9.29.2.11	pack_uim_VerifyPin(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_VerifyPin_t *reqArg)	695
9.29.2.12	unpack_uim_ChangePin(uint8_t *pResp, uint16_t respLen, unpack_uim_ChangePin_t *pOutput)	695
9.29.2.13	unpack_uim_GetCardStatus(uint8_t *pResp, uint16_t respLen, unpack_uim_GetCardStatus_t *pOutput)	696
9.29.2.14	unpack_uim_ReadTransparent(uint8_t *pResp, uint16_t respLen, unpack_uim_ReadTransparent_t *pOutput)	696
9.29.2.15	unpack_uim_SetPinProtection(uint8_t *pResp, uint16_t respLen, unpack_uim_SetPinProtection_t *pOutput)	696
9.29.2.16	unpack_uim_SetUimSlotStatusChangeCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_uim_SetUimSlotStatusChangeCallback_ind_t *pOutput)	697
9.29.2.17	unpack_uim_SLQSUIEventRegister(uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIEventRegister_t *pOutput)	697
9.29.2.18	unpack_uim_SLQSUIGetSlotsStatus(uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIGetSlotsStatus_t *pOutput)	698
9.29.2.19	unpack_uim_SLQSUIPowerDown(uint8_t *pResp, uint16_t respLen)	698
9.29.2.20	unpack_uim_SLQSUIPowerUp(uint8_t *pResp, uint16_t respLen)	698

9.29.2.21	unpack_uim_SLQSUIMSetStatusChangeCallBack_ind(uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t *pOutput)	699
9.29.2.22	unpack_uim_SLQSUIMSwitchSlot(uint8_t *pResp, uint16_t respLen)	699
9.29.2.23	unpack_uim_UnblockPin(uint8_t *pResp, uint16_t respLen, unpack_uim_UnblockPin_t *pOutput)	700
9.29.2.24	unpack_uim_VerifyPin(uint8_t *pResp, uint16_t respLen, unpack_uim_VerifyPin_t *pOutput)	700
9.30	wds.h File Reference	700
9.30.1	Macro Definition Documentation	706
9.30.1.1	BYT_STAT_STAT_MASK	706
9.30.1.2	IPV6_ADDRESS_ARRAY_SIZE	706
9.30.1.3	MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	706
9.30.1.4	PACK_WDS_IPV4	706
9.30.1.5	PACK_WDS_IPV6	706
9.30.1.6	WDS_DHCP_MAX_NUM_OPTIONS	706
9.30.1.7	WDS_DHCP_OPTION_DATA_BUF_SIZE	706
9.30.2	Typedef Documentation	706
9.30.2.1	UnpackQmiProfileInfo	706
9.30.3	Function Documentation	706
9.30.3.1	pack_wds_DHCPv4ClientLeaseChange(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_DHCPv4ClientLeaseChange_t *reqArg)	706
9.30.3.2	pack_wds_GetAutoconnect(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	707
9.30.3.3	pack_wds_GetByteTotals(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	707
9.30.3.4	pack_wds_GetConnectionRate(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	708
9.30.3.5	pack_wds_GetDataBearerTechnology(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)	708
9.30.3.6	pack_wds_GetDefaultProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfile_t *reqParam)	709
9.30.3.7	pack_wds_GetDefaultProfileNum(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfileNum_t *reqParam)	709
9.30.3.8	pack_wds_GetDormancyState(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDormancyState_t *reqParam)	709
9.30.3.9	pack_wds_GetLastMobileIPError(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetLastMobileIPError_t *reqParam)	710

9.30.3.10 pack_wds_GetMobileIP(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIP_t *pReqParam) . . . . .	710
9.30.3.11 pack_wds_GetMobileIPProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIPProfile_t *reqParam) . . . . .	711
9.30.3.12 pack_wds_GetPacketStatistics(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatistics_t *pReq) . . . . .	711
9.30.3.13 pack_wds_GetPacketStatus(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatus_t *reqParam) . . . . .	712
9.30.3.14 pack_wds_GetSessionDuration(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetSessionDuration_t *reqParam) . . . . .	712
9.30.3.15 pack_wds_GetSessionState(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen) . . . . .	713
9.30.3.16 pack_wds_RMSetTransferStatistics(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_RMSetTransferStatistics_t *reqParam) . . . . .	713
9.30.3.17 pack_wds_SetAutoconnect(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetAutoconnect_t *reqArg) . . . . .	714
9.30.3.18 pack_wds_SetDefaultProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfile_t *reqParam) . . . . .	714
9.30.3.19 pack_wds_SetDefaultProfileNum(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfileNum_t *pReqParam) . . . . .	715
9.30.3.20 pack_wds_SetMobileIP(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIP_t *reqArg) . . . . .	715
9.30.3.21 pack_wds_SetMobileIPParameters(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPParameters_t *reqArg) . . . . .	715
9.30.3.22 pack_wds_SetMobileIPProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPProfile_t *reqParam) . . . . .	716
9.30.3.23 pack_wds_SLQSCreateProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSCreateProfile_t *reqArg) . . . . .	716
9.30.3.24 pack_wds_SLQSDeleteProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSDeleteProfile_t *reqParam) . . . . .	717
9.30.3.25 pack_wds_SLQSGet3GPPConfigItem(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen) . . . . .	717
9.30.3.26 pack_wds_SLQSGetCurrDataSystemStat(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetCurrDataSystemStat_t *pReqParam) . . . . .	718
9.30.3.27 pack_wds_SLQSGetCurrentChannelRate(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen) . . . . .	718
9.30.3.28 pack_wds_SLQSGetDataBearerTechnology(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDataBearerTechnology_t *pReqParam) . . . . .	719
9.30.3.29 pack_wds_SLQSGetDUNCallInfo(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDUNCallInfo_t *reqParam) . . . . .	719

9.30.3.30	<code>pack_wds_SLQSGetProfileSettings(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetProfileSettings_t *reqArg)</code>	720
9.30.3.31	<code>pack_wds_SLQSGetRuntimeSettings(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetRuntimeSettings_t *reqArg)</code>	720
9.30.3.32	<code>pack_wds_SLQSModifyProfile(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSModifyProfile_t *reqArg)</code>	720
9.30.3.33	<code>pack_wds_SLQSResetPacketStatics(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	721
9.30.3.34	<code>pack_wds_SLQSSet3GPPConfigItem(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSet3GPPConfigItem_t *reqParam)</code>	721
9.30.3.35	<code>pack_wds_SLQSSetIPFamilyPreference(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetIPFamilyPreference_t *pReqParam)</code>	722
9.30.3.36	<code>pack_wds_SLQSSetWdsEventCallback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetWdsEventCallback_t *reqArg)</code>	722
9.30.3.37	<code>pack_wds_SLQSSetDHCPv4ClientConfig(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetDHCPv4ClientConfig_t *pReq)</code>	723
9.30.3.38	<code>pack_wds_SLQSSetLoopback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	723
9.30.3.39	<code>pack_wds_SLQSSetDHCPv4ClientConfig(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetDHCPv4ClientConfig_t *reqArg)</code>	724
9.30.3.40	<code>pack_wds_SLQSSetLoopback(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetLoopback_t *reqArg)</code>	724
9.30.3.41	<code>pack_wds_SLQSStartDataSession(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSStartDataSession_t *reqArg)</code>	725
9.30.3.42	<code>pack_wds_SLQSStopDataSession(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSStopDataSession_t *reqArg)</code>	725
9.30.3.43	<code>pack_wds_SLQSWdsGoActive(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	726
9.30.3.44	<code>pack_wds_SLQSWdsGoDormant(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)</code>	726
9.30.3.45	<code>pack_wds_SLQSWdsSetEventReport(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSetEventReport_t *reqArg)</code>	726
9.30.3.46	<code>pack_wds_SLQSWdsSwiPDPRuntimeSettings(pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t *reqParam)</code>	727
9.30.3.47	<code>unpack_wds_DHCPv4ClientLease_ind(uint8_t *pResp, uint16_t respLen, unpack_wds_DHCPv4ClientLease_ind_t *pOutput)</code>	727
9.30.3.48	<code>unpack_wds_DHCPv4ClientLeaseChange(uint8_t *pResp, uint16_t respLen)</code>	728
9.30.3.49	<code>unpack_wds_GetAutoconnect(uint8_t *pResp, uint16_t respLen, unpack_wds_GetAutoconnect_t *pOutput)</code>	728

9.30.3.50 unpack_wds_GetByteTotals(uint8_t *pResp, uint16_t respLen, unpack_wds_↔ GetByteTotals_t *pOutput) . . . . .	729
9.30.3.51 unpack_wds_GetConnectionRate(uint8_t *pResp, uint16_t respLen, unpack_↔ wds_GetConnectionRate_t *pOutput) . . . . .	729
9.30.3.52 unpack_wds_GetDataBearerTechnology(uint8_t *pResp, uint16_t respLen, unpack_wds_GetDataBearerTechnology_t *pOutput) . . . . .	729
9.30.3.53 unpack_wds_GetDefaultProfile(uint8_t *pResp, uint16_t respLen, unpack_wds_↔ _GetDefaultProfile_t *pOutput) . . . . .	730
9.30.3.54 unpack_wds_GetDefaultProfileNum(uint8_t *pResp, uint16_t respLen, unpack_↔ _wds_GetDefaultProfileNum_t *pOutput) . . . . .	730
9.30.3.55 unpack_wds_GetDormancyState(uint8_t *pResp, uint16_t respLen, unpack_↔ wds_GetDormancyState_t *pOutput) . . . . .	730
9.30.3.56 unpack_wds_GetLastMobileIPError(uint8_t *pResp, uint16_t respLen, unpack_↔ _wds_GetLastMobileIPError_t *pOutput) . . . . .	731
9.30.3.57 unpack_wds_GetMobileIP(uint8_t *pResp, uint16_t respLen, unpack_wds_Get_↔ MobileIP_t *pOutput) . . . . .	731
9.30.3.58 unpack_wds_GetMobileIPProfile(uint8_t *pResp, uint16_t respLen, unpack_↔ wds_GetMobileIPProfile_t *pOutput) . . . . .	732
9.30.3.59 unpack_wds_GetPacketStatistics(uint8_t *pResp, uint16_t respLen, unpack_↔ wds_GetPacketStatistics_t *pOutput) . . . . .	732
9.30.3.60 unpack_wds_GetPacketStatus(uint8_t *pResp, uint16_t respLen, unpack_wds_↔ _GetPacketStatus_t *pOutput) . . . . .	732
9.30.3.61 unpack_wds_GetSessionDuration(uint8_t *pResp, uint16_t respLen, unpack_↔ wds_GetSessionDuration_t *pOutput) . . . . .	733
9.30.3.62 unpack_wds_GetSessionState(uint8_t *pResp, uint16_t respLen, unpack_wds_↔ _GetSessionState_t *pOutput) . . . . .	733
9.30.3.63 unpack_wds_RMSetTransferStatistics(uint8_t *pResp, uint16_t respLen, unpack_wds_RMSetTransferStatistics_t *pOutput) . . . . .	734
9.30.3.64 unpack_wds_RMTransferStatistics_ind(uint8_t *pResp, uint16_t respLen, unpack_RMTransferStatistics_ind_t *pOutput) . . . . .	734
9.30.3.65 unpack_wds_SetAutoconnect(uint8_t *pResp, uint16_t respLen) . . . . .	734
9.30.3.66 unpack_wds_SetDefaultProfile(uint8_t *pResp, uint16_t respLen) . . . . .	735
9.30.3.67 unpack_wds_SetDefaultProfileNum(uint8_t *pResp, uint16_t respLen) . . . . .	735
9.30.3.68 unpack_wds_SetMobileIP(uint8_t *pResp, uint16_t respLen) . . . . .	736
9.30.3.69 unpack_wds_SetMobileIPParameters(uint8_t *pResp, uint16_t respLen) . . . . .	736
9.30.3.70 unpack_wds_SetMobileIPProfile(uint8_t *pResp, uint16_t respLen, unpack_↔ wds_SetMobileIPProfile_t *pOutput) . . . . .	736

9.30.3.71	unpack_wds_SLQSCreateProfile(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSCreateProfile_t *pOutput)	737
9.30.3.72	unpack_wds_SLQSDeleteProfile(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSDeleteProfile_t *pOutput)	737
9.30.3.73	unpack_wds_SLQSGet3GPPConfigItem(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGet3GPPConfigItem_t *pOutput)	737
9.30.3.74	unpack_wds_SLQSGetCurrDataSystemStat(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrDataSystemStat_t *pOutput)	738
9.30.3.75	unpack_wds_SLQSGetCurrentChannelRate(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrentChannelRate_t *pOutput)	738
9.30.3.76	unpack_wds_SLQSGetDataBearerTechnology(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDataBearerTechnology_t *pOutput)	739
9.30.3.77	unpack_wds_SLQSGetDUNCallInfo(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDUNCallInfo_t *pOutput)	739
9.30.3.78	unpack_wds_SLQSGetProfileSettings(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetProfileSettings_t *pOutput)	739
9.30.3.79	unpack_wds_SLQSGetRuntimeSettings(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetRuntimeSettings_t *pOutput)	740
9.30.3.80	unpack_wds_SLQSModifyProfile(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSModifyProfile_t *pOutput)	740
9.30.3.81	unpack_wds_SLQSResetPacketStatics(uint8_t *pResp, uint16_t respLen)	740
9.30.3.82	unpack_wds_SLQSSet3GPPConfigItem(uint8_t *pResp, uint16_t respLen)	741
9.30.3.83	unpack_wds_SLQSSetIPFamilyPreference(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetIPFamilyPreference_t *pOutput)	741
9.30.3.84	unpack_wds_SLQSSetPacketSrvStatusCallback(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetPacketSrvStatusCallback_t *pOutput)	741
9.30.3.85	unpack_wds_SLQSSetWdsEventCallback(uint8_t *pResp, uint16_t respLen)	742
9.30.3.86	unpack_wds_SLQSSetWdsEventCallback_ind(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetWdsEventCallback_ind_t *pOutput)	742
9.30.3.87	unpack_wds_SLQSSetDHCPv4ClientConfig(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetDHCPv4ClientConfig_t *pOutput)	743
9.30.3.88	unpack_wds_SLQSSetLoopback(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetLoopback_t *pOutput)	743
9.30.3.89	unpack_wds_SLQSSetDHCPv4ClientConfig(uint8_t *pResp, uint16_t respLen)	743
9.30.3.90	unpack_wds_SLQSSetLoopback(uint8_t *pResp, uint16_t respLen)	744
9.30.3.91	unpack_wds_SLQSStartDataSession(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSStartDataSession_t *pOutput)	744
9.30.3.92	unpack_wds_SLQSStopDataSession(uint8_t *pResp, uint16_t respLen)	744
9.30.3.93	unpack_wds_SLQSWdsGoActive(uint8_t *pResp, uint16_t respLen)	745
9.30.3.94	unpack_wds_SLQSWdsGoDormant(uint8_t *pResp, uint16_t respLen)	745
9.30.3.95	unpack_wds_SLQSWdsSetEventReport(uint8_t *pResp, uint16_t respLen)	746
9.30.3.96	unpack_wds_SLQSWdsSwiPDPRuntimeSettings(uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t *pOutput)	746



## 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) . . . . .	<a href="#">19</a>
Streaming Download Protocol (sdp) . . . . .	<a href="#">20</a>



## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Tables</a> . . . . .	21
----------------------------------	----



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_libSDP_FirmwareInfo_ . . . . .</a>	<a href="#">23</a>
<a href="#">altSrcInfo_t . . . . .</a>	<a href="#">24</a>
<a href="#">appStats . . . . .</a>	<a href="#">25</a>
<a href="#">CarrierImage_t . . . . .</a>	<a href="#">28</a>
<a href="#">cdmaSSInfo . . . . .</a>	<a href="#">29</a>
<a href="#">connectionStatus . . . . .</a>	<a href="#">30</a>
<a href="#">crashInfoParams . . . . .</a>	<a href="#">30</a>
<a href="#">crashInformation . . . . .</a>	<a href="#">31</a>
<a href="#">currNetworkInfo . . . . .</a>	<a href="#">32</a>
<a href="#">dms_ActivationStatusTlv . . . . .</a>	<a href="#">32</a>
<a href="#">dms_OperatingModeTlv . . . . .</a>	<a href="#">33</a>
<a href="#">DMScustSettingInfo . . . . .</a>	<a href="#">34</a>
<a href="#">DMScustSettingList . . . . .</a>	<a href="#">35</a>
<a href="#">DMSgetCustomFeatureV2 . . . . .</a>	<a href="#">35</a>
<a href="#">DMSgetCustomInput . . . . .</a>	<a href="#">36</a>
<a href="#">dunchannelRate . . . . .</a>	<a href="#">36</a>
<a href="#">eriDataparams . . . . .</a>	<a href="#">37</a>
<a href="#">eTWSPLMNInfoTlv . . . . .</a>	<a href="#">37</a>
<a href="#">FMSImageElement . . . . .</a>	<a href="#">38</a>
<a href="#">FMSImageIdElement . . . . .</a>	<a href="#">39</a>
<a href="#">FMSImageIdEntries . . . . .</a>	<a href="#">40</a>
<a href="#">FMSImageList . . . . .</a>	<a href="#">41</a>
<a href="#">FMSPrefImageList . . . . .</a>	<a href="#">41</a>
<a href="#">hdrSSInfo . . . . .</a>	<a href="#">42</a>
<a href="#">image_info_t . . . . .</a>	<a href="#">42</a>
<a href="#">ipv6AddressInfo . . . . .</a>	<a href="#">43</a>
<a href="#">LibPackGPRSRequestedQoS . . . . .</a>	<a href="#">43</a>
<a href="#">LibpackProfile3GPP . . . . .</a>	<a href="#">44</a>
<a href="#">LibpackProfile3GPP2 . . . . .</a>	<a href="#">50</a>
<a href="#">LibPackprofile_3GPP . . . . .</a>	<a href="#">56</a>
<a href="#">LibPackprofile_3GPP2 . . . . .</a>	<a href="#">62</a>
<a href="#">LibPackQosClassID . . . . .</a>	<a href="#">67</a>
<a href="#">LibPackTFTIDParams . . . . .</a>	<a href="#">68</a>
<a href="#">LibPackUMTSQoS . . . . .</a>	<a href="#">70</a>
<a href="#">LibPackUMTSReqQoSSigInd . . . . .</a>	<a href="#">72</a>

loc_BdsSV	73
loc_BdsSVInfo	74
loc_CellDb	74
loc_ClkInfo	75
loc_GnssData	76
loc_gpsTime	78
loc_LocApplicationInfo	78
loc_precisionDilution	79
loc_satelliteInfo	80
loc_sensorDataUsage	82
loc_SV	83
loc_SVInfo	84
loc_svUsedforFix	85
lteSSInfo	85
messageModeTlv	86
nas_acqOrderPref	86
nas_AddCDMASysInfo	87
nas_AddSysInfo	88
nas_CallBarringSysInfo	88
nas_callBarStatus	89
nas_CDMAECIOThresh	90
nas_CDMAInfo	91
nas_CDMARSSIThresh	92
nas_CDMASysInfo	92
nas_CDMASysInfoExt	96
nas_cellParams	97
nas_CommInfo	98
nas_CSGID	99
nas_currentPLMN	100
nas_dataSrvCapabilities	101
nas_detailSvcInfo	102
nas_ecioListElement	104
nas_errorRateListElement	104
nas_GERANInfo	105
nas_geranInstInfo	107
nas_gsmCellInfo	108
nas_GSMRSSIThresh	109
nas_GSMSrvStatusInfo	109
nas_GSMSysInfo	110
nas_HDRECIOThresh	113
nas_HDRIOThresh	114
nas_HDRRSSIThresh	114
nas_HDRSINRThreshold	115
nas_HDRSysInfo	116
nas_infoInterFreq	118
nas_lteGsmCellInfo	119
nas_LTEInfo	121
nas_LTEInfoInterfreq	123
nas_LTEInfoIntrafreq	123
nas_LTEInfoNeighboringGSM	126
nas_LTEInfoNeighboringWCDMA	126
nas_lteRsrpInformation	127
nas_LTERSRRPThresh	128
nas_LTERSRRQThresh	128
nas_LTERSSIThresh	129
nas_LTESigRptConfig	129
nas_lteSnrinformation	130
nas_LTESNRThreshold	131



nas_LTESysInfo	131
nas_lteWcdmaCellInfo	134
nas_MNRInfo	135
nas_netSelectionPref	136
nas_nmrCellInfo	136
nas_PhyCaAggPcellInfo	138
nas_PhyCaAggScellDIBw	139
nas_PhyCaAggScellIndex	139
nas_PhyCaAggScellIndType	140
nas_PhyCaAggScellInfo	141
nas_qaQmi3Gpp2TimeZone	143
nas_QmiNas3GppNetworkInfo	144
nas_QmiNas3GppNetworkRAT	145
nas_QmisNasPcsDigit	145
nas_RejectReasonTlv	146
nas_RFInfoTlv	147
nas_roamIndList	147
nas_rsrqInformation	148
nas_RxSigInfo	149
nas_rxSignalStrengthListElement	150
nas_SccRxInfo	151
nas_servSystem	152
nas_SignalStrengthTlv	153
nas_SLQSSignalStrengthsIndReq	154
nas_SLQSSignalStrengthsInformation	155
nas_SLQSSignalStrengthsTlv	156
nas_SrvStatusInfo	156
nas_sysInfoCommon	157
nas_TDSCDMAECIOThresh	160
nas_TDSCDMARSCPTthresh	160
nas_TDSCDMARSSIThresh	161
nas_TDSCDMASINRThresh	161
nas_timeInfo	162
nas_UMTSInfo	164
nas_UMTSinstInfo	166
nas_umtsLTENbrCell	167
nas_UniversalTime	168
nas_wcdmaCellInfo	169
nas_WCDMAECIOThresh	170
nas_WCDMAInfoLTENeighborCell	171
nas_WCDMARSSIThresh	172
nas_WCDMASysInfo	172
NASBandPreferenceTlv	176
NASEmergencyModeTlv	176
NasGetLTECphyCalInfo	177
NASGWAacqOrderPrefTlv	177
NASLTEBandPreferenceTlv	177
NASLteNasReleaseInfoTlv	178
NASModePreferenceTlv	178
NASNetSelPreferenceTlv	178
NASOTAMessageTlv	179
NASPhyCaAggPcellInfo	179
NASPhyCaAggScellDIBw	180
NASPhyCaAggScellIndex	181
NASPhyCaAggScellIndType	181
NASPhyCaAggScellInfo	182
NASPRLPreferenceTlv	183
NASQmiCbkNasSwiOTAMessageInd	184

NASQmiCbkNasSystemSelPrefInd	184
NASRoamPreferenceTlv	185
NASServDomainPrefTlv	185
NASServingSystemInfo	186
NASTimeInfoTlv	187
newMTMessageTlv	188
pack_dms_ActivateAutomatic_t	188
pack_dms_GetCustFeaturesV2_t	189
pack_dms_ResetToFactoryDefaults_t	189
pack_dms_SetActivationStatusCallback_t	190
pack_dms_SetCrashAction_t	190
pack_dms_SetCustFeature_t	191
pack_dms_SetCustFeaturesV2_t	191
pack_dms_SetEventReport_t	192
pack_dms_SetPower_t	192
pack_dms_SetUSBComp_t	193
pack_dms_SLQSDmsSwiIndicationRegister_t	193
pack_dms_SLQSSwiGetCrashInfo_t	193
pack_dms_SLQSSwiSetDyingGaspCfg_t	194
pack_dms_SLQSSwiSetHostDevInfo_t	194
pack_dms_SLQSSwiSetOSInfo_t	195
pack_dms_UIMChangePIN_t	196
pack_dms_UIMGetControlKeyStatus_t	196
pack_dms_UIMGetICCID_t	197
pack_dms_UIMSetControlKeyProtection_t	197
pack_dms_UIMSetPINProtection_t	198
pack_dms_UIMUnblockControlKey_t	199
pack_dms_UIMUnblockPIN_t	200
pack_dms_UIMVerifyPIN_t	200
pack_fms_GetImagesPreference_t	201
pack_fms_GetStoredImages_t	201
pack_fms_SetImagesPreference_t	202
pack_loc_Delete_Assist_Data_t	203
pack_loc_EventRegister_t	204
pack_loc_SetExtPowerState_t	206
pack_loc_SetOperationMode_t	207
pack_loc_SLQSLOCGetBestAvailPos_t	207
pack_loc_SLQSLOCInjectPosition_t	208
pack_loc_SLQSLOCInjectSensorData_t	213
pack_loc_SLQSLOCInjectUTCTime_t	215
pack_loc_SLQSLOCSetCradleMountConfig_t	216
pack_loc_Start_t	216
pack_loc_Stop_t	218
pack_nas_SetACCOLC_t	218
pack_nas_SetNetworkPreference_t	219
pack_nas_SLQSGetPLMNName_t	220
pack_nas_SLQSInitiateNetworkRegistration_t	221
pack_nas_SLQSNasConfigSigInfo2_t	222
pack_nas_SLQSNasIndicationRegisterExt_t	226
pack_nas_SLQSNasSwiIndicationRegister_t	229
pack_nas_SLQSSetSignalStrengthsCallback_t	231
pack_nas_SLQSSetSysSelectionPref_t	231
pack_qmi_t	236
pack_qos_SLQSQosSwiReadApnExtraParams_t	237
pack_qos_SLQSQosSwiReadDataStats_t	237
pack_qos_SLQSSetQosEventCallback_t	238
pack_sms_SendSMS_t	238
pack_sms_SetNewSMSCallback_t	239

pack_sms_SLQSDelateSMS_t	239
pack_sms_SLQSGetSMS_t	240
pack_sms_SLQSGetSMSList_t	241
pack_sms_SLQSModifySMSStatus_t	241
pack_swiloc_SwiLocSetAutoStart_t	242
pack_swima_SLQSOMADMCancelSession_t	244
pack_swima_SLQSOMADMGetSessionInfo_t	245
pack_swima_SLQSOMADMSelectSelection_t	245
pack_swima_SLQSOMADMSetSettings_t	246
pack_swima_SLQSOMADMStartSession_t	247
pack_uim_ChangePin_t	247
pack_uim_ReadTransparent_t	248
pack_uim_SetPinProtection_t	249
pack_uim_SLQSUIEventRegister_t	250
pack_uim_SLQSUIPowerDown_t	251
pack_uim_SLQSUIPowerUp_t	251
pack_uim_SLQSUISwitchSlot_t	252
pack_uim_UnblockPin_t	253
pack_uim_VerifyPin_t	254
pack_wds_DHCPv4ClientLeaseChange_t	255
pack_wds_GetDefaultProfile_t	255
pack_wds_GetDefaultProfileNum_t	256
pack_wds_GetDormancyState_t	256
pack_wds_GetLastMobileIPError_t	256
pack_wds_GetMobileIP_t	256
pack_wds_GetMobileIPProfile_t	256
pack_wds_GetPacketStatistics_t	257
pack_wds_GetPacketStatus_t	257
pack_wds_GetSessionDuration_t	258
pack_wds_RMSetTransferStatistics_t	258
pack_wds_SetAutoconnect_t	258
pack_wds_SetDefaultProfile_t	259
pack_wds_SetDefaultProfileNum_t	260
pack_wds_SetMobileIP_t	260
pack_wds_SetMobileIPParameters_t	260
pack_wds_SetMobileIPProfile_t	262
pack_wds_SLQSCreateProfile_t	263
pack_wds_SLQSDelateProfile_t	264
pack_wds_SLQSGetCurrDataSystemStat_t	264
pack_wds_SLQSGetDataBearerTechnology_t	264
pack_wds_SLQSGetDUNCallInfo_t	264
pack_wds_SLQSGetProfileSettings_t	265
pack_wds_SLQSGetRuntimeSettings_t	266
pack_wds_SLQSModifyProfile_t	267
pack_wds_SLQSSet3GPPConfigItem_t	268
pack_wds_SLQSSetIPFamilyPreference_t	270
pack_wds_SLQSSetWdsEventCallback_t	270
pack_wds_SLQSSetDHCPv4ClientConfig_t	271
pack_wds_SLQSSetDHCPv4ClientConfig_t	271
pack_wds_SLQSSetLoopback_t	272
pack_wds_SLQSStartDataSession_t	273
pack_wds_SLQSStopDataSession_t	274
pack_wds_SLQSWdsSetEventReport_t	274
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	276
PackCreateProfileOut	277
packgetDyingGaspCfg	277
packgetDyingGaspStatistics	277
qmiSmsMessageList	278

qmiWSDDataBearerTechnology	278
RFBandInfoElements	279
rmTrasferStaticsReq	279
sensorData_t	280
slot_t	281
slotInf	282
slots_t	284
sMSCAddress	284
sMSCAddressTlv	285
sMSEtwsMessage	285
sMSEtwsMessageTlv	286
sMSEtwsPImn	286
sMSMessageMode	287
sMSMTMessage	287
sMSOnIMS	288
sMSOnIMSTlv	288
sMSTransferRouteMTMessage	289
tdscdmaSigInfoExt	290
tempData_t	290
transferRouteMessageTlv	291
transferStatInd	292
uim_appStatus	292
uim_cardResult	295
uim_cardStatus	296
uim_changeUIMPIN	297
uim_encryptedPIN1	298
uim_fileInfo	299
uim_hotSwapStatus	300
uim_readResult	300
uim_readTransparentInfo	301
uim_remainingRetries	301
uim_sessionInformation	302
uim_setPINProtection	303
uim_slotInfo	304
uim_UIMSessionInformation	305
uim_unblockUIMPIN	306
uim_verifyUIMPIN	307
unpack_dms_GetActivationState_t	308
unpack_dms_GetBandCapability_t	309
unpack_dms_GetCrashAction_t	309
unpack_dms_GetCustFeature_t	309
unpack_dms_GetCustFeaturesV2_t	310
unpack_dms_GetDeviceCap_t	311
unpack_dms_GetDeviceCapabilities_t	311
unpack_dms_GetDeviceHardwareRev_t	312
unpack_dms_GetDeviceMfr_t	312
unpack_dms_GetDeviceSerialNumbers_t	313
unpack_dms_GetFirmwareInfo_t	313
unpack_dms_GetFirmwareRevision_t	314
unpack_dms_GetFirmwareRevisions_t	315
unpack_dms_GetFSN_t	316
unpack_dms_GetHardwareRevision_t	316
unpack_dms_GetIMSI_t	316
unpack_dms_GetManufacturer_t	316
unpack_dms_GetModelID_t	317
unpack_dms_GetNetworkTime_t	317
unpack_dms_GetOfflineReason_t	318
unpack_dms_GetPower_t	319

unpack_dms_GetPRLVersion_t	320
unpack_dms_GetSerialNumbers_t	320
unpack_dms_GetUSBComp_t	321
unpack_dms_GetVoiceNumber_t	321
unpack_dms_ResetToFactoryDefaults_t	321
unpack_dms_SetActivationStatusCallback_t	322
unpack_dms_SetCrashAction_t	322
unpack_dms_SetCustFeature_t	323
unpack_dms_SetCustFeaturesV2_t	323
unpack_dms_SetEventReport_ind_t	323
unpack_dms_SetEventReport_t	324
unpack_dms_SetFirmwarePreference_t	324
unpack_dms_SetPower_t	324
unpack_dms_SetUSBComp_t	325
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t	325
unpack_dms_SLQSDmsSwiGetResetInfo_t	325
unpack_dms_SLQSDmsSwiIndicationRegister_t	326
unpack_dms_SLQSGetBandCapability_t	326
unpack_dms_SLQSGetERIFile_t	329
unpack_dms_SLQSSwiClearDyingGaspStatistics_t	330
unpack_dms_SLQSSwiGetCrashInfo_t	330
unpack_dms_SLQSSwiGetDyingGaspCfg_t	331
unpack_dms_SLQSSwiGetDyingGaspStatistics_t	331
unpack_dms_SLQSSwiGetFirmwareCurr_t	332
unpack_dms_SLQSSwiGetFwUpdateStatus_t	333
unpack_dms_SLQSSwiGetHostDevInfo_t	334
unpack_dms_SLQSSwiGetOSInfo_t	335
unpack_dms_SLQSSwiGetSerialNoExt_t	335
unpack_dms_SLQSSwiSetDyingGaspCfg_t	336
unpack_dms_SLQSSwiSetHostDevInfo_t	336
unpack_dms_SLQSSwiSetOSInfo_t	337
unpack_dms_SLQSUIMGetState_t	337
unpack_dms_UIMGetControlKeyStatus_t	338
unpack_dms_UIMGetICCID_t	339
unpack_dms_UIMGetPINStatus_t	339
unpack_dms_UIMSetControlKeyProtection_t	341
unpack_dms_UIMSetPINProtection_t	342
unpack_dms_UIMUnblockControlKey_t	342
unpack_fms_GetImagesPreference_t	343
unpack_fms_GetStoredImages_t	344
unpack_fms_SetImagesPreference_t	344
unpack_loc_BestAvailPos_Ind_t	345
unpack_loc_Delete_Assist_Data_t	351
unpack_loc_DeleteAssistData_Ind_t	351
unpack_loc_EngineState_Ind_t	352
unpack_loc_EventRegister_t	353
unpack_loc_GnssSvInfo_Ind_t	353
unpack_loc_PositionRpt_Ind_t	354
unpack_loc_SetExtPowerConfig_Ind_t	360
unpack_loc_SetExtPowerState_t	361
unpack_loc_SetOperationMode_Ind_t	362
unpack_loc_SetOperationMode_t	362
unpack_loc_SLQSLOCGetBestAvailPos_t	363
unpack_loc_Start_t	363
unpack_loc_Stop_t	364
unpack_nas_GetCDMANetworkParameters_t	364
unpack_nas_GetHomeNetwork_t	365
unpack_nas_GetNetworkPreference_t	366

<a href="#">unpack_nas_GetRFInfo_t</a>	367
<a href="#">unpack_nas_GetServingNetwork_t</a>	368
<a href="#">unpack_nas_GetServingNetworkCapabilities_t</a>	369
<a href="#">unpack_nas_GetSignalStrengths_t</a>	369
<a href="#">unpack_nas_PerformNetworkScan_t</a>	370
<a href="#">unpack_nas_SetDataCapabilitiesCallback_ind_t</a>	371
<a href="#">unpack_nas_SetEventReportInd_t</a>	371
<a href="#">unpack_nas_SetNasLTECphyCaIndCallback_ind_t</a>	372
<a href="#">unpack_nas_SetNetworkPreference_t</a>	373
<a href="#">unpack_nas_SetRoamingIndicatorCallback_ind_t</a>	373
<a href="#">unpack_nas_SetServingSystemCallback_ind_t</a>	374
<a href="#">unpack_nas_SlqsGetLTECphyCAInfo_t</a>	374
<a href="#">unpack_nas_SLQSGetNetworkTime_t</a>	375
<a href="#">unpack_nas_SLQSGetPLMNName_t</a>	376
<a href="#">unpack_nas_SLQSGetServingSystem_t</a>	376
<a href="#">unpack_nas_SLQSGetSignalStrength_t</a>	379
<a href="#">unpack_nas_SLQSGetSysInfo_t</a>	380
<a href="#">unpack_nas_SLQSGetSysSelectionPref_t</a>	383
<a href="#">unpack_nas_SLQSNasGetCellLocationInfo_t</a>	386
<a href="#">unpack_nas_SLQSNasGetSigInfo_t</a>	388
<a href="#">unpack_nas_SLQSNasNetworkTimeCallBack_ind_t</a>	389
<a href="#">unpack_nas_SLQSNasSigInfoCallback_ind_t</a>	390
<a href="#">unpack_nas_SLQSNasSwiModemStatus_t</a>	390
<a href="#">unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t</a>	391
<a href="#">unpack_nas_SLQSNasTimerCallback_ind_t</a>	391
<a href="#">unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t</a>	392
<a href="#">unpack_nas_SLQSSwiGetLteCQI_t</a>	393
<a href="#">unpack_nas_SLQSSwiGetLteSccRxInfo_t</a>	394
<a href="#">unpack_nas_SLQSSysInfoCallback_ind_t</a>	394
<a href="#">unpack_omaDmConfigTlv_t</a>	397
<a href="#">unpack_omaDmFotaTlv_t</a>	398
<a href="#">unpack_omaDmNotificationsTlv_t</a>	400
<a href="#">unpack_qmi_t</a>	401
<a href="#">unpack_qos_dataRate_t</a>	401
<a href="#">unpack_qos_IPv4Addr_t</a>	402
<a href="#">unpack_qos_IPv6Addr_t</a>	402
<a href="#">unpack_qos_IPv6TrafCls_t</a>	403
<a href="#">unpack_qos_pktErrRate_t</a>	404
<a href="#">unpack_qos_Port_t</a>	404
<a href="#">unpack_qos_QosFlowInfo_t</a>	405
<a href="#">unpack_qos_QosFlowInfoState_t</a>	406
<a href="#">unpack_qos_SLQSQosGetNetworkStatus_t</a>	407
<a href="#">unpack_qos_SLQSQosSwiReadApnExtraParams_t</a>	408
<a href="#">unpack_qos_SLQSQosSwiReadDataStats_t</a>	409
<a href="#">unpack_qos_SLQSSetQosEventCallback_ind_t</a>	410
<a href="#">unpack_qos_SLQSSetQosNWStatusCallback_ind_t</a>	411
<a href="#">unpack_qos_SLQSSetQosPriEventCallback_ind_t</a>	411
<a href="#">unpack_qos_SLQSSetQosStatusCallback_ind_t</a>	412
<a href="#">unpack_qos_swiQosFilter_t</a>	413
<a href="#">unpack_qos_swiQosFlow_t</a>	417
<a href="#">unpack_qos_tokenBucket_t</a>	421
<a href="#">unpack_qos_Tos_t</a>	422
<a href="#">unpack_QosFlowStat_t</a>	422
<a href="#">unpack_RMTransferStatistics_ind_t</a>	423
<a href="#">unpack_sms_SendSMS_t</a>	424
<a href="#">unpack_sms_SetNewSMSCallback_ind_t</a>	425
<a href="#">unpack_sms_SetNewSMSCallback_t</a>	426
<a href="#">unpack_sms_SLQSDeleteSMS_t</a>	426

unpack_sms_SLQSGetSMS_t	426
unpack_sms_SLQSGetSMSList_t	427
unpack_sms_SLQSModifySMSStatus_t	427
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	428
unpack_swilloc_SwiLocGetAutoStart_t	428
unpack_swioma_SLQSOMADMAAlertCallback_ind_t	430
unpack_swioma_SLQSOMADMGetSessionInfo_t	431
unpack_swioma_SLQSOMADMGetSettings_t	434
unpack_swioma_SLQSOMADMStartSession_t	435
unpack_uim_ChangePin_t	436
unpack_uim_GetCardStatus_t	437
unpack_uim_ReadTransparent_t	437
unpack_uim_SetPinProtection_t	438
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	439
unpack_uim_SLQSUIEventRegister_t	440
unpack_uim_SLQSUIGetSlotsStatus_t	440
unpack_uim_SLQSUISetStatusChangeCallBack_ind_t	441
unpack_uim_UnblockPin_t	441
unpack_uim_VerifyPin_t	442
unpack_wds_DHCPv4ClientLease_ind_t	443
unpack_wds_GetAutoconnect_t	443
unpack_wds_GetByteTotals_t	444
unpack_wds_GetConnectionRate_t	444
unpack_wds_GetDataBearerTechnology_t	445
unpack_wds_GetDefaultProfile_t	446
unpack_wds_GetDefaultProfileNum_t	447
unpack_wds_GetDormancyState_t	447
unpack_wds_GetLastMobileIPError_t	448
unpack_wds_GetMobileIP_t	448
unpack_wds_GetMobileIPProfile_t	448
unpack_wds_GetPacketStatistics_t	450
unpack_wds_GetPacketStatus_t	451
unpack_wds_GetSessionDuration_t	453
unpack_wds_GetSessionState_t	453
unpack_wds_RMSetTransferStatistics_t	454
unpack_wds_SetMobileIPProfile_t	454
unpack_wds_SLQSCreateProfile_t	454
unpack_wds_SLQSDeleteProfile_t	454
unpack_wds_SLQSGet3GPPConfigItem_t	455
unpack_wds_SLQSGetCurrDataSystemStat_t	456
unpack_wds_SLQSGetCurrentChannelRate_t	457
unpack_wds_SLQSGetDataBearerTechnology_t	458
unpack_wds_SLQSGetDUNCallInfo_t	458
unpack_wds_SLQSGetProfileSettings_t	459
unpack_wds_SLQSGetRuntimeSettings_t	460
unpack_wds_SLQSModifyProfile_t	462
unpack_wds_SLQSSetIPFamilyPreference_t	462
unpack_wds_SLQSSetPacketSrvStatusCallback_t	463
unpack_wds_SLQSSetWdsEventCallback_ind_t	464
unpack_wds_SLQSSetDHCPv4ClientConfig_t	465
unpack_wds_SLQSSetLoopback_t	466
unpack_wds_SLQSSetDataSession_t	466
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	467
UnPackGetProfileSettingOut	469
unpackWdsProfileParam	469
wds_currNetworkInfo	469
wds_DataULongLongTlv	471
wds_DataULongTlv	471

wds_DHCPLeaseOptTlv	472
wds_DHCPLeaseStateTlv	472
wds_DHCPOpt	472
wds_DHCPProfileIdTlv	473
wds_DHCPv4HWConfig	473
wds_DHCPv4Option	474
wds_DHCPv4OptionList	474
wds_DHCPv4ProfileId	475
wds_Domain	476
wds_DomainNameList	476
wds_GPRSQoS	477
wds_IPv4AdTlv	478
wds_IPV6AddressInfo	478
wds_IPV6GWAddressInfo	478
wds_PCSCFFQDNAddress	479
wds_PCSCFFQDNAddressList	480
wds_PCSCFIPv4ServerAddressList	480
wds_ProfileIdentifier	481
wds_profileInfo	481
wds_TrStatInd	482
wds_UMTSMInQoS	482
wdsDhcpv4HwConfig	485
wdsDhcpv4Option	486
wdsDhcpv4OptionList	486
wdsDhcpv4ProfileId	487



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">apdoxypages.c</a>	Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages	489
<a href="#">common.h</a>		489
<a href="#">dms.h</a>		494
<a href="#">fms.h</a>		552
<a href="#">libsdp.h</a>		556
<a href="#">loc.h</a>		563
<a href="#">nas.h</a>		580
<a href="#">qaGobiApiTableBandClasses.h</a>	Network Access Service API Band Classes table	612
<a href="#">qaGobiApiTableCallControlReturnReasons.h</a>	Call Control Return Reasons table	615
<a href="#">qaGobiApiTableCallEndReasons.h</a>	Wireless Data Service Call End Reasons	616
<a href="#">qaGobiApiTableCarrierCodes.h</a>	Carrier Codes table	631
<a href="#">qaGobiApiTableCodingScheme.h</a>	Data Coding Scheme	633
<a href="#">qaGobiApiTableGpsCapabilityCodes.h</a>	Position Determination Service API GPS Capability Codes	636
<a href="#">qaGobiApiTablePowerModes.h</a>	Device Management Service API Power Modes table	637
<a href="#">qaGobiApiTableRadioInterfaces.h</a>	Network Access Service API Radio Interfaces table	637
<a href="#">qaGobiApiTableRegionCodes.h</a>	Region Codes table	638
<a href="#">qaGobiApiTableServiceOptions.h</a>	Voice Service Options	638
<a href="#">qaGobiApiTableSupServiceInfoClasses.h</a>	Voice Supplementary Service Information Classes	641
<a href="#">qaGobiApiTableSwiAudio.h</a>	Swi Audio related tables	641
<a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a>	Update Complete Status table	642

<a href="#">qaGobiApiTableVoiceCallEndReasons.h</a>	
Voice Service Call and supplementary services end reasons . . . . .	643
<a href="#">qmerrno.h</a> . . . . .	650
<a href="#">qos.h</a> . . . . .	657
<a href="#">sms.h</a> . . . . .	666
<a href="#">SwiDataTypes.h</a>	
SWI data types . . . . .	676
<a href="#">swiloc.h</a> . . . . .	677
<a href="#">swioma.h</a> . . . . .	679
<a href="#">SWIWWANCMAPI.h</a> . . . . .	689
<a href="#">uim.h</a> . . . . .	689
<a href="#">wds.h</a> . . . . .	700

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

## 8.1.2 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 altSrcInfo\_t Struct Reference

### Data Fields

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

### 8.2.1 Detailed Description

This structure specifies information regarding the altitude source

#### Parameters

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



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

## 8.2.2 Field Documentation

8.2.2.1 `uint32_t altSrcInfo_t::coverage`

8.2.2.2 `uint32_t altSrcInfo_t::linkage`

8.2.2.3 `uint32_t altSrcInfo_t::source`

## 8.3 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`

### 8.3.1 Detailed Description

This structure contains Application Status Information loaded on the card.

## Parameters

<i>appType</i>	<ul style="list-style-type: none"> <li>Indicates the type of the application. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - SIM card</li> <li>2 - USIM application</li> <li>3 - RUIM card</li> <li>4 - CSIM application</li> <li>5 - ISIM application</li> </ul> </li> <li>Other values are reserved for the future and are to be handled as "Unknown".</li> </ul>
<i>appState</i>	<ul style="list-style-type: none"> <li>Indicates the state of the application. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Detected</li> <li>2 - PIN1 or UPIN is required</li> <li>3 - PUK1 or PUK for UPIN is required</li> <li>4 - Personalization state must be checked</li> <li>5 - PIN1 is blocked</li> <li>6 - Illegal</li> <li>7 - Ready</li> </ul> </li> </ul>
<i>persoState</i>	<ul style="list-style-type: none"> <li>Indicates the state of the personalization for the application. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Personalization operation is in progress</li> <li>2 - Ready</li> <li>3 - Personalization code is required</li> <li>4 - PUK for personalization code is required</li> <li>5 - Permanently blocked</li> </ul> </li> </ul>
<i>persoFeature</i>	<ul style="list-style-type: none"> <li>Indicates the personalization feature.</li> <li>This applies only when a personalization code is required to deactivate or unblock personalization. <ul style="list-style-type: none"> <li>0 - GW network personalization</li> <li>1 - GW network subset personalization</li> <li>2 - GW service provider personalization</li> <li>3 - GW corporate personalization</li> <li>4 - GW UIM personalization</li> <li>5 - 1X network type 1 personalization</li> <li>6 - 1X network type 2 personalization</li> <li>7 - 1X HRPD personalization</li> <li>8 - 1X service provider personalization</li> <li>9 - 1X corporate personalization</li> <li>10 - 1X RUIM personalization</li> <li>11 - Unknown</li> </ul> </li> </ul>
<i>persoRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to disable the personalization.</li> </ul>

<i>persoUnblockRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock the personalization.</li> </ul>
<i>aidLength</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements. i.e. aidVal</li> <li>If zero(0) then no aidVal information exists.</li> </ul>
<i>aidVal[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>Application identifier value.</li> </ul>
<i>univPin</i>	<ul style="list-style-type: none"> <li>Indicates whether UPIN replaces PIN1. <ul style="list-style-type: none"> <li>0 - PIN1 is used</li> <li>1 - UPIN replaces PIN1</li> </ul> </li> </ul>
<i>pin1State</i>	<ul style="list-style-type: none"> <li>Indicates the state of PIN1. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Enabled and not verified</li> <li>2 - Enabled and verified</li> <li>3 - Disabled</li> <li>4 - Blocked</li> <li>5 - Permanently blocked</li> </ul> </li> </ul>
<i>pin1Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to verify PIN1.</li> </ul>
<i>puk1Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock PIN1.</li> </ul>
<i>pin2State</i>	<ul style="list-style-type: none"> <li>Indicates the state of PIN2. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Enabled and not verified</li> <li>2 - Enabled and verified</li> <li>3 - Disabled</li> <li>4 - Blocked</li> <li>5 - Permanently blocked</li> </ul> </li> </ul>
<i>pin2Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to verify PIN2.</li> </ul>
<i>puk2Retries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock PIN2.</li> </ul>

### 8.3.2 Field Documentation

#### 8.3.2.1 uint8\_t appStats::aidLength

#### 8.3.2.2 uint8\_t appStats::aidVal[255]

8.3.2.3 uint8\_t appStats::appState

8.3.2.4 uint8\_t appStats::appType

8.3.2.5 uint8\_t appStats::persoFeature

8.3.2.6 uint8\_t appStats::persoRetries

8.3.2.7 uint8\_t appStats::persoState

8.3.2.8 uint8\_t appStats::persoUnblockRetries

8.3.2.9 uint8\_t appStats::pin1Retries

8.3.2.10 uint8\_t appStats::pin1State

8.3.2.11 uint8\_t appStats::pin2Retries

8.3.2.12 uint8\_t appStats::pin2State

8.3.2.13 uint8\_t appStats::puk1Retries

8.3.2.14 uint8\_t appStats::puk2Retries

8.3.2.15 uint8\_t appStats::univPin

## 8.4 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]

### 8.4.1 Detailed Description

This structure contains the Carrier Image parameters.

#### Parameters

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

### 8.4.2 Field Documentation

8.4.2.1 `uint8_t CarrierImage_t::m_FwBuildId[32]`

8.4.2.2 `uint8_t CarrierImage_t::m_FwImageId[16]`

8.4.2.3 `uint32_t CarrierImage_t::m_nCarrierId`

8.4.2.4 `uint32_t CarrierImage_t::m_nFolderId`

8.4.2.5 `uint32_t CarrierImage_t::m_nStorage`

8.4.2.6 `uint8_t CarrierImage_t::m_PriBuildId[32]`

8.4.2.7 `uint8_t CarrierImage_t::m_PriImageId[16]`

## 8.5 cdmaSSInfo Struct Reference

#### Data Fields

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

### 8.5.1 Detailed Description

#### Parameters

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

### 8.5.2 Field Documentation

8.5.2.1 `int16_t cdmaSSInfo::ecio`

8.5.2.2 `int8_t cdmaSSInfo::rssI`

## 8.6 connectionStatus Struct Reference

#### Data Fields

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

### 8.6.1 Detailed Description

#### Parameters

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

### 8.6.2 Field Documentation

8.6.2.1 `uint64_t connectionStatus::MDMCallDuration`

8.6.2.2 `uint8_t connectionStatus::MDMConnStatus`

## 8.7 crashInfoParams Struct Reference

#### Data Fields

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

### 8.7.1 Detailed Description

This structure contains [crashInfoParams](#)

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>crashInfo</i> ↔ <i>Param[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">crashInfoParams</a></li> </ul>

## 8.7.2 Field Documentation

### 8.7.2.1 crashInformation crashInfoParams::crashInfo

### 8.7.2.2 uint8\_t crashInfoParams::crashStatus

## 8.8 crashInformation Struct Reference

### Data Fields

- uint16\_t [numCrashes](#)
- uint32\_t [crashId](#)
- uint32\_t [crashData](#)
- uint16\_t [crashStrlen](#)
- char [crashString](#) [255]
- uint16\_t [gcdumpStrlen](#)
- char [gcdumpString](#) [1024]

### 8.8.1 Detailed Description

This structure contains [crashInformation](#)

#### Parameters

<i>numCrashes</i> ↔ <i>OUT</i>	<ul style="list-style-type: none"> <li>number of instances of the remaining fields</li> </ul>
<i>crashId[OUT]</i>	<ul style="list-style-type: none"> <li>random crash id assigned at crash</li> </ul>
<i>crashData</i>	<ul style="list-style-type: none"> <li>crash data[OUT]</li> </ul>
<i>crashStrlen</i> ↔ <i>N/OUT</i>	<ul style="list-style-type: none"> <li>length of the crashString field as an input, length of the crashString field returned by API as an OUTPUT</li> </ul>
<i>crashString</i>	<ul style="list-style-type: none"> <li>crash string</li> </ul>
<i>gcdumpStrlen</i> ↔ <i>OUT</i>	<ul style="list-style-type: none"> <li>length of the gcdumpString field as an input, length of the gcdumpString field returned by API as an OUTPUT</li> </ul>
Generated by Doxygen	
<i>gcdumpString</i> ↔ <i>OUT</i>	<ul style="list-style-type: none"> <li>gcdump string for the crash</li> </ul>

## 8.8.2 Field Documentation

8.8.2.1 `uint32_t` `crashInformation::crashData`

8.8.2.2 `uint32_t` `crashInformation::crashId`

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

8.8.2.4 `uint16_t` `crashInformation::crashStrlen`

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

8.8.2.6 `uint16_t` `crashInformation::gcdumpStrlen`

8.8.2.7 `uint16_t` `crashInformation::numCrashes`

## 8.9 currNetworkInfo Struct Reference

### Data Fields

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

### 8.9.1 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.

## 8.9.2 Field Documentation

8.9.2.1 `uint8_t` `currNetworkInfo::NetworkType`

8.9.2.2 `uint32_t` `currNetworkInfo::RATMask`

8.9.2.3 `uint32_t` `currNetworkInfo::SOMask`

## 8.10 dms\_ActivationStatusTlv Struct Reference

### Data Fields

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



### 8.10.1 Detailed Description

Activation Status Tlv

Parameters

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

### 8.10.2 Field Documentation

8.10.2.1 uint32\_t dms\_ActivationStatusTlv::activationStatus

8.10.2.2 uint16\_t dms\_ActivationStatusTlv::TlvPresent

## 8.11 dms\_OperatingModeTlv Struct Reference

Data Fields

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

### 8.11.1 Detailed Description

Operating Mode Tlv

Parameters

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

### 8.11.2 Field Documentation

8.11.2.1 uint32\_t dms\_OperatingModeTlv::operatingMode

8.11.2.2 uint16\_t dms\_OperatingModeTlv::TlvPresent

## 8.12 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](#)

### 8.12.1 Detailed Description

This structure contains information about Customization Setting. This TLV is only applicable for 9x30 modules so far

#### Parameters

<i>id_length</i>	<ul style="list-style-type: none"> <li>• length of cust_id field</li> </ul>
<i>cust_id</i>	<ul style="list-style-type: none"> <li>• Customization ID (Maximum 64 bytes)</li> </ul>
<i>value_length</i>	<ul style="list-style-type: none"> <li>• length of cust_value field</li> </ul>
<i>cust_value</i>	<ul style="list-style-type: none"> <li>• Customization Setting Value (Maximum 8 bytes)</li> </ul>
<i>cust_attr</i>	<ul style="list-style-type: none"> <li>• Customization Setting attribute through QMI             <ul style="list-style-type: none"> <li>– bit 0: Values:                 <ul style="list-style-type: none"> <li>* 0 - read only</li> <li>* 1 - read/write</li> </ul> </li> </ul> </li> </ul>

### 8.12.2 Field Documentation

8.12.2.1 uint16\_t DMScustSettingInfo::cust\_attr

8.12.2.2 uint8\_t DMScustSettingInfo::cust\_id[64+1]

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

8.12.2.4 `uint16_t DMScustSettingInfo::id_length`

8.12.2.5 `uint16_t DMScustSettingInfo::value_length`

## 8.13 DMScustSettingList Struct Reference

### Data Fields

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

### 8.13.1 Detailed Description

This structure contains the fields of TLV Customization Setting List. This TLV is only applicable for 9x30 modules so far

#### Parameters

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

### 8.13.2 Field Documentation

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

8.13.2.2 `uint8_t DMScustSettingList::list_type`

8.13.2.3 `uint16_t DMScustSettingList::num_instances`

## 8.14 DMSgetCustomFeatureV2 Struct Reference

### Data Fields

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

### 8.14.1 Detailed Description

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

#### Parameters

--	--

### 8.14.2 Field Documentation

8.14.2.1 **DMScustSettingInfo\*** DMSgetCustomFeatureV2::pCustSettingInfo

8.14.2.2 **DMScustSettingList\*** DMSgetCustomFeatureV2::pCustSettingList

8.14.2.3 **DMSgetCustomInput\*** DMSgetCustomFeatureV2::pGetCustomInput

## 8.15 DMSgetCustomInput Struct Reference

### Data Fields

- uint8\_t [cust\\_id](#) [64+1]
- uint8\_t [list\\_type](#)

### 8.15.1 Detailed Description

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

#### Parameters

<i>cust_id</i>	<ul style="list-style-type: none"> <li>• Customization ID (Maximum 64 bytes)</li> </ul>
<i>list_type</i>	<ul style="list-style-type: none"> <li>• list type requested</li> </ul>

### 8.15.2 Field Documentation

8.15.2.1 uint8\_t DMSgetCustomInput::cust\_id[64+1]

8.15.2.2 uint8\_t DMSgetCustomInput::list\_type

## 8.16 dunchannelRate Struct Reference

## Data Fields

- uint32\_t [CurrChanTxRate](#)
- uint32\_t [CurrChanRxRate](#)
- uint32\_t [MaxChanTxRate](#)
- uint32\_t [MaxChanRxRate](#)

### 8.16.1 Detailed Description

#### Parameters

<i>CurrChanTxRate</i>	instantaneous channel Tx rate in bits per second
<i>CurrChanRx↔Rate</i>	instantaneous channel Rx rate in bits per second
<i>MaxChanTxRate</i>	maximum Tx rate that can be assigned to the device
<i>MaxChanRx↔Rate</i>	maximum Rx rate that can be assigned to the device

### 8.16.2 Field Documentation

8.16.2.1 uint32\_t dunchannelRate::CurrChanRxRate

8.16.2.2 uint32\_t dunchannelRate::CurrChanTxRate

8.16.2.3 uint32\_t dunchannelRate::MaxChanRxRate

8.16.2.4 uint32\_t dunchannelRate::MaxChanTxRate

## 8.17 eriDataparams Struct Reference

## Data Fields

- uint16\_t [eriDataLen](#)
- uint8\_t [eriData](#) [1024]

### 8.17.1 Field Documentation

8.17.1.1 uint8\_t eriDataparams::eriData[1024]

8.17.1.2 uint16\_t eriDataparams::eriDataLen

## 8.18 eTWSPLMNInfoTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- [sMSEtwsPlmnInfo](#) [ETWSPLMNInfo](#)

### 8.18.1 Detailed Description

#### Parameters

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

## 8.18.2 Field Documentation

### 8.18.2.1 sMSEtwsPlmnInfo eTWSPLMNInfoTlv::ETWSPLMNInfo

#### 8.18.2.2 uint8\_t eTWSPLMNInfoTlv::TlvPresent

## 8.19 FMSImageElement Struct Reference

### Data Fields

- uint8\_t [imageType](#)
- uint8\_t [imageId](#) [16]
- uint8\_t [buildIdLength](#)
- uint8\_t [buildId](#) [100]

### 8.19.1 Detailed Description

This structure contains the Image Element information

#### Parameters

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

## 8.19.2 Field Documentation

### 8.19.2.1 uint8\_t FMSImageElement::buildId[100]

8.19.2.2 `uint8_t FMSImageElement::buildIdLength`

8.19.2.3 `uint8_t FMSImageElement::imageId[16]`

8.19.2.4 `uint8_t FMSImageElement::imageType`

## 8.20 FMSImageIdElement Struct Reference

### Data Fields

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

### 8.20.1 Detailed Description

This structure contains the Image ID list element Information

#### Parameters

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

### 8.20.2 Field Documentation

8.20.2.1 `uint8_t FMSImageIdElement::buildID[100]`

8.20.2.2 `uint8_t FMSImageIdElement::buildIDLength`

8.20.2.3 `uint8_t FMSImageIdElement::failureCount`

8.20.2.4 `uint8_t FMSImageIdElement::imageID[16]`

8.20.2.5 `uint8_t FMSImageIdElement::storageIndex`

## 8.21 FMSImageIDEntries Struct Reference

### Data Fields

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

### 8.21.1 Detailed Description

This structure contains the list entry Information

#### Parameters

<i>imageType</i>	<ul style="list-style-type: none"> <li>• Type of image <ul style="list-style-type: none"> <li>– 0 - Modem</li> <li>– 1 - PRI</li> </ul> </li> </ul>
<i>maxImages</i>	<ul style="list-style-type: none"> <li>• Maximum number of images of this type that may be stored concurrently on the device</li> </ul>
<i>executingImage</i>	<ul style="list-style-type: none"> <li>• Index (into the next array) of image that is currently executing</li> </ul>
<i>imageIDSize</i>	<ul style="list-style-type: none"> <li>• The number of elements in the image ID list</li> </ul>
<i>imageIDElement</i>	<ul style="list-style-type: none"> <li>• Array of ImageIDElement Structure ( Max 50 elements )</li> <li>• See <a href="#">FMSImageIdElement</a></li> </ul>

### 8.21.2 Field Documentation

8.21.2.1 `uint8_t FMSImageIDEntries::executingImage`

8.21.2.2 `FMSImageIdElement FMSImageIDEntries::imageIDElement[50]`

8.21.2.3 `uint8_t FMSImageIDEntries::imageIDSize`



8.21.2.4 `uint8_t FMSPrefImageList::imageType`

8.21.2.5 `uint8_t FMSPrefImageList::maxImages`

## 8.22 FMSImageList Struct Reference

### Data Fields

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

### 8.22.1 Detailed Description

This structure contains the Get Stored Images List

#### Parameters

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

### 8.22.2 Field Documentation

8.22.2.1 `FMSImageIDEntries FMSPrefImageList::imageIDEntries`[2]

8.22.2.2 `uint8_t FMSPrefImageList::listSize`

## 8.23 FMSPrefImageList Struct Reference

### Data Fields

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

### 8.23.1 Detailed Description

This structure contains the Preference Image List information

#### Parameters

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

## 8.23.2 Field Documentation

### 8.23.2.1 FMSPrefImageElement FMSPrefImageList::listEntries[2]

#### 8.23.2.2 uint8\_t FMSPrefImageList::listSize

## 8.24 hdrSSInfo Struct Reference

### Data Fields

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

### 8.24.1 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.

## 8.24.2 Field Documentation

### 8.24.2.1 int16\_t hdrSSInfo::ecio

### 8.24.2.2 int32\_t hdrSSInfo::io

### 8.24.2.3 int8\_t hdrSSInfo::rssi

### 8.24.2.4 uint8\_t hdrSSInfo::sinr

## 8.25 image\_info\_t Struct Reference

### Data Fields

- uint8\_t [imageType](#)
- uint8\_t [uniqueID](#) [16]
- uint8\_t [buildIDLen](#)
- uint8\_t [buildID](#) [255]

### 8.25.1 Field Documentation

8.25.1.1 `uint8_t image_info_t::buildID[255]`

8.25.1.2 `uint8_t image_info_t::buildIDLen`

8.25.1.3 `uint8_t image_info_t::imageType`

8.25.1.4 `uint8_t image_info_t::uniqueID[16]`

## 8.26 ipv6AddressInfo Struct Reference

### Data Fields

- `uint8_t IPV6PrefixLen`
- `uint16_t IPAddressV6 [8]`

### 8.26.1 Detailed Description

#### Parameters

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

### 8.26.2 Field Documentation

8.26.2.1 `uint16_t ipv6AddressInfo::IPAddressV6[8]`

8.26.2.2 `uint8_t ipv6AddressInfo::IPV6PrefixLen`

## 8.27 LibPackGPRSRequestedQoS Struct Reference

### Data Fields

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

### 8.27.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

#### Parameters

<i>precedence</i> ↔ Class	• Precedence class
<i>delayClass</i>	• Delay class
<i>reliabilityClass</i>	• Reliability class
<i>peak</i> ↔ Throughput↔ Class	• Peak throughput class
<i>mean</i> ↔ Throughput↔ Class	• Mean throughput class

## 8.27.2 Field Documentation

8.27.2.1 `uint32_t LibPackGPRSRequestedQoS::delayClass`

8.27.2.2 `uint32_t LibPackGPRSRequestedQoS::meanThroughputClass`

8.27.2.3 `uint32_t LibPackGPRSRequestedQoS::peakThroughputClass`

8.27.2.4 `uint32_t LibPackGPRSRequestedQoS::precedenceClass`

8.27.2.5 `uint32_t LibPackGPRSRequestedQoS::reliabilityClass`

## 8.28 LibpackProfile3GPP Struct Reference

### Data Fields

- `uint8_t * pProfilename`
- `uint16_t * pProfilenameSize`
- `uint8_t * pPDPTtype`
- `uint8_t * pPdpHdrCompType`
- `uint8_t * pPdpDataCompType`
- `uint8_t * pAPNName`
- `uint16_t * pAPNnameSize`
- `uint32_t * pPriDNSIPv4AddPref`
- `uint32_t * pSecDNSIPv4AddPref`
- `LibPackUMTSQoS * pUMTSReqQoS`
- `LibPackUMTSQoS * pUMTSMinQoS`
- `LibPackGPRSRequestedQoS * pGPRSRequestedQoS`
- `LibPackGPRSRequestedQoS * pGPRSMinimumQoS`
- `uint8_t * pUsername`
- `uint16_t * pUsernameSize`
- `uint8_t * pPassword`
- `uint16_t * pPasswordSize`

- uint8\_t \* pAuthenticationPref
- uint32\_t \* pIPv4AddrPref
- uint8\_t \* pPcsfAddrUsingPCO
- uint8\_t \* pPdpAccessConFlag
- uint8\_t \* pPcsfAddrUsingDhcp
- 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

### 8.28.1 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"> <li>Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.</li> </ul>
<i>pProfileName</i>	<ul style="list-style-type: none"> <li>One or more uint8_ts describing the profile</li> </ul>
<i>pProfilename</i> ↔ Size;	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pPDPTType</i>	<ul style="list-style-type: none"> <li>Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile <ul style="list-style-type: none"> <li>0x00 - PDP-IP (IPv4)</li> <li>0x01 - PDP-PPP</li> <li>0x02 - PDP-IPV6</li> <li>0x03 - PDP-IPV4V6</li> </ul> </li> </ul>
<i>pPdpHdr</i> ↔ CompType	<ul style="list-style-type: none"> <li>PDP header compression type <ul style="list-style-type: none"> <li>0 - PDP header compression is OFF</li> <li>1 - Manufacturer preferred compression</li> <li>2 - PDP header compression based on RFC 1144</li> <li>3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095</li> </ul> </li> </ul>

<i>pPdpData</i> ↔ <i>CompType</i>	<ul style="list-style-type: none"> <li>PDP data compression type <ul style="list-style-type: none"> <li>0 - PDP data compression is OFF</li> <li>1 - Manufacturer preferred compression</li> <li>2 - V.42BIS data compression</li> <li>3 - V.44 data compression</li> </ul> </li> </ul>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>Access point name</li> </ul>
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pPriDNSIPv4</i> ↔ <i>AddPref</i>	<ul style="list-style-type: none"> <li>Primary DNS IPv4 Address Preference</li> </ul>
<i>pSecDNSIPv4</i> ↔ <i>AddPref</i>	<ul style="list-style-type: none"> <li>Secondary DNS IPv4 Address Preference</li> </ul>
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> <li>UMTS Requested QoS</li> </ul>
<i>pUMTSMinQoS</i>	<ul style="list-style-type: none"> <li>UMTS Minimum QoS</li> </ul>
<i>pGPRS</i> ↔ <i>RequestedQoS</i>	<ul style="list-style-type: none"> <li>GPRS Minimum QoS</li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>User name</li> </ul>
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pPassword</i>	<ul style="list-style-type: none"> <li>Password</li> </ul>
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 uint8_ts.</li> </ul>

<i>pAuthPref</i>	<ul style="list-style-type: none"> <li>• Authentication Preference <ul style="list-style-type: none"> <li>– Bit map that indicates the authentication algorithm preference <ul style="list-style-type: none"> <li>* Bit 0 - PAP preference <ul style="list-style-type: none"> <li>• 0 - PAP is never performed</li> <li>• 1 - PAP may be performed</li> </ul> </li> <li>* Bit 1 - CHAP preference <ul style="list-style-type: none"> <li>• 0 - CHAP is never performed</li> <li>• 1 - CHAP may be performed</li> </ul> </li> <li>* If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul> </li> </ul>
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> <li>• IPv4 Address Preference</li> </ul>
<i>pPcscfAddrUsingPCO</i>	<ul style="list-style-type: none"> <li>• P-CSCF Address using PCO Flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request PCSCF address using PCO</li> <li>– 0 - (FALSE) implies do not request By default, this value is 0</li> </ul> </li> </ul>
<i>pPdpAccessConFlag</i>	<ul style="list-style-type: none"> <li>• PDP access control flag <ul style="list-style-type: none"> <li>– 0 - PDP access control none</li> <li>– 1 - PDP access control reject</li> <li>– 2 - PDP access control permission</li> </ul> </li> </ul>
<i>pPcscfAddrUsingDhcp</i>	<ul style="list-style-type: none"> <li>• P-CSCF address using DHCP <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies Request PCSCF address using DHCP</li> <li>– 0 - (FALSE) implies do not request By default, value is 0</li> </ul> </li> </ul>
<i>pImCnFlag</i>	<ul style="list-style-type: none"> <li>• IM CN flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request IM CN flag for this profile</li> <li>– 0 - (FALSE) implies do not request IM CN flag for this profile</li> </ul> </li> </ul>
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> </ul>
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> </ul>
<i>pPdpContext</i>	<ul style="list-style-type: none"> <li>• PDP context number</li> </ul>
<i>pSecondaryFlag</i>	<ul style="list-style-type: none"> <li>• PDP context secondary flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies this is secondary profile</li> <li>– 0 - (FALSE) implies this is not secondary profile</li> </ul> </li> </ul>

<i>pPrimaryID</i>	<ul style="list-style-type: none"> <li>PDP context primary ID</li> <li>function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device</li> </ul>
<i>pIPv6AddPref</i>	<ul style="list-style-type: none"> <li>IPv6 address preference Preferred IPv6 address to be assigned to the TE; actual assigned address is negotiated with the network and may differ from this value; if not specified, the IPv6 address is obtained automatically from the network</li> </ul>
<i>pUMTSReqQoS</i> ↔ <i>SSigInd</i>	<ul style="list-style-type: none"> <li>UMTS requested QoS with Signalling Indication flag</li> </ul>
<i>pUMTSMinQoS</i> ↔ <i>SSigInd</i>	<ul style="list-style-type: none"> <li>UMTS minimum QoS with Signalling Indication flag</li> </ul>
<i>pPrimaryDNS</i> ↔ <i>Pv6addpref</i>	<ul style="list-style-type: none"> <li>Primary DNS IPv6 address preference <ul style="list-style-type: none"> <li>The value may be used as a preference during negotiation with the network; if not specified, the wireless device will attempt to obtain the DNS address automatically from the network; the negotiated value is provided to the host via DHCP</li> </ul> </li> </ul>
<i>pSecondaryDNS</i> ↔ <i>IPv6addpref</i>	<ul style="list-style-type: none"> <li>Secondary DNS IPv6 address preference</li> </ul>
<i>paddr</i> ↔ <i>AllocationPref</i>	<ul style="list-style-type: none"> <li>DHCP/NAS preference <ul style="list-style-type: none"> <li>This enumerated value may be used to indicate the address allocation preference <ul style="list-style-type: none"> <li>* 0 - NAS signaling is used for address allocation</li> <li>* 1 - DHCP is used for address allocation</li> </ul> </li> </ul> </li> </ul>
<i>pQoSClassID</i>	<ul style="list-style-type: none"> <li>3GPP LTE QoS parameters</li> </ul>
<i>pAPNDisabled</i> ↔ <i>Flag</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t Flag indicating if the APN is disabled/enabled</li> <li>If set, the profile can not be used for making data calls</li> <li>Any data call is failed locally</li> <li>Values: <ul style="list-style-type: none"> <li>0 - FALSE(default)</li> <li>1 - True</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pPDNInactiv</i> ↔ <i>Timeout</i>	<ul style="list-style-type: none"> <li>Optional 4 uint8_ts indicating the duration of inactivity timer in seconds</li> <li>If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected</li> <li>Default value of zero indicates infinite value</li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>



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

## 8.28.2 Field Documentation

8.28.2.1 uint8\_t\* LibpackProfile3GPP::pAddrAllocPref

8.28.2.2 uint8\_t\* LibpackProfile3GPP::pAPNClass

8.28.2.3 uint8\_t\* LibpackProfile3GPP::pAPNDisabledFlag

8.28.2.4 uint8\_t\* LibpackProfile3GPP::pAPNName

8.28.2.5 uint16\_t\* LibpackProfile3GPP::pAPNnameSize

8.28.2.6 uint8\_t\* LibpackProfile3GPP::pAuthenticationPref

8.28.2.7 LibPackGPRSRequestedQoS\* LibpackProfile3GPP::pGPRSMinimumQoS

8.28.2.8 LibPackGPRSRequestedQoS\* LibpackProfile3GPP::pGPRSRequestedQoS

8.28.2.9 uint8\_t\* LibpackProfile3GPP::pImCnFlag

8.28.2.10 uint32\_t\* LibpackProfile3GPP::pIPv4AddrPref

8.28.2.11 uint16\_t\* LibpackProfile3GPP::pIPv6AddPref

8.28.2.12 uint8\_t\* LibpackProfile3GPP::pPassword

8.28.2.13 uint16\_t\* LibpackProfile3GPP::pPasswordSize

8.28.2.14 uint8\_t\* LibpackProfile3GPP::pPcscfAddrUsingDhcp

8.28.2.15 uint8\_t\* LibpackProfile3GPP::pPcscfAddrUsingPCO

8.28.2.16 uint32\_t\* LibpackProfile3GPP::pPDNInactivTimeout

8.28.2.17 uint8\_t\* LibpackProfile3GPP::pPdpAccessConFlag

8.28.2.18 uint8\_t\* LibpackProfile3GPP::pPdpContext

- 8.28.2.19 `uint8_t*` `LibpackProfile3GPP::pPdpDataCompType`
- 8.28.2.20 `uint8_t*` `LibpackProfile3GPP::pPdpHdrCompType`
- 8.28.2.21 `uint8_t*` `LibpackProfile3GPP::pPDPTtype`
- 8.28.2.22 `uint32_t*` `LibpackProfile3GPP::pPriDNSIPv4AddPref`
- 8.28.2.23 `uint16_t*` `LibpackProfile3GPP::pPriDNSIPv6addpref`
- 8.28.2.24 `uint8_t*` `LibpackProfile3GPP::pPrimaryID`
- 8.28.2.25 `uint8_t*` `LibpackProfile3GPP::pProfileName`
- 8.28.2.26 `uint16_t*` `LibpackProfile3GPP::pProfileNameSize`
- 8.28.2.27 `LibPackQosClassID*` `LibpackProfile3GPP::pQosClassID`
- 8.28.2.28 `uint32_t*` `LibpackProfile3GPP::pSecDNSIPv4AddPref`
- 8.28.2.29 `uint16_t*` `LibpackProfile3GPP::pSecDNSIPv6addpref`
- 8.28.2.30 `uint8_t*` `LibpackProfile3GPP::pSecondaryFlag`
- 8.28.2.31 `LibPackTFTIDParams*` `LibpackProfile3GPP::pTFTID1Params`
- 8.28.2.32 `LibPackTFTIDParams*` `LibpackProfile3GPP::pTFTID2Params`
- 8.28.2.33 `LibPackUMTSQoS*` `LibpackProfile3GPP::pUMTSMinQoS`
- 8.28.2.34 `LibPackUMTSReqQoSSigInd*` `LibpackProfile3GPP::pUMTSMinQoSsigInd`
- 8.28.2.35 `LibPackUMTSQoS*` `LibpackProfile3GPP::pUMTSReqQoS`
- 8.28.2.36 `LibPackUMTSReqQoSSigInd*` `LibpackProfile3GPP::pUMTSReqQoSSigInd`
- 8.28.2.37 `uint8_t*` `LibpackProfile3GPP::pUsername`
- 8.28.2.38 `uint16_t*` `LibpackProfile3GPP::pUsernameSize`

## 8.29 LibpackProfile3GPP2 Struct Reference

### Data Fields

- `uint8_t *` [pNegoDnsSvrPref](#)
- `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 \* pAuthPasswordSize
- uint8\_t \* pDataRate
- uint32\_t \* pAppType
- uint8\_t \* pDataMode
- uint8\_t \* pAppPriority
- uint8\_t \* pApnString
- uint16\_t \* pApnStringSize
- uint8\_t \* pPdnType
- uint8\_t \* plsPscfAddressNedded
- uint32\_t \* pPrimaryV4DnsAddress
- uint32\_t \* pSecondaryV4DnsAddress
- uint16\_t \* pPriV6DnsAddress
- uint16\_t \* pSecV6DnsAddress
- uint8\_t \* pRATType
- uint8\_t \* pAPNEnabled3GPP2
- uint32\_t \* pPDNInactivTimeout3GPP2
- uint8\_t \* pAPNClass3GPP2

### 8.29.1 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</i> ↔ <i>Pref</i>	<ul style="list-style-type: none"> <li>• Negotiate DNS Server Preference             <ul style="list-style-type: none"> <li>– 1 - (TRUE)implies request DNS addresses from the PDSN</li> <li>– 0 - (FALSE)implies do not request DNS addresses from the PDSN</li> <li>– Default value is 1 (TRUE)</li> </ul> </li> </ul>
<i>pPppSess</i> ↔ <i>CloseTimerDO</i>	<ul style="list-style-type: none"> <li>• PPP Session Close Timer for DO             <ul style="list-style-type: none"> <li>– Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down</li> </ul> </li> </ul>

<i>pPppSess↔ CloseTimer1x</i>	<ul style="list-style-type: none"> <li>• PPP Session Close Timer for 1X <ul style="list-style-type: none"> <li>– Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down</li> </ul> </li> </ul>
<i>pAllowLinger</i>	<ul style="list-style-type: none"> <li>• Allow/disallow lingering of interface <ul style="list-style-type: none"> <li>– 1 -(TRUE) implies allow lingering</li> <li>– 0 -(FALSE) implies do not allow lingering</li> </ul> </li> </ul>
<i>pLcpAckTimeout</i>	<ul style="list-style-type: none"> <li>• LCP ACK Timeout <ul style="list-style-type: none"> <li>– Value of LCP ACK Timeout in milliseconds</li> </ul> </li> </ul>
<i>plpcpAck↔ Timeout</i>	<ul style="list-style-type: none"> <li>• IPCP ACK Timeout <ul style="list-style-type: none"> <li>– Value of IPCP ACK Timeout in milliseconds</li> </ul> </li> </ul>
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> <li>• AUTH Timeout <ul style="list-style-type: none"> <li>– Value of Authentication Timeout in milliseconds</li> </ul> </li> </ul>
<i>pLcpCreq↔ RetryCount</i>	<ul style="list-style-type: none"> <li>• LCP Configuration Request Retry Count</li> </ul>
<i>plpcpCreq↔ RetryCount</i>	<ul style="list-style-type: none"> <li>• IPCP Configuration Request Retry Count</li> </ul>
<i>pAuthRetry↔ Count</i>	<ul style="list-style-type: none"> <li>• Authentication Retry Count value</li> </ul>
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> <li>• Authentication Protocol <ul style="list-style-type: none"> <li>– 1 - PAP</li> <li>– 2 - CHAP</li> <li>– 3 - PAP or CHAP</li> </ul> </li> </ul>
<i>pUserId</i>	<ul style="list-style-type: none"> <li>• User ID to be used during data network authentication</li> <li>• maximum length allowed is 127 uint8_ts;</li> <li>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.</li> </ul>
<i>pUserIdSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pAuthPassword</i>	<ul style="list-style-type: none"> <li>• Password to be used during data network authentication;</li> <li>• maximum length allowed is 127 uint8_ts</li> <li>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value.</li> </ul>

<i>pAuth</i> ↔ <i>PasswordSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pDataRate</i>	<ul style="list-style-type: none"> <li>Data Rate Requested <ul style="list-style-type: none"> <li>0 - Low (Low speed Service Options (SO15) only)</li> <li>1 - Medium (SO33 + low R-SCH)</li> <li>2 - High (SO33 + high R-SCH)</li> <li>Default is 2</li> </ul> </li> </ul>
<i>pAppType</i>	<ul style="list-style-type: none"> <li>Application Type: <ul style="list-style-type: none"> <li>0x00000001 - Default Application Type</li> <li>0x00000020 - LBS Application Type</li> <li>0x00000040 - Tethered Application Type</li> <li>This parameter is not used while creating/modifying a profile</li> </ul> </li> </ul>
<i>pDataMode</i>	<ul style="list-style-type: none"> <li>Data Mode to use: <ul style="list-style-type: none"> <li>0 - CDMA or HDR (Hybrid 1X/1xEV-DO)</li> <li>1 - CDMA Only (1X only)</li> <li>2 - HDR Only (1xEV-DO only)</li> <li>Default is 0</li> </ul> </li> </ul>
<i>pAppPriority</i>	<ul style="list-style-type: none"> <li>Application Priority <ul style="list-style-type: none"> <li>Numerical 1 uint8_t value defining the application priority; higher value implies higher priority</li> <li>This parameter is not used while creating/modifying a profile</li> </ul> </li> </ul>
<i>pApnString</i>	<ul style="list-style-type: none"> <li>String representing the Access Point Name</li> <li>maximum length allowed is 100 uint8_ts</li> <li>QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long.</li> </ul>
<i>pApnStringSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pPdnType</i>	<ul style="list-style-type: none"> <li>Packed Data Network Type Requested: <ul style="list-style-type: none"> <li>0 - IPv4 PDN Type</li> <li>1 - IPv6 PDN Type</li> <li>2 - IPv4 or IPv6 PDN Type</li> <li>3 - Unspecified PDN Type (implying no preference)</li> </ul> </li> </ul>
<i>plsPcscf</i> ↔ <i>AddressNedded</i>	<ul style="list-style-type: none"> <li>This boolean value is used to control if PCSCF address is requested from PDSN <ul style="list-style-type: none"> <li>1 -(TRUE) implies request for PCSCF value from the PDSN</li> <li>0 -(FALSE) implies do not request for PCSCF value from the PDSN</li> </ul> </li> </ul>

<i>pPrimaryV4↔ DnsAddress</i>	<ul style="list-style-type: none"> <li>IPv4 Primary DNS address <ul style="list-style-type: none"> <li>The Primary IPv4 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pSecondaryV4↔ DnsAddress</i>	<ul style="list-style-type: none"> <li>IPv4 Secondary DNS address <ul style="list-style-type: none"> <li>The Secondary IPv4 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pPriV6Dns↔ Address</i>	<ul style="list-style-type: none"> <li>Primary IPv6 DNS address <ul style="list-style-type: none"> <li>The Primary IPv6 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pSecV6Dns↔ Address</i>	<ul style="list-style-type: none"> <li>Secondary IPv6 DNS address <ul style="list-style-type: none"> <li>The Secondary IPv6 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pRATType</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t Flag indicating RAT Type</li> <li>Values: <ul style="list-style-type: none"> <li>1 - HRPD</li> <li>2 - EHRPD</li> <li>3 - HRPD_EHRPD</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pAPN↔ Enabled3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t Flag indicating if the APN is disabled/enabled</li> <li>If disabled, the profile can not be used for making data calls</li> <li>Values: <ul style="list-style-type: none"> <li>0 - Disabled</li> <li>1 - Enabled(default value)</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pPDNInactiv↔ Timeout3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 4 uint8_ts indicating the duration of inactivity timer in seconds</li> <li>If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected</li> <li>Default value of zero indicates infinite value</li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pAPNClass3G↔ PP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t numeric identifier representing the APN in profile</li> <li>Can be set and queried but is not used by the modem</li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>

## 8.29.2 Field Documentation

### 8.29.2.1 uint8\_t\* LibpackProfile3GPP2::pAllowLinger

- 8.29.2.2    `uint8_t*` LibpackProfile3GPP2::pAPNClass3GPP2
- 8.29.2.3    `uint8_t*` LibpackProfile3GPP2::pAPNEnabled3GPP2
- 8.29.2.4    `uint8_t*` LibpackProfile3GPP2::pApnString
- 8.29.2.5    `uint16_t*` LibpackProfile3GPP2::pApnStringSize
- 8.29.2.6    `uint8_t*` LibpackProfile3GPP2::pAppPriority
- 8.29.2.7    `uint32_t*` LibpackProfile3GPP2::pAppType
- 8.29.2.8    `uint8_t*` LibpackProfile3GPP2::pAuthPassword
- 8.29.2.9    `uint16_t*` LibpackProfile3GPP2::pAuthPasswordSize
- 8.29.2.10    `uint8_t*` LibpackProfile3GPP2::pAuthProtocol
- 8.29.2.11    `uint8_t*` LibpackProfile3GPP2::pAuthRetryCount
- 8.29.2.12    `uint16_t*` LibpackProfile3GPP2::pAuthTimeout
- 8.29.2.13    `uint8_t*` LibpackProfile3GPP2::pDataMode
- 8.29.2.14    `uint8_t*` LibpackProfile3GPP2::pDataRate
- 8.29.2.15    `uint16_t*` LibpackProfile3GPP2::plpcpAckTimeout
- 8.29.2.16    `uint8_t*` LibpackProfile3GPP2::plpcpCreqRetryCount
- 8.29.2.17    `uint8_t*` LibpackProfile3GPP2::plsPcscfAddressNedded
- 8.29.2.18    `uint16_t*` LibpackProfile3GPP2::pLcpAckTimeout
- 8.29.2.19    `uint8_t*` LibpackProfile3GPP2::pLcpCreqRetryCount
- 8.29.2.20    `uint8_t*` LibpackProfile3GPP2::pNegoDnsSrvrPref
- 8.29.2.21    `uint32_t*` LibpackProfile3GPP2::pPDNInactivTimeout3GPP2
- 8.29.2.22    `uint8_t*` LibpackProfile3GPP2::pPdnType
- 8.29.2.23    `uint32_t*` LibpackProfile3GPP2::pPppSessCloseTimer1x
- 8.29.2.24    `uint32_t*` LibpackProfile3GPP2::pPppSessCloseTimerDO

- 8.29.2.25    `uint32_t*` `LibpackProfile3GPP2::pPrimaryV4DnsAddress`
- 8.29.2.26    `uint16_t*` `LibpackProfile3GPP2::pPriV6DnsAddress`
- 8.29.2.27    `uint8_t*` `LibpackProfile3GPP2::pRATType`
- 8.29.2.28    `uint32_t*` `LibpackProfile3GPP2::pSecondaryV4DnsAddress`
- 8.29.2.29    `uint16_t*` `LibpackProfile3GPP2::pSecV6DnsAddress`
- 8.29.2.30    `uint8_t*` `LibpackProfile3GPP2::pUserId`
- 8.29.2.31    `uint16_t*` `LibpackProfile3GPP2::pUserIdSize`

## 8.30 LibPackprofile\_3GPP Struct Reference

### Data Fields

- `uint8_t *` `pProfilename`
- `uint16_t *` `pProfilenameSize`
- `uint8_t *` `pPDPTtype`
- `uint8_t *` `pPdpHdrCompType`
- `uint8_t *` `pPdpDataCompType`
- `uint8_t *` `pAPNName`
- `uint16_t *` `pAPNnameSize`
- `uint32_t *` `pPriDNSIPv4AddPref`
- `uint32_t *` `pSecDNSIPv4AddPref`
- `LibPackUMTSQoS *` `pUMTSReqQoS`
- `LibPackUMTSQoS *` `pUMTSMinQoS`
- `LibPackGPRSRequestedQoS *` `pGPRSRequestedQos`
- `LibPackGPRSRequestedQoS *` `pGPRSMinimumQoS`
- `uint8_t *` `pUsername`
- `uint16_t *` `pUsernameSize`
- `uint8_t *` `pPassword`
- `uint16_t *` `pPasswordSize`
- `uint8_t *` `pAuthenticationPref`
- `uint32_t *` `pIPv4AddrPref`
- `uint8_t *` `pPcsfAddrUsingPCO`
- `uint8_t *` `pPdpAccessConFlag`
- `uint8_t *` `pPcsfAddrUsingDhcp`
- `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`



### 8.30.1 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"> <li>• One or more bytes describing the profile</li> </ul>
<i>pProfilename</i> ↔ <i>Size;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPDPTType</i>	<ul style="list-style-type: none"> <li>• Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile             <ul style="list-style-type: none"> <li>– 0x00 - PDP-IP (IPv4)</li> <li>– 0x01 - PDP-PPP</li> <li>– 0x02 - PDP-IPV6</li> <li>– 0x03 - PDP-IPV4V6</li> </ul> </li> </ul>
<i>pPdpHdr</i> ↔ <i>CompType</i>	<ul style="list-style-type: none"> <li>• PDP header compression type             <ul style="list-style-type: none"> <li>– 0 - PDP header compression is OFF</li> <li>– 1 - Manufacturer preferred compression</li> <li>– 2 - PDP header compression based on RFC 1144</li> <li>– 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095</li> </ul> </li> </ul>
<i>pPdpData</i> ↔ <i>CompType</i>	<ul style="list-style-type: none"> <li>• PDP data compression type             <ul style="list-style-type: none"> <li>– 0 - PDP data compression is OFF</li> <li>– 1 - Manufacturer preferred compression</li> <li>– 2 - V.42BIS data compression</li> <li>– 3 - V.44 data compression</li> </ul> </li> </ul>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>• Access point name</li> </ul>
<i>pAPNnameSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPriDNSIPv4</i> ↔ <i>AddPref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address Preference</li> </ul>
<i>pSecDNSIPv4</i> ↔ <i>AddPref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 Address Preference</li> </ul>

<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Requested QoS</li> </ul>
<i>pUMTSMinQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Minimum QoS</li> </ul>
<i>pGPRS↔ RequestedQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Minimum QoS</li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>• User name</li> </ul>
<i>pUsernameSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPassword</i>	<ul style="list-style-type: none"> <li>• Password</li> </ul>
<i>pPasswordSize;</i>	<ul style="list-style-type: none"> <li>• This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes.</li> </ul>
<i>p↔ Authentication↔ Pref</i>	<ul style="list-style-type: none"> <li>• Authentication Preference <ul style="list-style-type: none"> <li>– Bit map that indicates the authentication algorithm preference <ul style="list-style-type: none"> <li>* Bit 0 - PAP preference <ul style="list-style-type: none"> <li>• 0 - PAP is never performed</li> <li>• 1 - PAP may be performed</li> </ul> </li> <li>* Bit 1 - CHAP preference <ul style="list-style-type: none"> <li>• 0 - CHAP is never performed</li> <li>• 1 - CHAP may be performed</li> </ul> </li> <li>* If more than one bit is set, then the device decides which authentication procedure is performed while setting up the data session. For example, the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul> </li> </ul>
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> <li>• IPv4 Address Preference</li> </ul>
<i>pPcscfAddr↔ UsingPCO</i>	<ul style="list-style-type: none"> <li>• P-CSCF Address using PCO Flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request PCSCF address using PCO</li> <li>– 0 - (FALSE) implies do not request By default, this value is 0</li> </ul> </li> </ul>
<i>pPdpAccess↔ ConFlag</i>	<ul style="list-style-type: none"> <li>• PDP access control flag <ul style="list-style-type: none"> <li>– 0 - PDP access control none</li> <li>– 1 - PDP access control reject</li> <li>– 2 - PDP access control permission</li> </ul> </li> </ul>

<i>pPcscfAddr</i> ↔ <i>UsingDhcp</i>	<ul style="list-style-type: none"> <li>• P-CSCF address using DHCP <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies Request PCSCF address using DHCP</li> <li>– 0 - (FALSE) implies do not request By default, value is 0</li> </ul> </li> </ul>
<i>pImCnFlag</i>	<ul style="list-style-type: none"> <li>• IM CN flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request IM CN flag for this profile</li> <li>– 0 - (FALSE) implies do not request IM CN flag for this profile</li> </ul> </li> </ul>
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> </ul>
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> </ul>
<i>pPdpContext</i>	<ul style="list-style-type: none"> <li>• PDP context number</li> </ul>
<i>pSecondaryFlag</i>	<ul style="list-style-type: none"> <li>• PDP context secondary flag <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies this is secondary profile</li> <li>– 0 - (FALSE) implies this is not secondary profile</li> </ul> </li> </ul>
<i>pPrimaryID</i>	<ul style="list-style-type: none"> <li>• PDP context primary ID</li> <li>• function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device</li> </ul>
<i>pIPv6AddPref</i>	<ul style="list-style-type: none"> <li>• IPv6 address preference Preferred IPv6 address to be assigned to the TE; actual assigned address is negotiated with the network and may differ from this value; if not specified, the IPv6 address is obtained automatically from the network</li> </ul>
<i>pUMTSReqQoS</i> ↔ <i>SSigInd</i>	<ul style="list-style-type: none"> <li>• UMTS requested QoS with Signalling Indication flag</li> </ul>
<i>pUMTSMinQoS</i> ↔ <i>SSigInd</i>	<ul style="list-style-type: none"> <li>• UMTS minimum QoS with Signalling Indication flag</li> </ul>
<i>pPrimaryDNS</i> ↔ <i>Pv6addpref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv6 address preference <ul style="list-style-type: none"> <li>– The value may be used as a preference during negotiation with the network; if not specified, the wireless device will attempt to obtain the DNS address automatically from the network; the negotiated value is provided to the host via DHCP</li> </ul> </li> </ul>
<i>pSecondaryD</i> ↔ <i>NSIPv6addpref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv6 address preference</li> </ul>

<i>pAddr</i> ↔ <i>AllocationPref</i>	<ul style="list-style-type: none"> <li>DHCP/NAS preference <ul style="list-style-type: none"> <li>This enumerated value may be used to indicate the address allocation preference <ul style="list-style-type: none"> <li>* 0 - NAS signaling is used for address allocation</li> <li>* 1 - DHCP is used for address allocation</li> </ul> </li> </ul> </li> </ul>
<i>pQosClassID</i>	<ul style="list-style-type: none"> <li>3GPP LTE QoS parameters</li> </ul>
<i>pAPNDisabled</i> ↔ <i>Flag</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t Flag indicating if the APN is disabled/enabled</li> <li>If set, the profile can not be used for making data calls</li> <li>Any data call is failed locally</li> <li>Values: <ul style="list-style-type: none"> <li>0 - FALSE(default)</li> <li>1 - True</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pPDNInactiv</i> ↔ <i>Timeout</i>	<ul style="list-style-type: none"> <li>Optional 4 Bytes indicating the duration of inactivity timer in seconds</li> <li>If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected</li> <li>Default value of zero indicates infinite value</li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pAPNClass</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t numeric identifier representing the APN in profile</li> <li>Can be set and queried but is not used by the modem</li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>

## 8.30.2 Field Documentation

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

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

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

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

8.30.2.5 uint16\_t\* LibPackprofile\_3GPP::pAPNNameSize

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

8.30.2.7 LibPackGPRSRequestedQoS\* LibPackprofile\_3GPP::pGPRSMinimumQoS

8.30.2.8 LibPackGPRSRequestedQoS\* LibPackprofile\_3GPP::pGPRSRequestedQoS

8.30.2.9 uint8\_t\* LibPackprofile\_3GPP::pImCnFlag

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

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

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

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

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

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

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

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

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

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

8.30.2.20 uint8\_t\* LibPackprofile\_3GPP::pPdpHdrCompType

8.30.2.21 uint8\_t\* LibPackprofile\_3GPP::pPDPTtype

8.30.2.22 uint32\_t\* LibPackprofile\_3GPP::pPriDNSIPv4AddPref

8.30.2.23 uint16\_t\* LibPackprofile\_3GPP::pPriDNSIPv6addpref

8.30.2.24 uint8\_t\* LibPackprofile\_3GPP::pPrimaryID

8.30.2.25 uint8\_t\* LibPackprofile\_3GPP::pProfileName

8.30.2.26 uint16\_t\* LibPackprofile\_3GPP::pProfileNameSize

8.30.2.27 LibPackQosClassID\* LibPackprofile\_3GPP::pQosClassID

8.30.2.28 uint32\_t\* LibPackprofile\_3GPP::pSecDNSIPv4AddPref

8.30.2.29 uint16\_t\* LibPackprofile\_3GPP::pSecDNSIPv6addpref

- 8.30.2.30 `uint8_t*` `LibPackprofile_3GPP::pSecondaryFlag`
- 8.30.2.31 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID1Params`
- 8.30.2.32 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID2Params`
- 8.30.2.33 `LibPackUMTSQoS*` `LibPackprofile_3GPP::pUMTSMinQoS`
- 8.30.2.34 `LibPackUMTSReqQoSSigInd*` `LibPackprofile_3GPP::pUMTSMinQoSsigInd`
- 8.30.2.35 `LibPackUMTSQoS*` `LibPackprofile_3GPP::pUMTSReqQoS`
- 8.30.2.36 `LibPackUMTSReqQoSSigInd*` `LibPackprofile_3GPP::pUMTSReqQoSsigInd`
- 8.30.2.37 `uint8_t*` `LibPackprofile_3GPP::pUsername`
- 8.30.2.38 `uint16_t*` `LibPackprofile_3GPP::pUsernameSize`

## 8.31 LibPackprofile\_3GPP2 Struct Reference

### Data Fields

- `uint8_t*` `pNegoDnsSrvrPref`
- `uint32_t*` `pPppSessCloseTimerDO`
- `uint32_t*` `pPppSessCloseTimer1x`
- `uint8_t*` `pAllowLinger`
- `uint16_t*` `pLcpAckTimeout`
- `uint16_t*` `pIpccpAckTimeout`
- `uint16_t*` `pAuthTimeout`
- `uint8_t*` `pLcpCreqRetryCount`
- `uint8_t*` `pIpccpCreqRetryCount`
- `uint8_t*` `pAuthRetryCount`
- `uint8_t*` `pAuthProtocol`
- `uint8_t*` `pUserId`
- `uint16_t*` `pUserIdSize`
- `uint8_t*` `pAuthPassword`
- `uint16_t*` `pAuthPassword_tSize`
- `uint8_t*` `pDataRate`
- `uint32_t*` `pAppType`
- `uint8_t*` `pDataMode`
- `uint8_t*` `pAppPriority`
- `uint8_t*` `pApnString`
- `uint16_t*` `pApnStringSize`
- `uint8_t*` `pPdnType`
- `uint8_t*` `pIsPcscfAddressNedded`
- `uint32_t*` `pPrimaryV4DnsAddress`
- `uint32_t*` `pSecondaryV4DnsAddress`
- `uint16_t*` `pPriV6DnsAddress`
- `uint16_t*` `pSecV6DnsAddress`
- `uint8_t*` `pRATType`
- `uint8_t*` `pAPNEnabled3GPP2`
- `uint32_t*` `pPDNInactivTimeout3GPP2`
- `uint8_t*` `pAPNClass3GPP2`

## 8.31.1 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</i> ↔ <i>Pref</i>	<ul style="list-style-type: none"> <li>• Negotiate DNS Server Preference <ul style="list-style-type: none"> <li>– 1 - (TRUE) implies request DNS addresses from the PDSN</li> <li>– 0 - (FALSE) implies do not request DNS addresses from the PDSN</li> <li>– Default value is 1 (TRUE)</li> </ul> </li> </ul>
<i>pPppSess</i> ↔ <i>CloseTimerDO</i>	<ul style="list-style-type: none"> <li>• PPP Session Close Timer for DO <ul style="list-style-type: none"> <li>– Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down</li> </ul> </li> </ul>
<i>pPppSess</i> ↔ <i>CloseTimer1x</i>	<ul style="list-style-type: none"> <li>• PPP Session Close Timer for 1X <ul style="list-style-type: none"> <li>– Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down</li> </ul> </li> </ul>
<i>pAllowLinger</i>	<ul style="list-style-type: none"> <li>• Allow/disallow lingering of interface <ul style="list-style-type: none"> <li>– 1 -(TRUE) implies allow lingering</li> <li>– 0 -(FALSE) implies do not allow lingering</li> </ul> </li> </ul>
<i>pLcpAckTimeout</i>	<ul style="list-style-type: none"> <li>• LCP ACK Timeout <ul style="list-style-type: none"> <li>– Value of LCP ACK Timeout in milliseconds</li> </ul> </li> </ul>
<i>pIpcpAck</i> ↔ <i>Timeout</i>	<ul style="list-style-type: none"> <li>• IPCP ACK Timeout <ul style="list-style-type: none"> <li>– Value of IPCP ACK Timeout in milliseconds</li> </ul> </li> </ul>
<i>pAuthTimeout</i>	<ul style="list-style-type: none"> <li>• AUTH Timeout <ul style="list-style-type: none"> <li>– Value of Authentication Timeout in milliseconds</li> </ul> </li> </ul>
<i>pLcpCreq</i> ↔ <i>RetryCount</i>	<ul style="list-style-type: none"> <li>• LCP Configuration Request Retry Count</li> </ul>
<i>pIpcpCreq</i> ↔ <i>RetryCount</i>	<ul style="list-style-type: none"> <li>• IPCP Configuration Request Retry Count</li> </ul>
<i>pAuthRetry</i> ↔ <i>Count</i>	<ul style="list-style-type: none"> <li>• Authentication Retry Count value</li> </ul>
<i>pAuthProtocol</i>	<ul style="list-style-type: none"> <li>• Authentication Protocol <ul style="list-style-type: none"> <li>– 1 - PAP</li> <li>– 2 - CHAP</li> <li>– 3 - PAP or CHAP</li> </ul> </li> </ul>
Generated by Doxygen	

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



<i>pApnStringSize;</i>	<ul style="list-style-type: none"> <li>This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPdnType</i>	<ul style="list-style-type: none"> <li>Packed Data Network Type Requested: <ul style="list-style-type: none"> <li>0 - IPv4 PDN Type</li> <li>1 - IPv6 PDN Type</li> <li>2 - IPv4 or IPv6 PDN Type</li> <li>3 - Unspecified PDN Type (implying no preference)</li> </ul> </li> </ul>
<i>plsPcscf↔ AddressNedded</i>	<ul style="list-style-type: none"> <li>This boolean value is used to control if PCSCF address is requested from PDSN <ul style="list-style-type: none"> <li>1 -(TRUE) implies request for PCSCF value from the PDSN</li> <li>0 -(FALSE) implies do not request for PCSCF value from the PDSN</li> </ul> </li> </ul>
<i>pPrimaryV4↔ DnsAddress</i>	<ul style="list-style-type: none"> <li>IPv4 Primary DNS address <ul style="list-style-type: none"> <li>The Primary IPv4 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pSecondaryV4↔ DnsAddress</i>	<ul style="list-style-type: none"> <li>IPv4 Secondary DNS address <ul style="list-style-type: none"> <li>The Secondary IPv4 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pPriV6Dns↔ Address</i>	<ul style="list-style-type: none"> <li>Primary IPv6 DNS address <ul style="list-style-type: none"> <li>The Primary IPv6 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pSecV6Dns↔ Address</i>	<ul style="list-style-type: none"> <li>Secondary IPv6 DNS address <ul style="list-style-type: none"> <li>The Secondary IPv6 DNS address that can be statically assigned to the UE</li> </ul> </li> </ul>
<i>pRATType</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t Flag indicating RAT Type</li> <li>Values: <ul style="list-style-type: none"> <li>1 - HRPD</li> <li>2 - EHRPD</li> <li>3 - HRPD_EHRPD</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pAPN↔ Enabled3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t Flag indicating if the APN is disabled/enabled</li> <li>If disabled, the profile can not be used for making data calls</li> <li>Values: <ul style="list-style-type: none"> <li>0 - Disabled</li> <li>1 - Enabled(default value)</li> </ul> </li> <li>This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>

<i>pPDNInactiv↔ Timeout3GPP2</i>	<ul style="list-style-type: none"> <li>• Optional 4 Bytes indicating the duration of inactivity timer in seconds</li> <li>• If the PDP context/PDN connection is inactive for this duration i.e. No data Tx/Rx occurs, the PDP context/PDN connection is disconnected</li> <li>• Default value of zero indicates infinite value</li> <li>• This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>
<i>pAPNClass3G↔ PP2</i>	<ul style="list-style-type: none"> <li>• Optional 1 uint8_t numeric identifier representing the APN in profile</li> <li>• Can be set and queried but is not used by the modem</li> <li>• This parameter is currently read only and can be read by using the function SLQSGetProfile↔Settings().</li> </ul>

## 8.31.2 Field Documentation

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

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

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

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

8.31.2.5 uint16\_t\* LibPackprofile\_3GPP2::pApnStringSize

8.31.2.6 uint8\_t\* LibPackprofile\_3GPP2::pAppPriority

8.31.2.7 uint32\_t\* LibPackprofile\_3GPP2::pAppType

8.31.2.8 uint8\_t\* LibPackprofile\_3GPP2::pAuthPassword

8.31.2.9 uint16\_t\* LibPackprofile\_3GPP2::pAuthPassword\_tSize

8.31.2.10 uint8\_t\* LibPackprofile\_3GPP2::pAuthProtocol

8.31.2.11 uint8\_t\* LibPackprofile\_3GPP2::pAuthRetryCount

8.31.2.12 uint16\_t\* LibPackprofile\_3GPP2::pAuthTimeout

8.31.2.13 uint8\_t\* LibPackprofile\_3GPP2::pDataMode

8.31.2.14 uint8\_t\* LibPackprofile\_3GPP2::pDataRate

8.31.2.15 uint16\_t\* LibPackprofile\_3GPP2::plpcpAckTimeout

- 8.31.2.16 `uint8_t*` `LibPackprofile_3GPP2::plpcpCreqRetryCount`
- 8.31.2.17 `uint8_t*` `LibPackprofile_3GPP2::plsPcscfAddressNedded`
- 8.31.2.18 `uint16_t*` `LibPackprofile_3GPP2::pLcpAckTimeout`
- 8.31.2.19 `uint8_t*` `LibPackprofile_3GPP2::pLcpCreqRetryCount`
- 8.31.2.20 `uint8_t*` `LibPackprofile_3GPP2::pNegoDnsSrvrPref`
- 8.31.2.21 `uint32_t*` `LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2`
- 8.31.2.22 `uint8_t*` `LibPackprofile_3GPP2::pPdnType`
- 8.31.2.23 `uint32_t*` `LibPackprofile_3GPP2::pPppSessCloseTimer1x`
- 8.31.2.24 `uint32_t*` `LibPackprofile_3GPP2::pPppSessCloseTimerDO`
- 8.31.2.25 `uint32_t*` `LibPackprofile_3GPP2::pPrimaryV4DnsAddress`
- 8.31.2.26 `uint16_t*` `LibPackprofile_3GPP2::pPriV6DnsAddress`
- 8.31.2.27 `uint8_t*` `LibPackprofile_3GPP2::pRATType`
- 8.31.2.28 `uint32_t*` `LibPackprofile_3GPP2::pSecondaryV4DnsAddress`
- 8.31.2.29 `uint16_t*` `LibPackprofile_3GPP2::pSecV6DnsAddress`
- 8.31.2.30 `uint8_t*` `LibPackprofile_3GPP2::pUserId`
- 8.31.2.31 `uint16_t*` `LibPackprofile_3GPP2::pUserIdSize`

## 8.32 LibPackQosClassID Struct Reference

### Data Fields

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

### 8.32.1 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"> <li>• QOS specified using the QOS Class Identifier (QOS) values QCI value 0 - Requests the network to assign the appropriate QCI value QCI values 1-4 - Associated with guaranteed bit rates QCI values 5-9 - Associated with non-guaranteed bit rates</li> </ul>
<i>gDIBitRate</i>	<ul style="list-style-type: none"> <li>• Guaranteed DL bit rate</li> </ul>
<i>maxDIBitRate</i>	<ul style="list-style-type: none"> <li>• maxDIBitRate</li> </ul>
<i>gUIBitRate</i>	<ul style="list-style-type: none"> <li>• Guaranteed UL bit rate</li> </ul>
<i>maxUIBitRate</i>	<ul style="list-style-type: none"> <li>• Maximum UL bit rate</li> </ul>

### 8.32.2 Field Documentation

8.32.2.1 `uint8_t LibPackQosClassID::gDIBitRate`

8.32.2.2 `uint32_t LibPackQosClassID::gUIBitRate`

8.32.2.3 `uint32_t LibPackQosClassID::maxDIBitRate`

8.32.2.4 `uint32_t LibPackQosClassID::maxUIBitRate`

8.32.2.5 `uint8_t LibPackQosClassID::QCI`

## 8.33 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`

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

### 8.33.2 Field Documentation

- 8.33.2.1 `uint16_t LibPackTFTIDParams::destPortRangeEnd`
- 8.33.2.2 `uint32_t LibPackTFTIDParams::destPortRangeStart`
- 8.33.2.3 `uint8_t LibPackTFTIDParams::eValid`
- 8.33.2.4 `uint8_t LibPackTFTIDParams::filterId`
- 8.33.2.5 `uint32_t LibPackTFTIDParams::flowLabel`
- 8.33.2.6 `uint32_t LibPackTFTIDParams::IPSECSPI`
- 8.33.2.7 `uint8_t LibPackTFTIDParams::ipVersion`
- 8.33.2.8 `uint8_t LibPackTFTIDParams::nextHeader`
- 8.33.2.9 `uint16_t* LibPackTFTIDParams::pSourceIP`
- 8.33.2.10 `uint8_t LibPackTFTIDParams::sourceIPMask`
- 8.33.2.11 `uint16_t LibPackTFTIDParams::srcPortRangeEnd`
- 8.33.2.12 `uint16_t LibPackTFTIDParams::srcPortRangeStart`
- 8.33.2.13 `uint16_t LibPackTFTIDParams::tosMask`

## 8.34 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`

### 8.34.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

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

### Parameters

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

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

### 8.34.2 Field Documentation

- 8.34.2.1 `uint8_t LibPackUMTSQoS::deliveryErrSDU`
- 8.34.2.2 `uint32_t LibPackUMTSQoS::grntDownlinkBitrate`
- 8.34.2.3 `uint32_t LibPackUMTSQoS::grntUplinkBitrate`
- 8.34.2.4 `uint32_t LibPackUMTSQoS::maxDownlinkBitrate`
- 8.34.2.5 `uint32_t LibPackUMTSQoS::maxSDUSize`
- 8.34.2.6 `uint32_t LibPackUMTSQoS::maxUplinkBitrate`
- 8.34.2.7 `uint8_t LibPackUMTSQoS::qosDeliveryOrder`
- 8.34.2.8 `uint8_t LibPackUMTSQoS::resBerRatio`
- 8.34.2.9 `uint8_t LibPackUMTSQoS::sduErrorRatio`
- 8.34.2.10 `uint8_t LibPackUMTSQoS::trafficClass`
- 8.34.2.11 `uint32_t LibPackUMTSQoS::trafficPriority`
- 8.34.2.12 `uint32_t LibPackUMTSQoS::transferDelay`

## 8.35 LibPackUMTSReqQoSsigInd Struct Reference

### Data Fields

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



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

### 8.35.2 Field Documentation

8.35.2.1 `uint8_t LibPackUMTSReqQoSSigInd::SigInd`

8.35.2.2 `LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS`

## 8.36 loc\_BdsSV Struct Reference

### Data Fields

- `uint16_t id`
- `uint8_t mask`

### 8.36.1 Detailed Description

This structure contains the BDS SV Info

#### Parameters

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

### 8.36.2 Field Documentation

8.36.2.1 `uint16_t loc_BdsSV::id`

8.36.2.2 `uint8_t loc_BdsSV::mask`

## 8.37 `loc_BdsSVInfo` Struct Reference

### Data Fields

- `uint8_t len`
- `loc_BdsSV * pSV`

### 8.37.1 Detailed Description

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

#### Parameters

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

### 8.37.2 Field Documentation

8.37.2.1 `uint8_t loc_BdsSVInfo::len`

8.37.2.2 `loc_BdsSV* loc_BdsSVInfo::pSV`

## 8.38 `loc_CellDb` Struct Reference

### Data Fields

- `uint32_t mask`

### 8.38.1 Detailed Description

This structure contains the cell database

#### Parameters

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

### 8.38.2 Field Documentation

#### 8.38.2.1 uint32\_t loc\_CellDb::mask

## 8.39 loc\_ClkInfo Struct Reference

### Data Fields

- uint32\_t [mask](#)

### 8.39.1 Detailed Description

This structure contains the clock Info

## Parameters

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

## 8.39.2 Field Documentation

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

## 8.40 loc\_GnssData Struct Reference

## Data Fields

- uint64\_t [mask](#)

### 8.40.1 Detailed Description

This structure contains the GNSS data

#### Parameters

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

### 8.40.2 Field Documentation

8.40.2.1 `uint64_t loc_GnssData::mask`

## 8.41 `loc_gpsTime` Struct Reference

### Data Fields

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

### 8.41.1 Detailed Description

This structure contains GPS Time info.

#### Parameters

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

### 8.41.2 Field Documentation

8.41.2.1 `uint32_t loc_gpsTime::gpsTimeOfWeekMs`

8.41.2.2 `uint16_t loc_gpsTime::gpsWeek`

## 8.42 `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](#)

### 8.42.1 Detailed Description

This structure contains the Application Information

#### Parameters

<i>appProvider</i> ↔ <i>Length</i>	<ul style="list-style-type: none"> <li>Length of the Application Provider</li> </ul>
<i>pAppProvider</i>	<ul style="list-style-type: none"> <li>Application Provider</li> <li>Depends upon the Length of application Provider</li> </ul>
<i>appNameLength</i>	<ul style="list-style-type: none"> <li>Length of Application Name</li> </ul>
<i>pAppName</i>	<ul style="list-style-type: none"> <li>Application Name</li> <li>Depends upon the Length of application Name</li> </ul>
<i>appVersionValid</i>	<ul style="list-style-type: none"> <li>Specifies whether the application version string contains a valid value</li> <li>0x00 (FALSE) Application version string is invalid</li> <li>0x01 (TRUE) Application version string is valid</li> </ul>
<i>appVersion</i> ↔ <i>Length</i>	<ul style="list-style-type: none"> <li>Length of Application Version</li> </ul>
<i>pAppVersion</i>	<ul style="list-style-type: none"> <li>Application Version</li> <li>Depends upon the Length of application Version</li> </ul>

## 8.42.2 Field Documentation

8.42.2.1 `uint8_t loc_LocApplicationInfo::appNameLength`

8.42.2.2 `uint8_t loc_LocApplicationInfo::appProviderLength`

8.42.2.3 `uint8_t loc_LocApplicationInfo::appVersionLength`

8.42.2.4 `uint8_t loc_LocApplicationInfo::appVersionValid`

8.42.2.5 `uint8_t* loc_LocApplicationInfo::pAppName`

8.42.2.6 `uint8_t* loc_LocApplicationInfo::pAppProvider`

8.42.2.7 `uint8_t* loc_LocApplicationInfo::pAppVersion`

## 8.43 loc\_precisionDilution Struct Reference

### Data Fields

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

### 8.43.1 Detailed Description

This structure contains Dilution of precision associated with this position.

#### Parameters

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

### 8.43.2 Field Documentation

8.43.2.1 `uint32_t loc_precisionDilution::HDOP`

8.43.2.2 `uint32_t loc_precisionDilution::PDOP`

8.43.2.3 `uint32_t loc_precisionDilution::VDOP`

## 8.44 `loc_satelliteInfo` Struct Reference

#### Data Fields

- `uint8_t svListLen`
- `uint32_t validMask`
- `uint32_t system`
- `uint16_t gnssSvId`
- `uint8_t healthStatus`
- `uint32_t svStatus`
- `uint8_t svInfoMask`
- `float elevation`
- `float azimuth`
- `float snr`

### 8.44.1 Detailed Description

Contain fields in struct `loc_satelliteInfo`

#### Parameters



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

<i>svInfoMask</i>	<ul style="list-style-type: none"> <li>Indicates whether almanac and ephemeris information is available. Valid bitmasks: <ul style="list-style-type: none"> <li>0x01 - SVINFO_HAS_EPHEMERIS</li> <li>0x02 - SVINFO_HAS_ALMANAC</li> </ul> </li> </ul>
<i>elevation</i>	<ul style="list-style-type: none"> <li>SV elevation angle. <ul style="list-style-type: none"> <li>Units: Degrees</li> <li>Range: 0 to 90</li> </ul> </li> </ul>
<i>azimuth</i>	<ul style="list-style-type: none"> <li>SV azimuth angle. <ul style="list-style-type: none"> <li>Units: Degrees</li> <li>Range: 0 to 360</li> </ul> </li> </ul>
<i>snr</i>	<ul style="list-style-type: none"> <li>SV signal-to-noise ratio <ul style="list-style-type: none"> <li>Units: dB-Hz</li> </ul> </li> </ul>

### 8.44.2 Field Documentation

8.44.2.1 float loc\_satelliteInfo::azimuth

8.44.2.2 float loc\_satelliteInfo::elevation

8.44.2.3 uint16\_t loc\_satelliteInfo::gnssSvId

8.44.2.4 uint8\_t loc\_satelliteInfo::healthStatus

8.44.2.5 float loc\_satelliteInfo::snr

8.44.2.6 uint8\_t loc\_satelliteInfo::svInfoMask

8.44.2.7 uint8\_t loc\_satelliteInfo::svListLen

8.44.2.8 uint32\_t loc\_satelliteInfo::svStatus

8.44.2.9 uint32\_t loc\_satelliteInfo::system

8.44.2.10 uint32\_t loc\_satelliteInfo::validMask

## 8.45 loc\_sensorDataUsage Struct Reference

### Data Fields

- uint32\_t [usageMask](#)
- uint32\_t [aidingIndicatorMask](#)

### 8.45.1 Detailed Description

This structure contains Sensor Data Usage info.

#### Parameters

<i>usageMask</i>	<ul style="list-style-type: none"><li>• Specifies which sensors were used in calculating the position in the position report.</li></ul>
------------------	---

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

#### Parameters

<i>aidingIndicator↔ Mask</i>	
----------------------------------	--

- Specifies which results were aided by sensors.
- Value
  - 0x00000001 - AIDED\_HEADING
  - 0x00000002 - AIDED\_SPEED
  - 0x00000004 - AIDED\_POSITION
  - 0x00000008 - AIDED\_VELOCITY

### 8.45.2 Field Documentation

8.45.2.1 `uint32_t loc_sensorDataUsage::aidingIndicatorMask`

8.45.2.2 `uint32_t loc_sensorDataUsage::usageMask`

## 8.46 loc\_SV Struct Reference

### Data Fields

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

### 8.46.1 Detailed Description

This structure contains the Delete LOC SV Info

## Parameters

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

## 8.46.2 Field Documentation

8.46.2.1 `uint16_t loc_SV::id`8.46.2.2 `uint8_t loc_SV::mask`8.46.2.3 `uint32_t loc_SV::system`8.47 `loc_SVInfo` Struct Reference

## Data Fields

- `uint8_t len`
- `loc_SV * pSV`

## 8.47.1 Detailed Description

This structure contains the elements of Delete LOC SV Info

## Parameters

<i>len</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements in struct <a href="#">loc_SV</a>: <ul style="list-style-type: none"> <li>– <code>gnssSvId</code></li> <li>– <code>system</code></li> <li>– <code>deleteSvInfoMask</code></li> </ul> </li> </ul>
<i>pSV</i>	

## 8.47.2 Field Documentation

8.47.2.1 `uint8_t loc_SVInfo::len`

8.47.2.2 `loc_SV* loc_SVInfo::pSV`

## 8.48 loc\_svUsedforFix Struct Reference

### Data Fields

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

### 8.48.1 Detailed Description

This structure contains SVs Used to Calculate the Fix.

#### Parameters

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

## 8.48.2 Field Documentation

8.48.2.1 `uint16_t loc_svUsedforFix::gnssSvUsedList[255]`

8.48.2.2 `uint8_t loc_svUsedforFix::gnssSvUsedList_len`

## 8.49 IteSSInfo Struct Reference

### Data Fields

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

### 8.49.1 Detailed Description

#### Parameters

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

### 8.49.2 Field Documentation

8.49.2.1 `int16_t lteSSInfo::rsrp`

8.49.2.2 `int8_t lteSSInfo::rsrq`

8.49.2.3 `int8_t lteSSInfo::rssI`

8.49.2.4 `int16_t lteSSInfo::snr`

## 8.50 messageModeTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `sMSMessageModelInfo MessageModelInfo`

### 8.50.1 Detailed Description

#### Parameters

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

### 8.50.2 Field Documentation

8.50.2.1 `sMSMessageModelInfo messageModeTlv::MessageModelInfo`

8.50.2.2 `uint8_t messageModeTlv::TlvPresent`

## 8.51 nas\_acqOrderPref Struct Reference

## Data Fields

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

## 8.51.1 Detailed Description

Contain the Acquisition Order Preference.

## Parameters

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

## 8.51.2 Field Documentation

8.51.2.1 uint8\_t nas\_acqOrderPref::acqOrdeLen

8.51.2.2 uint8\_t\* nas\_acqOrderPref::pAcqOrder

## 8.52 nas\_AddCDMASysInfo Struct Reference

## Data Fields

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

## 8.52.1 Detailed Description

Structure for storing the Additional CDMA System Information.

## Parameters

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

## 8.52.2 Field Documentation

8.52.2.1 uint16\_t nas\_AddCDMASysInfo::geoSysIdx

8.52.2.2 uint16\_t nas\_AddCDMASysInfo::regPrd

## 8.53 nas\_AddSysInfo Struct Reference

### Data Fields

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

### 8.53.1 Detailed Description

Structure for storing the Additional GSM and WCDMA System Information.

#### Parameters

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

## 8.53.2 Field Documentation

8.53.2.1 uint32\_t nas\_AddSysInfo::cellBroadcastCap

8.53.2.2 uint16\_t nas\_AddSysInfo::geoSysIdx

## 8.54 nas\_CallBarringSysInfo Struct Reference

### Data Fields

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



### 8.54.1 Detailed Description

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

#### Parameters

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

### 8.54.2 Field Documentation

8.54.2.1 `uint32_t nas_CallBarringSysInfo::csBarStatus`

8.54.2.2 `uint32_t nas_CallBarringSysInfo::psBarStatus`

## 8.55 nas\_callBarStatus Struct Reference

### Data Fields

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

### 8.55.1 Detailed Description

This structure contains Call Barring Status.

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

## Parameters

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

## 8.55.2 Field Documentation

8.55.2.1 uint32\_t nas\_callBarStatus::csBarStatus

8.55.2.2 uint32\_t nas\_callBarStatus::psBarStatus

## 8.56 nas\_CDMAECIOThresh Struct Reference

## Data Fields

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

## 8.56.1 Detailed Description

This structure contains CDMA ECIO threshold related parameters.

## Parameters

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

## 8.56.2 Field Documentation

8.56.2.1 `uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen`

8.56.2.2 `int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList`

## 8.57 nas\_CDMAInfo Struct Reference

### Data Fields

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

### 8.57.1 Detailed Description

This structure contains information about the CDMA Network.

#### Parameters

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

## 8.57.2 Field Documentation

8.57.2.1 `uint16_t nas_CDMAInfo::baseId`

8.57.2.2 `uint32_t nas_CDMAInfo::baseLat`

8.57.2.3 `uint32_t nas_CDMAInfo::baseLong`

8.57.2.4 `uint16_t nas_CDMAInfo::nid`

8.57.2.5 `uint16_t nas_CDMAInfo::refpn`

8.57.2.6 `uint16_t nas_CDMAInfo::sid`

## 8.58 nas\_CDMARSSIThresh Struct Reference

### Data Fields

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

### 8.58.1 Detailed Description

This structure contains CDMA RSSI threshold related parameters.

#### Parameters

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

## 8.58.2 Field Documentation

8.58.2.1 `uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen`

8.58.2.2 `int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList`

## 8.59 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]

### 8.59.1 Detailed Description

Structure for storing the CDMA System Information.

#### Parameters

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

<i>pRevInUse</i>	<ul style="list-style-type: none"> <li>• The lesser of the base station P_Rev and mobile P_Rev</li> <li>• Only applicable for CDMA. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>bsPRevValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the base station P_Rev is valid <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>bsPRev</i>	<ul style="list-style-type: none"> <li>• Base station P_Rev.</li> <li>• Only applicable for CDMA. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>ccsSupported</i> ↔ <i>Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the supported concurrent service is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>ccsSupported</i>	<ul style="list-style-type: none"> <li>• Whether concurrent service is supported.</li> <li>• Only applicable for CDMA. <ul style="list-style-type: none"> <li>– 0x00 - Not supported</li> <li>– 0x01 - Supported</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cdmaSysIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the CDMA system ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>systemID</i>	<ul style="list-style-type: none"> <li>• System ID. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>networkID</i>	<ul style="list-style-type: none"> <li>• Network ID. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>bsInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the base station information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>baseLat</i>	<ul style="list-style-type: none"> <li>• Base station latitude in units of 0.25 sec.</li> <li>• Expressed as a two's complement signed number with positive numbers signifying North latitudes. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>baseLong</i>	<ul style="list-style-type: none"> <li>• Base station longitude in units of 0.25 sec.</li> <li>• Expressed as a two's complement signed number with positive numbers signifying East latitudes. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>packetZoneValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the packet zone is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>packetZone</i>	<ul style="list-style-type: none"> <li>• Packet zone (8-bit). <ul style="list-style-type: none"> <li>– 0xFFFF indicates no packet zone.</li> </ul> </li> <li>• Only applicable for CDMA.</li> </ul>
<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_LEN-1:0]</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code.</li> <li>• MCC digits in ASCII characters</li> <li>• MCC wildcard value is returned as {'3', 0xFF, 0xFF}.</li> </ul>
<i>MNC[PLMN_LEN-1:0]</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code.</li> <li>• MNC digits in ASCII characters</li> <li>• An unused byte is set to 0xFF.</li> <li>• MNC wildcard value is returned as {'7', 0xFF, 0xFF}.</li> </ul>

## 8.59.2 Field Documentation

8.59.2.1 uint16\_t nas\_CDMASysInfo::baseId

8.59.2.2 uint32\_t nas\_CDMASysInfo::baseLat

8.59.2.3 uint32\_t nas\_CDMASysInfo::baseLong

8.59.2.4 uint8\_t nas\_CDMASysInfo::bsInfoValid

- 8.59.2.5    `uint8_t nas_CDMASysInfo::bsPRev`
- 8.59.2.6    `uint8_t nas_CDMASysInfo::bsPRevValid`
- 8.59.2.7    `uint8_t nas_CDMASysInfo::ccsSupported`
- 8.59.2.8    `uint8_t nas_CDMASysInfo::ccsSupportedValid`
- 8.59.2.9    `uint8_t nas_CDMASysInfo::cdmaSysIdValid`
- 8.59.2.10   `uint8_t nas_CDMASysInfo::isSysPrIMatch`
- 8.59.2.11   `uint8_t nas_CDMASysInfo::isSysPrIMatchValid`
- 8.59.2.12   `uint8_t nas_CDMASysInfo::MCC[3]`
- 8.59.2.13   `uint8_t nas_CDMASysInfo::MNC[3]`
- 8.59.2.14   `uint16_t nas_CDMASysInfo::networkID`
- 8.59.2.15   `uint8_t nas_CDMASysInfo::networkIdValid`
- 8.59.2.16   `uint16_t nas_CDMASysInfo::packetZone`
- 8.59.2.17   `uint8_t nas_CDMASysInfo::packetZoneValid`
- 8.59.2.18   `uint8_t nas_CDMASysInfo::pRevInUse`
- 8.59.2.19   `uint8_t nas_CDMASysInfo::pRevInUseValid`
- 8.59.2.20   `nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA`
- 8.59.2.21   `uint16_t nas_CDMASysInfo::systemID`

## 8.60 nas\_CDMASysInfoExt Struct Reference

### Data Fields

- `uint16_t` [MCC](#)
- `uint8_t` [imsi\\_11\\_12](#)

### 8.60.1 Detailed Description

This structure contains CDMA system information extension

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

### Parameters



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

## 8.60.2 Field Documentation

8.60.2.1 uint8\_t nas\_CDMASysInfoExt::imsi\_11\_12

8.60.2.2 uint16\_t nas\_CDMASysInfoExt::MCC

## 8.61 nas\_cellParams Struct Reference

### Data Fields

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

### 8.61.1 Detailed Description

This structure contains information about the Cell parameters.

#### Parameters

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

## 8.61.2 Field Documentation

8.61.2.1 `uint16_t nas_cellParams::pci`

8.61.2.2 `int16_t nas_cellParams::rsrp`

8.61.2.3 `int16_t nas_cellParams::rsrq`

8.61.2.4 `int16_t nas_cellParams::rssi`

8.61.2.5 `int16_t nas_cellParams::srxlev`

## 8.62 `nas_CommInfo` Struct Reference

### Data Fields

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

### 8.62.1 Detailed Description

Structure for storing the common information for the device.

#### Parameters

<i>temperature</i>	<ul style="list-style-type: none"> <li>• Temperature. <ul style="list-style-type: none"> <li>– 8-bit signed integer</li> <li>– 0xFF - Not Available.</li> </ul> </li> </ul>
<i>modemMode</i>	<ul style="list-style-type: none"> <li>• Modem Operating Mode. <ul style="list-style-type: none"> <li>– 0x00 - POWERING OFF</li> <li>– 0x01 - FACTORY TEST</li> <li>– 0x02 - OFFLINE</li> <li>– 0x03 - OFFLINE_AMPS</li> <li>– 0x04 - OFFLINE_CDMA</li> <li>– 0x05 - ONLINE</li> <li>– 0x06 - LOW POWER MODE</li> <li>– 0x07 - RESETTING</li> <li>– 0x08 - NETWORK TEST</li> <li>– 0x09 - OFFLINE REQUEST</li> <li>– 0x0A - PSEUDO ONLINE</li> <li>– 0x0B - RESETTING MODEM</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>

<i>systemMode</i>	<ul style="list-style-type: none"> <li>• System Acquisition Mode. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - AMPS</li> <li>– 0x02 - CDMA</li> <li>– 0x03 - GSM</li> <li>– 0x04 - HDR</li> <li>– 0x05 - WCDMA</li> <li>– 0x06 - GPS</li> <li>– 0x08 - WLAN</li> <li>– 0x09 - LTE</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>imsRegState</i>	<ul style="list-style-type: none"> <li>• IMS Registration State. <ul style="list-style-type: none"> <li>– 0x00 - NO SRV</li> <li>– 0x01 - IN PROG</li> <li>– 0x02 - FAILED</li> <li>– 0x03 - LIMITED</li> <li>– 0x04 - FULL SRV</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>psState</i>	<ul style="list-style-type: none"> <li>• PS Attach State. <ul style="list-style-type: none"> <li>– 0x00 - Attached</li> <li>– 0x01 - Detached</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>

## 8.62.2 Field Documentation

8.62.2.1 `uint8_t nas_CommInfo::imsRegState`

8.62.2.2 `uint8_t nas_CommInfo::modemMode`

8.62.2.3 `uint8_t nas_CommInfo::psState`

8.62.2.4 `uint8_t nas_CommInfo::systemMode`

8.62.2.5 `int8_t nas_CommInfo::temperature`

## 8.63 nas\_CSGID Struct Reference

### Data Fields

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

### 8.63.1 Detailed Description

Contain the CSGID.

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"> <li>MCC value. Range 0 to 999</li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>MNC value. Range 0 to 999</li> </ul>
<i>mncPcsDigits</i>	<ul style="list-style-type: none"> <li>TRUE - MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090</li> <li>FALSE - MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90</li> </ul>
<i>id</i>	<ul style="list-style-type: none"> <li>Closed subscriber group identifier.</li> </ul>
<i>rat</i>	<ul style="list-style-type: none"> <li>Radio interface technology of the CSG network. Values: <ul style="list-style-type: none"> <li>0x04 - RADIO_IF_GSM - GSM</li> <li>0x05 - RADIO_IF_UMTS - UMTS</li> <li>0x08 - RADIO_IF_LTE - LTE</li> <li>0x09 - RADIO_IF_TDSCDMA - TDS</li> </ul> </li> </ul>

### 8.63.2 Field Documentation

8.63.2.1 `uint32_t nas_CSGID::id`

8.63.2.2 `uint16_t nas_CSGID::mcc`

8.63.2.3 `uint16_t nas_CSGID::mnc`

8.63.2.4 `uint8_t nas_CSGID::mncPcsDigits`

8.63.2.5 `uint8_t nas_CSGID::rat`

## 8.64 nas\_currentPLMN Struct Reference

#### Data Fields

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

### 8.64.1 Detailed Description

This structure contains the current PLMN parameters

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

#### Parameters

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

### 8.64.2 Field Documentation

8.64.2.1 `uint16_t nas_currentPLMN::MCC`

8.64.2.2 `uint16_t nas_currentPLMN::MNC`

8.64.2.3 `uint8_t nas_currentPLMN::netDescr[255]`

8.64.2.4 `uint8_t nas_currentPLMN::netDescrLength`

## 8.65 nas\_dataSrvCapabilities Struct Reference

### Data Fields

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

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

## 8.65.2 Field Documentation

8.65.2.1 `uint8_t nas_dataSrvCapabilities::dataCapabilities[32]`

8.65.2.2 `uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen`

## 8.66 nas\_detailSvcInfo Struct Reference

### Data Fields

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

### 8.66.1 Detailed Description

This structure contains Detailed Service information

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

### Parameters

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

## 8.66.2 Field Documentation

8.66.2.1 `uint8_t nas_detailSvcInfo::hdrHybrid`

8.66.2.2 `uint8_t nas_detailSvcInfo::hdrSrvStatus`

8.66.2.3 `uint8_t nas_detailSvcInfo::isSysForbidden`

8.66.2.4 `uint8_t nas_detailSvcInfo::srvCapability`

8.66.2.5 `uint8_t nas_detailSvcInfo::srvStatus`

## 8.67 `nas_ecioListElement` Struct Reference

### Data Fields

- `int16_t ecio`
- `uint8_t radiolf`

### 8.67.1 Detailed Description

This structure contains the ECIO Information

#### Parameters

<i>ecio</i>	<ul style="list-style-type: none"><li>• ECIO value in dBm</li></ul>
<i>radiolf</i>	<ul style="list-style-type: none"><li>• Radio interface technology of the signal being measured<ul style="list-style-type: none"><li>– 0x00 – RADIO_IF_NO_SVC – None (no service)</li><li>– 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X</li><li>– 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000 HRPD (1xEV-DO)</li><li>– 0x03 – RADIO_IF_AMPS – AMPS</li><li>– 0x04 – RADIO_IF_GSM – GSM</li><li>– 0x05 – RADIO_IF_UMTS – UMTS</li></ul></li></ul>

### 8.67.2 Field Documentation

8.67.2.1 `int16_t nas_ecioListElement::ecio`

8.67.2.2 `uint8_t nas_ecioListElement::radiolf`

## 8.68 `nas_errorRateListElement` Struct Reference

### Data Fields

- `uint16_t errorRate`
- `uint8_t radiolf`

### 8.68.1 Detailed Description

This structure contains the Error Rate Information

#### Parameters



<i>errorRate</i>	<ul style="list-style-type: none"> <li>• Error rate value corresponds to the RAT that is currently registered. <ul style="list-style-type: none"> <li>– For CDMA, the error rate reported is Frame Error Rate: <ul style="list-style-type: none"> <li>* Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3%</li> <li>* A value of 0xFFFF indicates that the error rate is unknown or unavailable</li> </ul> </li> <li>– For HDR, the error rate reported is Packet Error Rate: <ul style="list-style-type: none"> <li>* Valid error rate values between 1 and 10000 are returned to indicate percentage, e.g., a value of 300 means the error rate is 3%</li> <li>* A value of 0xFFFF indicates that the error rate is unknown or unavailable</li> </ul> </li> <li>– For GSM, the error rate reported is Bit Error Rate: <ul style="list-style-type: none"> <li>* Valid values are 0, 100, 200, 300, 400, 500, 600, and 700 The reported value divided by 100 gives the error rate as an RxQual value, e.g., a value of 300 represents an RxQual value of 3.</li> <li>* A value of 25500 indicates No Data</li> </ul> </li> <li>– For WCDMA, the error rate reported is Block Error Rate (BLER): <ul style="list-style-type: none"> <li>* Valid values are 1 to 10000</li> <li>* The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3%.</li> <li>* A value of 0 indicates No Data</li> </ul> </li> </ul> </li> </ul>
<i>radioIrf</i>	<ul style="list-style-type: none"> <li>• Radio interface technology of the signal being measured <ul style="list-style-type: none"> <li>– 0x00 – RADIO_IF_NO_SVC – None (no service)</li> <li>– 0x01 – RADIO_IF_CDMA_1X – cdma2000@ 1X</li> <li>– 0x02 – RADIO_IF_CDMA_1xEVDO – cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 – RADIO_IF_AMPS – AMPS</li> <li>– 0x04 – RADIO_IF_GSM – GSM</li> <li>– 0x05 – RADIO_IF_UMTS – UMTS</li> </ul> </li> </ul>

## 8.68.2 Field Documentation

8.68.2.1 `uint16_t nas_errorRateListElement::errorRate`

8.68.2.2 `uint8_t nas_errorRateListElement::radioIrf`

## 8.69 nas\_GERANInfo Struct Reference

### Data Fields

- `uint32_t cellID`
- `uint8_t plmn` [3]
- `uint16_t lac`
- `uint16_t arfcn`
- `uint8_t bsic`
- `uint32_t timingAdvance`
- `uint16_t rxLev`
- `uint8_t nmrInst`
- `nas_nmrCellInfo insNmrCellInfo` [255]

### 8.69.1 Detailed Description

This structure contains information about the GERAN Network.

#### Parameters

<i>cellID</i>	<ul style="list-style-type: none"> <li>Cell ID.</li> <li>0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>plmn</i> [ <i>NAS_PL</i> ↔ <i>MN_LENGTH</i> ]	<ul style="list-style-type: none"> <li>MCC/MNC information coded as octet 3, 4, and 5.</li> <li>This field is ignored when <i>nmrCellID</i> is not present.</li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>Location area code.</li> <li>This field is ignored when <i>nmrCellID</i> is not present. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>arfcn</i>	<ul style="list-style-type: none"> <li>Absolute RF channel number. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>bsic</i>	<ul style="list-style-type: none"> <li>Base station identity code. <ul style="list-style-type: none"> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>timingAdvance</i>	<ul style="list-style-type: none"> <li>Measured delay (in bit periods; 1 bit period = 48/13 microsecond) of access burst transmission on RACH or PRACH to the expected signal from an MS at zero distance under static channel conditions. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>rxLev</i>	<ul style="list-style-type: none"> <li>Serving Cell Rx measurement.</li> <li>Values range between 0 and 63.</li> <li>Mapped to a measured signal level: <ul style="list-style-type: none"> <li>Rxlev 0 is a signal strength less than -110 dBm</li> <li>Rxlev 1 is -110 dBm to -109 dBm</li> <li>Rxlev 2 is -109 dBm to -108 dBm</li> <li>...</li> <li>Rxlev 62 is -49 dBm to -48 dBm</li> <li>Rxlev 63 is greater than -48 dBm</li> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>nmrInst</i>	<ul style="list-style-type: none"> <li>Provides the number of set of instances which follow.</li> <li>If 0(zero), then no information follows it.</li> </ul>
<i>insNmrCell</i> ↔ <i>Info</i> [ <i>NAS_MA</i> ↔ <i>X_DESCRIPTOR</i> ↔ <i>ON_LENGTH</i> ]	<ul style="list-style-type: none"> <li>See <a href="#">nas_nmrCellInfo</a> for more information.</li> </ul>

## 8.69.2 Field Documentation

8.69.2.1 uint16\_t nas\_GERANInfo::arfcn

8.69.2.2 uint8\_t nas\_GERANInfo::bsic

8.69.2.3 uint32\_t nas\_GERANInfo::cellID

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

8.69.2.5 uint16\_t nas\_GERANInfo::lac

8.69.2.6 uint8\_t nas\_GERANInfo::nmrInst

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

8.69.2.8 uint16\_t nas\_GERANInfo::rxLev

8.69.2.9 uint32\_t nas\_GERANInfo::timingAdvance

## 8.70 nas\_geranInstInfo Struct Reference

### Data Fields

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

### 8.70.1 Detailed Description

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

#### Parameters

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

## 8.70.2 Field Documentation

8.70.2.1 uint16\_t nas\_geranInstInfo::geranArfcn

8.70.2.2 uint8\_t nas\_geranInstInfo::geranBsicBcc

8.70.2.3 uint8\_t nas\_geranInstInfo::geranBsicNcc

8.70.2.4 int16\_t nas\_geranInstInfo::geranRssi

## 8.71 nas\_gsmCellInfo Struct Reference

### Data Fields

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

### 8.71.1 Detailed Description

This structure contains information about the GSM Cell.

#### Parameters

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

### 8.71.2 Field Documentation

8.71.2.1 `uint16_t nas_gsmCellInfo::arfcn`

8.71.2.2 `uint8_t nas_gsmCellInfo::band1900`

8.71.2.3 `uint8_t nas_gsmCellInfo::bsicld`

8.71.2.4 `uint8_t nas_gsmCellInfo::cellldValid`

8.71.2.5 `int16_t nas_gsmCellInfo::rssi`

8.71.2.6 `int16_t nas_gsmCellInfo::srxlev`

## 8.72 nas\_GSMRSSIThresh Struct Reference

### Data Fields

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

### 8.72.1 Detailed Description

This structure contains GSM RSSI threshold related parameters.

#### Parameters

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

### 8.72.2 Field Documentation

8.72.2.1 `uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen`

8.72.2.2 `int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList`

## 8.73 nas\_GSMsrvStatusInfo Struct Reference

### Data Fields

- `uint8_t` [srvStatus](#)

- [uint8\\_t trueSrvStatus](#)
- [uint8\\_t isPrefDataPath](#)

### 8.73.1 Detailed Description

Structure for storing the service status information for GSM, WCDMA and LTE networks.

#### Parameters

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

### 8.73.2 Field Documentation

8.73.2.1 [uint8\\_t nas\\_GSMSrvStatusInfo::isPrefDataPath](#)

8.73.2.2 [uint8\\_t nas\\_GSMSrvStatusInfo::srvStatus](#)

8.73.2.3 [uint8\\_t nas\\_GSMSrvStatusInfo::trueSrvStatus](#)

## 8.74 nas\_GSMSysInfo Struct Reference

#### Data Fields

- [nas\\_sysInfoCommon sysInfoGSM](#)

- uint8\_t [lacValid](#)
- uint16\_t [lac](#)
- uint8\_t [cellIdValid](#)
- uint32\_t [cellId](#)
- uint8\_t [regRejectInfoValid](#)
- uint8\_t [rejectSrvDomain](#)
- uint8\_t [rejCause](#)
- uint8\_t [networkIdValid](#)
- uint8\_t [MCC](#) [3]
- uint8\_t [MNC](#) [3]
- uint8\_t [egprsSuppValid](#)
- uint8\_t [egprsSupp](#)
- uint8\_t [dtmSuppValid](#)
- uint8\_t [dtmSupp](#)

### 8.74.1 Detailed Description

Structure for storing the GSM System Information.

#### Parameters

<i>sysInfoGSM</i>	<ul style="list-style-type: none"> <li>• See sysInfoCommon for more information.</li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• Only applies to 3GPP. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>• Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejCause</i>	<ul style="list-style-type: none"> <li>• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_L↔ENGTH]</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code.</li> <li>• MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_L↔ENGTH]</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code.</li> <li>• MNC digits in ASCII characters</li> <li>• An unused byte is set to 0xFF.</li> <li>• In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.</li> </ul>
<i>egprsSuppValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the EGPRS support is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>egprsSupp</i>	<ul style="list-style-type: none"> <li>• EGPRS support indication.</li> <li>• Only applicable for GSM. <ul style="list-style-type: none"> <li>– 0x00 - Not available</li> <li>– 0x01 - Available</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>dtmSuppValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether Dual Transfer mode support is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>



<i>dtmSupp</i>	<ul style="list-style-type: none"> <li>• Dual Transfer mode support indication.</li> <li>• Only applicable for GSM. <ul style="list-style-type: none"> <li>– 0x00 - Not available</li> <li>– 0x01 - Available</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
----------------	--

## 8.74.2 Field Documentation

8.74.2.1 `uint32_t nas_GSMSysInfo::cellId`

8.74.2.2 `uint8_t nas_GSMSysInfo::cellIdValid`

8.74.2.3 `uint8_t nas_GSMSysInfo::dtmSupp`

8.74.2.4 `uint8_t nas_GSMSysInfo::dtmSuppValid`

8.74.2.5 `uint8_t nas_GSMSysInfo::egprsSupp`

8.74.2.6 `uint8_t nas_GSMSysInfo::egprsSuppValid`

8.74.2.7 `uint16_t nas_GSMSysInfo::lac`

8.74.2.8 `uint8_t nas_GSMSysInfo::lacValid`

8.74.2.9 `uint8_t nas_GSMSysInfo::MCC[3]`

8.74.2.10 `uint8_t nas_GSMSysInfo::MNC[3]`

8.74.2.11 `uint8_t nas_GSMSysInfo::networkIdValid`

8.74.2.12 `uint8_t nas_GSMSysInfo::regRejectInfoValid`

8.74.2.13 `uint8_t nas_GSMSysInfo::rejCause`

8.74.2.14 `uint8_t nas_GSMSysInfo::rejectSrvDomain`

8.74.2.15 `nas_sysInfoCommon nas_GSMSysInfo::sysInfoGSM`

## 8.75 nas\_HDRECIOTresh Struct Reference

### Data Fields

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

### 8.75.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

#### Parameters

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

## 8.75.2 Field Documentation

8.75.2.1 `uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen`

8.75.2.2 `int16_t* nas_HDRECIOThresh::pHDRECIOThreshList`

## 8.76 nas\_HDRIOThresh Struct Reference

### Data Fields

- `uint8_t` [HDRIOThreshListLen](#)
- `int16_t *` [pHDRIOThreshList](#)

### 8.76.1 Detailed Description

This structure contains HDR IO threshold related parameters.

#### Parameters

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

## 8.76.2 Field Documentation

8.76.2.1 `uint8_t nas_HDRIOThresh::HDRIOThreshListLen`

8.76.2.2 `int16_t* nas_HDRIOThresh::pHDRIOThreshList`

## 8.77 nas\_HDRRSSIThresh Struct Reference

## Data Fields

- uint8\_t [HDRRSSIThreshListLen](#)
- int16\_t \* [pHDRRSSIThreshList](#)

### 8.77.1 Detailed Description

This structure contains HDR RSSI threshold related parameters.

#### Parameters

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

### 8.77.2 Field Documentation

8.77.2.1 uint8\_t nas\_HDRRSSIThresh::HDRRSSIThreshListLen

8.77.2.2 int16\_t\* nas\_HDRRSSIThresh::pHDRRSSIThreshList

## 8.78 nas\_HDRSINRThreshold Struct Reference

## Data Fields

- uint8\_t [HDRSINRThreshListLen](#)
- uint16\_t \* [pHDRSINRThreshList](#)

### 8.78.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

#### Parameters

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

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

## 8.78.2 Field Documentation

8.78.2.1 `uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen`

8.78.2.2 `uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList`

## 8.79 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]`

### 8.79.1 Detailed Description

Structure for storing the HDR System Information.

## Parameters

<i>sysInfoHDR</i>	<ul style="list-style-type: none"> <li>• See sysInfoCommon for more information.</li> </ul>
<i>isSysPrIMatch</i> ↔ <i>Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the system PRL match is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>isSysPrIMatch</i>	<ul style="list-style-type: none"> <li>• Indicates whether the system is in a PRL.</li> <li>• Only applies to CDMA/HDR. <ul style="list-style-type: none"> <li>– 0x00 - System is not in a PRL</li> <li>– 0x01 - System is in a PRL</li> <li>– 0xFF - Not Available</li> </ul> </li> <li>• If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL.</li> <li>• If the system is in a PRL, roam_status is set to the value based on the standard specification.</li> </ul>
<i>hdrPersonality</i> ↔ <i>Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the HDR personality is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>hdrPersonality</i>	<ul style="list-style-type: none"> <li>• HDR personality information.</li> <li>• Only applicable for HDR. <ul style="list-style-type: none"> <li>– 0x00 - None</li> <li>– 0x02 - HRPD</li> <li>– 0x03 - eHRPD</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>hdrActiveProt</i> ↔ <i>Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the HDR active protocol revision information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>hdrActiveProt</i>	<ul style="list-style-type: none"> <li>• HDR active protocol revision information .</li> <li>• Only applicable for HDR. <ul style="list-style-type: none"> <li>– 0x00 - None</li> <li>– 0x02 - HDR Rel 0</li> <li>– 0x03 - HDR Rel A</li> <li>– 0x04 - HDR Rel B</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>is856SysIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the IS-856 system ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>is856SysId[SL↔ QS_SYSTEM↔ _ID_SIZE]</i>	<ul style="list-style-type: none"> <li>IS-856 system ID.</li> <li>Only applicable for HDR.</li> </ul>

## 8.79.2 Field Documentation

8.79.2.1 `uint8_t nas_HDRSysInfo::hdrActiveProt`

8.79.2.2 `uint8_t nas_HDRSysInfo::hdrActiveProtValid`

8.79.2.3 `uint8_t nas_HDRSysInfo::hdrPersonality`

8.79.2.4 `uint8_t nas_HDRSysInfo::hdrPersonalityValid`

8.79.2.5 `uint8_t nas_HDRSysInfo::is856SysId[16]`

8.79.2.6 `uint8_t nas_HDRSysInfo::is856SysIdValid`

8.79.2.7 `uint8_t nas_HDRSysInfo::isSysPrIMatch`

8.79.2.8 `uint8_t nas_HDRSysInfo::isSysPrIMatchValid`

8.79.2.9 `nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR`

## 8.80 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]`

### 8.80.1 Detailed Description

This structure contains information about the inter-frequency.

### Parameters

<i>earfcn</i>	<ul style="list-style-type: none"> <li>E-UTRA absolute radio frequency channel number of the serving cell.</li> <li>Range: 0 to 65535.</li> </ul>
<i>threshXLow</i>	<ul style="list-style-type: none"> <li>Cell Srxlev low threshold.</li> <li>Range: 0 to 31.</li> <li>When the serving cell does not exceed thresh_serving_low, the value of an evaluated cell must be smaller than this value to be considered for re-selection.</li> </ul>
<i>threshXHigh</i>	<ul style="list-style-type: none"> <li>Cell Srxlev high threshold.</li> <li>Range: 0 to 31.</li> <li>When the serving cell exceeds thresh_serving_low, the value of an evaluated cell must be greater than this value to be considered for re-selection.</li> </ul>
<i>cell_resel_↔ priority</i>	<ul style="list-style-type: none"> <li>Cell re-selection priority</li> <li>Range: 0 to 7.</li> <li>This field is only valid when ue_in_idle is TRUE.</li> </ul>
<i>cells_len</i>	<ul style="list-style-type: none"> <li>Provides the number of set of cell params.</li> </ul>
<i>cellInterFreq↔ Params[NAS ↔ MAX_DESCR↔ PTION_LEN↔ TH]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_cellParams</a> for more information.</li> </ul>

## 8.80.2 Field Documentation

8.80.2.1 `uint8_t nas_infoInterFreq::cell_resel_priority`

8.80.2.2 `nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]`

8.80.2.3 `uint8_t nas_infoInterFreq::cells_len`

8.80.2.4 `uint16_t nas_infoInterFreq::earfcn`

8.80.2.5 `uint8_t nas_infoInterFreq::threshXHigh`

8.80.2.6 `uint8_t nas_infoInterFreq::threshXLow`

## 8.81 nas\_IteGsmCellInfo Struct Reference

### Data Fields

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

### 8.81.1 Detailed Description

This structure contains information about the LTE GSM Cell.

#### Parameters

<i>cellReselPriority</i>	<ul style="list-style-type: none"> <li>• Priority of this frequency group.</li> <li>• Range: 0 to 7.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>threshGsmHigh</i>	<ul style="list-style-type: none"> <li>• Reselection threshold for high priority layers.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>threshGsmLow</i>	<ul style="list-style-type: none"> <li>• Reselection threshold for low priority layers.</li> <li>• Range: 0 to 31.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>nccPermitted</i>	<ul style="list-style-type: none"> <li>• Bitmask specifying whether a neighbor with a specific network color code is to be reported.</li> <li>• Range: 0 to 255.</li> <li>• Bit <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.</li> <li>• This field is only valid when <code>ue_in_idle</code> is TRUE.</li> </ul>
<i>cells_len</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of gsm cells.</li> </ul>
<i>GsmCellInfo</i> [ <i>N</i> ↔ <i>AS_MAX_DE</i> ↔ <i>SCRIPTION_L</i> ↔ <i>ENGTH</i> ]	<ul style="list-style-type: none"> <li>• See <a href="#">nas_gsmCellInfo</a> for more information.</li> </ul>

### 8.81.2 Field Documentation

8.81.2.1 `uint8_t nas_lteGsmCellInfo::cellReselPriority`

8.81.2.2 `uint8_t nas_lteGsmCellInfo::cells_len`

8.81.2.3 `nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]`

8.81.2.4 `uint8_t nas_lteGsmCellInfo::nccPermitted`

8.81.2.5 `uint8_t nas_lteGsmCellInfo::threshGsmHigh`

8.81.2.6 `uint8_t nas_lteGsmCellInfo::threshGsmLow`



## 8.82 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](#)

### 8.82.1 Detailed Description

Structure for storing the LTE information for the device.

#### Parameters

<i>band</i>	<ul style="list-style-type: none"> <li>• LTE Band <ul style="list-style-type: none"> <li>– 1 ~ 40 (Band in decimal)</li> <li>– 0xFF - Invalid</li> </ul> </li> </ul>
<i>bandwidth</i>	<ul style="list-style-type: none"> <li>• BandWidth. <ul style="list-style-type: none"> <li>– 0x00 - 1.4 MHz</li> <li>– 0x01 - 3 MHz</li> <li>– 0x02 - 5 MHz</li> <li>– 0x03 - 10 MHz</li> <li>– 0x04 - 15 MHz</li> <li>– 0x05 - 20 MHz</li> <li>– 0x06 - Invalid</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
<i>RXChan</i>	<ul style="list-style-type: none"> <li>• RX channel number in decimal <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>TXChan</i>	<ul style="list-style-type: none"> <li>• TX channel number in decimal <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>emmState</i>	<ul style="list-style-type: none"> <li>• EMM State. <ul style="list-style-type: none"> <li>– 0x00 - Deregistered</li> <li>– 0x01 - Reg Initiated</li> <li>– 0x02 - Registered</li> <li>– 0x03 - TAU Initiated</li> <li>– 0x04 - SR Initiated</li> <li>– 0x05 - Dereg Initiated</li> <li>– 0x06 - Invalid</li> <li>– 0xFF - Unknown</li> </ul> </li> </ul>
Generated by Doxygen	

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

## 8.82.2 Field Documentation

8.82.2.1 uint8\_t nas\_LTEInfo::band

8.82.2.2 uint8\_t nas\_LTEInfo::bandwidth

8.82.2.3 uint8\_t nas\_LTEInfo::emmConnState

8.82.2.4 uint8\_t nas\_LTEInfo::emmState

8.82.2.5 uint8\_t nas\_LTEInfo::emmSubState

8.82.2.6 uint16\_t nas\_LTEInfo::RXChan

8.82.2.7 uint16\_t nas\_LTEInfo::TXChan

## 8.83 nas\_LTEInfoInterfreq Struct Reference

### Data Fields

- uint8\_t [ueIdle](#)
- uint8\_t [freqsLen](#)
- [nas\\_infoInterFreq](#) [InfoInterfreq](#) [255]

### 8.83.1 Detailed Description

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

#### Parameters

<i>ueIdle</i>	<ul style="list-style-type: none"> <li>• TRUE if the UE is in Idle mode, otherwise FALSE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>freqsLen</i>	<ul style="list-style-type: none"> <li>• Provides the number of set of inter frequency information.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>InfoInterfreq[N↔ AS_MAX_DE↔ SCRIPTION_L↔ ENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_infoInterFreq</a> for more information.</li> </ul>

### 8.83.2 Field Documentation

8.83.2.1 uint8\_t nas\_LTEInfoInterfreq::freqsLen

8.83.2.2 nas\_infoInterFreq nas\_LTEInfoInterfreq::InfoInterfreq[255]

8.83.2.3 uint8\_t nas\_LTEInfoInterfreq::ueIdle

## 8.84 nas\_LTEInfoIntrafreq Struct Reference

### Data Fields

- uint8\_t [ueIdle](#)
- 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]

### 8.84.1 Detailed Description

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

#### Parameters

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

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

## 8.84.2 Field Documentation

8.84.2.1 `nas_cellParams` `nas_LTEInfoIntrafreq::CellParams[255]`

8.84.2.2 `uint8_t` `nas_LTEInfoIntrafreq::cellReselPriority`

8.84.2.3 `uint8_t` `nas_LTEInfoIntrafreq::cellsLen`

8.84.2.4 `uint16_t` `nas_LTEInfoIntrafreq::earfcn`

8.84.2.5 `uint32_t` `nas_LTEInfoIntrafreq::globalCellId`

8.84.2.6 `uint8_t` `nas_LTEInfoIntrafreq::plmn[3]`

8.84.2.7 `uint16_t` `nas_LTEInfoIntrafreq::servingCellId`

8.84.2.8 `uint8_t` `nas_LTEInfoIntrafreq::sIntraSearch`

8.84.2.9 `uint8_t` `nas_LTEInfoIntrafreq::sNonIntraSearch`

8.84.2.10 `uint16_t` `nas_LTEInfoIntrafreq::tac`

8.84.2.11 `uint8_t` `nas_LTEInfoIntrafreq::threshServingLow`

8.84.2.12 `uint8_t nas_LTEInfoIntrafreq::ueIdle`

## 8.85 nas\_LTEInfoNeighboringGSM Struct Reference

### Data Fields

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

### 8.85.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

#### Parameters

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

### 8.85.2 Field Documentation

8.85.2.1 `uint8_t nas_LTEInfoNeighboringGSM::freqsLen`

8.85.2.2 `nas_LteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo`[255]

8.85.2.3 `uint8_t nas_LTEInfoNeighboringGSM::ueIdle`

## 8.86 nas\_LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

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

### 8.86.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

#### Parameters

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

### 8.86.2 Field Documentation

8.86.2.1 `uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen`

8.86.2.2 `nas_IteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]`

8.86.2.3 `uint8_t nas_LTEInfoNeighboringWCDMA::ueIdle`

## 8.87 nas\_IteRsrpinformation Struct Reference

### Data Fields

- `int16_t rsrplevel`

### 8.87.1 Detailed Description

This structure contains the LTE RSRP Information

#### Parameters

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

### 8.87.2 Field Documentation

8.87.2.1 `int16_t nas_lteRsrpInformation::rsrplevel`

## 8.88 `nas_LTE SRPThresh` Struct Reference

### Data Fields

- `uint8_t LTERSRPThreshListLen`
- `int16_t * pLTERSRPThreshList`

### 8.88.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

#### Parameters

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

### 8.88.2 Field Documentation

8.88.2.1 `uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen`

8.88.2.2 `int16_t* nas_LTERSRPThresh::pLTERSRPThreshList`

## 8.89 `nas_LTE SRQThresh` Struct Reference

### Data Fields

- `uint8_t LTERSRQThreshListLen`
- `int16_t * pLTERSRQThreshList`

### 8.89.1 Detailed Description

This structure contains LTE RSRQ threshold related parameters.

#### Parameters

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



## 8.89.2 Field Documentation

8.89.2.1 `uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen`

8.89.2.2 `int16_t* nas_LTERSRQThresh::pLTERSRQThreshList`

## 8.90 nas\_LTERSSIThresh Struct Reference

### Data Fields

- `uint8_t LTERSSIThreshListLen`
- `int16_t* pLTERSSIThreshList`

### 8.90.1 Detailed Description

This structure contains LTE RSSI threshold related parameters.

#### Parameters

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

## 8.90.2 Field Documentation

8.90.2.1 `uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen`

8.90.2.2 `int16_t* nas_LTERSSIThresh::pLTERSSIThreshList`

## 8.91 nas\_LTESigRptConfig Struct Reference

### Data Fields

- `uint8_t rptRate`
- `uint8_t avgPeriod`

### 8.91.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

#### Parameters

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

## 8.91.2 Field Documentation

8.91.2.1 `uint8_t nas_LTESigRptConfig::avgPeriod`

8.91.2.2 `uint8_t nas_LTESigRptConfig::rptRate`

## 8.92 `nas_IteSnr`information Struct Reference

### Data Fields

- `int16_t snrlevel`

### 8.92.1 Detailed Description

This structure contains the LTE SNR Information

#### Parameters

<i>snrlevel</i>	<ul style="list-style-type: none"> <li>• LTE SNR level as a scaled integer in units of 0.1dB e.g. -16dB has a value of -160 and 24.6dB has value of 246.</li> </ul>
-----------------	---

### 8.92.2 Field Documentation

8.92.2.1 `int16_t nas_lteSnrinformation::snrlevel`

## 8.93 nas\_LTESNRThreshold Struct Reference

### Data Fields

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

### 8.93.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

#### Parameters

<i>LTESNR<sub>↔</sub></i> <i>ThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE SNR threshold list parameter to follow</li> </ul>
<i>pLTESNR<sub>↔</sub></i> <i>ThreshList</i>	<ul style="list-style-type: none"> <li>• Array of SNR thresholds (in units of 0.1 dB)</li> <li>• Maximum of 32 values</li> <li>• Range for SNR values: -20 to 30 (in dB).</li> </ul>

### 8.93.2 Field Documentation

8.93.2.1 `uint8_t nas_LTESNRThreshold::LTESNRThreshListLen`

8.93.2.2 `int16_t* nas_LTESNRThreshold::pLTESNRThreshList`

## 8.94 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](#)

### 8.94.1 Detailed Description

Structure for storing the LTE System Information.

#### Parameters

<i>sysInfoLTE</i>	<ul style="list-style-type: none"> <li>• See sysInfoCommon for more information.</li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• Only applies to 3GPP. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>• Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> <li>– 0x04 - Camped</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>rejCause</i>	<ul style="list-style-type: none"> <li>• Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_LEN-1:0]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_LEN-1:0]</i>	<ul style="list-style-type: none"> <li>Mobile Network Code.</li> <li>MNC digits in ASCII characters</li> <li>An unused byte is set to 0xFF.</li> <li>In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.</li> </ul>
<i>tacValid</i>	<ul style="list-style-type: none"> <li>Indicates whether tracking area code is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>tac</i>	<ul style="list-style-type: none"> <li>Tracking area code.</li> <li>Only applicable for LTE. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>

## 8.94.2 Field Documentation

8.94.2.1 `uint32_t nas_LTESysInfo::cellId`

8.94.2.2 `uint8_t nas_LTESysInfo::cellIdValid`

8.94.2.3 `uint16_t nas_LTESysInfo::lac`

8.94.2.4 `uint8_t nas_LTESysInfo::lacValid`

8.94.2.5 `uint8_t nas_LTESysInfo::MCC[3]`

8.94.2.6 `uint8_t nas_LTESysInfo::MNC[3]`

8.94.2.7 `uint8_t nas_LTESysInfo::networkIdValid`

8.94.2.8 `uint8_t nas_LTESysInfo::regRejectInfoValid`

8.94.2.9 `uint8_t nas_LTESysInfo::rejCause`

8.94.2.10 `uint8_t nas_LTESysInfo::rejectSrvDomain`

8.94.2.11 `nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE`

8.94.2.12 `uint16_t nas_LTESysInfo::tac`

8.94.2.13 `uint8_t nas_LTESysInfo::tacValid`

## 8.95 `nas_lteWcdmaCellInfo` Struct Reference

### Data Fields

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

### 8.95.1 Detailed Description

This structure contains information about the LTE WCDMA Cell.

#### Parameters

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

## 8.95.2 Field Documentation

8.95.2.1 `uint8_t nas_lteWcdmaCellInfo::cellReselPriority`

8.95.2.2 `uint8_t nas_lteWcdmaCellInfo::cellsLen`

8.95.2.3 `uint16_t nas_lteWcdmaCellInfo::threshXhigh`

8.95.2.4 `uint16_t nas_lteWcdmaCellInfo::threshXlow`

8.95.2.5 `uint16_t nas_lteWcdmaCellInfo::uarfcn`

8.95.2.6 `nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]`

## 8.96 nas\_MNRInfo Struct Reference

### Data Fields

- `uint16_t mcc`
- `uint16_t mnc`
- `uint32_t rat`

### 8.96.1 Detailed Description

Structure contains Manual Network Register Information parameters

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"><li>• A 16-bit integer representation of Mobile Country Code. Range - 0 to 999.</li></ul>
<i>mnc</i>	<ul style="list-style-type: none"><li>• A 16-bit integer representation of Mobile Network Code. Range - 0 to 999.</li></ul>
<i>rat</i>	<ul style="list-style-type: none"><li>• Radio access technology for which to register.<ul style="list-style-type: none"><li>– 0x04 - RADIO_IF_GSM</li><li>– 0x05 - RADIO_IF_UMTS</li><li>– 0x08 - RADIO_IF_LTE</li></ul></li></ul>

## 8.96.2 Field Documentation

8.96.2.1 `uint16_t nas_MNRInfo::mcc`

8.96.2.2 `uint16_t nas_MNRInfo::mnc`

8.96.2.3 `uint32_t nas_MNRInfo::rat`

## 8.97 `nas_netSelectionPref` Struct Reference

### Data Fields

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

### 8.97.1 Detailed Description

Contain the network selection preference.

#### Parameters

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

### 8.97.2 Field Documentation

8.97.2.1 `uint16_t nas_netSelectionPref::mcc`

8.97.2.2 `uint16_t nas_netSelectionPref::mnc`

8.97.2.3 `uint8_t nas_netSelectionPref::netReg`

## 8.98 `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`



### 8.98.1 Detailed Description

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

#### Parameters

<i>nmrCellID</i>	<ul style="list-style-type: none"> <li>Cell ID.</li> <li>0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>nmrPlmn[NAS← _PLMN_LEN← GTH]</i>	<ul style="list-style-type: none"> <li>MCC/MNC information coded as octet 3, 4, and 5.</li> <li>This field is ignored when nmrCellID is not present.</li> </ul>
<i>nmrLac</i>	<ul style="list-style-type: none"> <li>Location area code.</li> <li>This field is ignored when nmrCellID is not present. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>nmrArfcn</i>	<ul style="list-style-type: none"> <li>Absolute RF channel number. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>
<i>nmrBsic</i>	<ul style="list-style-type: none"> <li>Base station identity code. <ul style="list-style-type: none"> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>nmrRxLev</i>	<ul style="list-style-type: none"> <li>Cell Rx measurement.</li> <li>Values range between 0 and 63.</li> <li>Mapped to a measured signal level: <ul style="list-style-type: none"> <li>Rxlev 0 is a signal strength less than -110 dBm</li> <li>Rxlev 1 is -110 dBm to -109 dBm</li> <li>Rxlev 2 is -109 dBm to -108 dBm</li> <li>...</li> <li>Rxlev 62 is -49 dBm to -48 dBm</li> <li>Rxlev 63 is greater than -48 dBm</li> <li>0xFFFF - Not Available</li> </ul> </li> </ul>

### 8.98.2 Field Documentation

8.98.2.1 uint16\_t nas\_nmrCellInfo::nmrArfcn

8.98.2.2 uint8\_t nas\_nmrCellInfo::nmrBsic

8.98.2.3 uint32\_t nas\_nmrCellInfo::nmrCellID

8.98.2.4 uint16\_t nas\_nmrCellInfo::nmrLac

8.98.2.5 `uint8_t nas_nmrCellInfo::nmrPlmn[3]`

8.98.2.6 `uint16_t nas_nmrCellInfo::nmrRxLev`

## 8.99 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`

### 8.99.1 Detailed Description

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

#### Parameters

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

### 8.99.2 Field Documentation

8.99.2.1 `NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggPcellInfo::dl_bw_value`

8.99.2.2 `uint16_t nas_PhyCaAggPcellInfo::freq`

8.99.2.3 `uint16_t nas_PhyCaAggPcellInfo::iLTEbandValue`

8.99.2.4 uint16\_t nas\_PhyCaAggPcellInfo::pci

8.99.2.5 uint8\_t nas\_PhyCaAggPcellInfo::TlvPresent

## 8.100 nas\_PhyCaAggScellDIBw Struct Reference

### Data Fields

- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE dl\\_bw\\_value](#)
- uint8\_t [TlvPresent](#)

### 8.100.1 Detailed Description

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

#### Parameters

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

### 8.100.2 Field Documentation

8.100.2.1 [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) nas\_PhyCaAggScellDIBw::dl\_bw\_value

8.100.2.2 uint8\_t nas\_PhyCaAggScellDIBw::TlvPresent

## 8.101 nas\_PhyCaAggScellIndex Struct Reference

### Data Fields

- uint8\_t [scell\\_idx](#)
- uint8\_t [TlvPresent](#)

### 8.101.1 Detailed Description

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

#### Parameters

<i>scell_idx</i>	<ul style="list-style-type: none"><li>• Physical cell ID of the SCell Range.</li><li>• Range for ID values: 0 to 503.</li></ul>
<i>TlvPresent</i>	
Generated by Doxygen	<ul style="list-style-type: none"><li>• Tlv Present.</li></ul>

### 8.101.2 Field Documentation

8.101.2.1 `uint8_t nas_PhyCaAggScellIndex::scell_idx`

8.101.2.2 `uint8_t nas_PhyCaAggScellIndex::TlvPresent`

## 8.102 nas\_PhyCaAggScellIndType Struct Reference

### Data Fields

- `uint16_t pci`
- `uint16_t freq`
- `NAS_LTE_CPHY_SCELL_STATE_LITE scell_state`
- `uint8_t TlvPresent`

### 8.102.1 Detailed Description

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

#### Parameters

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

### 8.102.2 Field Documentation

8.102.2.1 `uint16_t nas_PhyCaAggScellIndType::freq`

8.102.2.2 `uint16_t nas_PhyCaAggScellIndType::pci`

8.102.2.3 `NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellIndType::scell_state`

8.102.2.4 `uint8_t nas_PhyCaAggScellIndType::TlvPresent`

## 8.103 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](#)

### 8.103.1 Detailed Description

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

#### Parameters

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

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

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

## 8.103.2 Field Documentation

8.103.2.1 NAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE nas\_PhyCaAggScellInfo::dl\_bw\_value

8.103.2.2 uint16\_t nas\_PhyCaAggScellInfo::freq

8.103.2.3 uint16\_t nas\_PhyCaAggScellInfo::ltebandValue

8.103.2.4 uint16\_t nas\_PhyCaAggScellInfo::pci

8.103.2.5 NAS\_LTE\_CPHY\_SCELL\_STATE\_LITE nas\_PhyCaAggScellInfo::scell\_state

8.103.2.6 uint8\_t nas\_PhyCaAggScellInfo::TlvPresent

## 8.104 nas\_qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- uint8\_t [leapSeconds](#)
- uint8\_t [localTimeOffset](#)
- uint8\_t [daylightSavings](#)

### 8.104.1 Detailed Description

This structure contains the 3GPP2TimeZone parameters

#### Parameters

<i>leapSeconds</i>	<ul style="list-style-type: none"><li>• leap seconds - Number of leap seconds since the start of CDMA system time.</li></ul>
<i>localTimeOffset</i>	<ul style="list-style-type: none"><li>• Local Time Offset - Offset of system time in units of 30 minutes; the value in this field conveys as 8 bit 2's compliment number.</li></ul>
<i>daylightSavings</i>	<ul style="list-style-type: none"><li>• Day Light Savings Indicator<ul style="list-style-type: none"><li>– 0x00 - OFF (daylight savings not in effect)</li><li>– 0x01 - ON (daylight savings in effect)</li></ul></li></ul>

### 8.104.2 Field Documentation

8.104.2.1 `uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings`

8.104.2.2 `uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds`

8.104.2.3 `uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset`

## 8.105 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]

### 8.105.1 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

### 8.105.2 Field Documentation

8.105.2.1 `char nas_QmiNas3GppNetworkInfo::Description[255]`

8.105.2.2 `uint32_t nas_QmiNas3GppNetworkInfo::Forbidden`

8.105.2.3 `uint32_t nas_QmiNas3GppNetworkInfo::InUse`

8.105.2.4 `uint16_t nas_QmiNas3GppNetworkInfo::MCC`

8.105.2.5 `uint16_t nas_QmiNas3GppNetworkInfo::MNC`

8.105.2.6 `uint32_t nas_QmiNas3GppNetworkInfo::Preferred`



8.105.2.7 uint32\_t nas\_QmiNas3GppNetworkInfo::Roaming

## 8.106 nas\_QmiNas3GppNetworkRAT Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [RAT](#)

### 8.106.1 Detailed Description

Contain the 3GPP radio access technology information.

#### Parameters

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

### 8.106.2 Field Documentation

8.106.2.1 uint16\_t nas\_QmiNas3GppNetworkRAT::MCC

8.106.2.2 uint16\_t nas\_QmiNas3GppNetworkRAT::MNC

8.106.2.3 uint8\_t nas\_QmiNas3GppNetworkRAT::RAT

## 8.107 nas\_QmisNasPcsDigit Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [includes\\_pcs\\_digit](#)

### 8.107.1 Detailed Description

Contain the PCS Digit information

#### Parameters

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

### 8.107.2 Field Documentation

8.107.2.1 `uint8_t nas_QmisNasPcsDigit::includes_pcs_digit`

8.107.2.2 `uint16_t nas_QmisNasPcsDigit::MCC`

8.107.2.3 `uint16_t nas_QmisNasPcsDigit::MNC`

## 8.108 nas\_RejectReasonTlv Struct Reference

### Data Fields

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

### 8.108.1 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

### 8.108.2 Field Documentation

8.108.2.1 `uint32_t nas_RejectReasonTlv::rejectCause`

8.108.2.2 `uint32_t nas_RejectReasonTlv::serviceDomain`

8.108.2.3 uint8\_t nas\_RejectReasonTlv::TlvPresent

## 8.109 nas\_RFInfoTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [radioInterfaceSize](#)
- uint32\_t [radioInterface](#) [255]
- uint32\_t [activeBandClass](#) [255]
- uint32\_t [activeChannel](#) [255]

### 8.109.1 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

### 8.109.2 Field Documentation

8.109.2.1 uint32\_t nas\_RFInfoTlv::activeBandClass[255]

8.109.2.2 uint32\_t nas\_RFInfoTlv::activeChannel[255]

8.109.2.3 uint32\_t nas\_RFInfoTlv::radioInterface[255]

8.109.2.4 uint8\_t nas\_RFInfoTlv::radioInterfaceSize

8.109.2.5 uint8\_t nas\_RFInfoTlv::TlvPresent

## 8.110 nas\_roamIndList Struct Reference

### Data Fields

- uint8\_t [numInstances](#)
- uint8\_t [radioInterface](#) [32]
- uint8\_t [roamIndicator](#) [32]

### 8.110.1 Detailed Description

This structure contains the Roaming Indicator List

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

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• number of sets of radio interface currently in use and roaming indicator <ul style="list-style-type: none"> <li>– defaults to zero</li> </ul> </li> </ul>
<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio Interface currently in use</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - RADIO_IF_AMPS - AMPS</li> <li>– 0x04 - RADIO_IF_GSM - GSM</li> <li>– 0x05 - RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - RADIO_IF_LTE - LTE</li> </ul> </li> </ul>
<i>roamIndicator</i>	<ul style="list-style-type: none"> <li>• Roaming Indicator</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Roaming</li> <li>– 0x01 - Home</li> </ul> </li> </ul>

## 8.110.2 Field Documentation

8.110.2.1 `uint8_t nas_roamIndList::numInstances`

8.110.2.2 `uint8_t nas_roamIndList::radioInterface[32]`

8.110.2.3 `uint8_t nas_roamIndList::roamIndicator[32]`

## 8.111 nas\_rsrqInformation Struct Reference

### Data Fields

- `int8_t rsrq`
- `uint8_t radiolf`

### 8.111.1 Detailed Description

This structure contains the RSRQ Information

#### Parameters

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

### 8.111.2 Field Documentation

8.111.2.1 `uint8_t nas_rsrqInformation::radiolf`

8.111.2.2 `int8_t nas_rsrqInformation::rsrq`

## 8.112 nas\_RxSigInfo Struct Reference

### Data Fields

- `uint8_t rxChainIndex`
- `uint8_t isRadioTuned`
- `int32_t rxPower`
- `int32_t rsrp`

### 8.112.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

#### Parameters

<i>rxChainIndex</i>	<ul style="list-style-type: none"><li>• Rx antenna path</li><li>• Valid Values<ul style="list-style-type: none"><li>– 0 - Primary Rx</li><li>– 1 - Diversity Rx</li></ul></li></ul>
<i>isRadioTuned</i>	<ul style="list-style-type: none"><li>• Rx path is tuned to a channel or Not</li><li>• Values<ul style="list-style-type: none"><li>– 0x00 - Not tuned</li><li>– 0x01 - Tuned</li></ul></li></ul>

#### Note

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

#### Parameters

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

### 8.112.2 Field Documentation

8.112.2.1 `uint8_t nas_RxSigInfo::isRadioTuned`

8.112.2.2 `int32_t nas_RxSigInfo::rsrp`

8.112.2.3 `uint8_t nas_RxSigInfo::rxChainIndex`

8.112.2.4 `int32_t nas_RxSigInfo::rxPower`

## 8.113 `nas_rxSignalStrengthListElement` Struct Reference

### Data Fields

- `int16_t rxSignalStrength`
- `uint8_t radiolf`

### 8.113.1 Detailed Description

This structure contains the Received Signal Strength Information

#### Parameters

<i>rxSignalStrength</i>	<ul style="list-style-type: none"> <li>Received signal strength in dBm             <ul style="list-style-type: none"> <li>For CDMA and UMTS, this indicates forward link pilotEc.</li> <li>For GSM, the received signal strength.</li> <li>For LTE, this indicates the total received wideband power observed by UE.</li> </ul> </li> </ul>
<i>radiolf</i>	<ul style="list-style-type: none"> <li>Radio interface technology of the signal being radio_if measured             <ul style="list-style-type: none"> <li>0x00 – RADIO_IF_NO_SVC – None (no service)</li> <li>0x01 – RADIO_IF_CDMA_1X – cdma2000@ 1X</li> <li>0x02 – RADIO_IF_CDMA_1XEVD0 – cdma2000 HRPD (1xEV-DO)</li> <li>0x03 – RADIO_IF_AMPS – AMPS</li> <li>0x04 – RADIO_IF_GSM – GSM</li> <li>0x05 – RADIO_IF_UMTS – UMTS</li> <li>0x08 – RADIO_IF_LTE – LTE</li> </ul> </li> </ul>

#### Note

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

### 8.113.2 Field Documentation

8.113.2.1 `uint8_t nas_rxSignalStrengthListElement::radiolf`

8.113.2.2 int16\_t nas\_rxSignalStrengthListElement::rxSignalStrength

## 8.114 nas\_SccRxInfo Struct Reference

### Data Fields

- int32\_t [rsrq](#)
- int16\_t [snr](#)
- uint8\_t [numInstances](#)
- [nas\\_RxSigInfo sigInfo](#) [255]
- uint8\_t [TlvPresent](#)

### 8.114.1 Detailed Description

This structure contains information about the SccRxInfo parameters.

#### Parameters

<i>rsrq</i>	<ul style="list-style-type: none"> <li>• Current reference signal</li> <li>• Receive quality in 1/10 dB resolution</li> </ul>
<i>snr</i>	<ul style="list-style-type: none"> <li>• Reference signal signal-to-noise ratio in dB.</li> <li>• Range -10 to 30</li> </ul>
<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following <ul style="list-style-type: none"> <li>– rxChainIndex</li> <li>– isRadioTuned</li> <li>– rxPower</li> <li>– rsrp</li> </ul> </li> </ul>
<i>sigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RxSigInfo</a> for more information</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.114.2 Field Documentation

8.114.2.1 uint8\_t nas\_SccRxInfo::numInstances

8.114.2.2 int32\_t nas\_SccRxInfo::rsrq

8.114.2.3 nas\_RxSigInfo nas\_SccRxInfo::sigInfo[255]

8.114.2.4 int16\_t nas\_SccRxInfo::snr

8.114.2.5 uint8\_t nas\_SccRxInfo::TlvPresent

## 8.115 nas\_servSystem Struct Reference

### Data Fields

- uint8\_t [regState](#)
- uint8\_t [csAttachState](#)
- uint8\_t [psAttachState](#)
- uint8\_t [selNetwork](#)
- uint8\_t [numRadioInterfaces](#)
- uint8\_t [radioInterface](#) [32]

### 8.115.1 Detailed Description

This structure contains the Serving System parameters

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

#### Parameters

<i>regState</i>	<ul style="list-style-type: none"> <li>• Registration state - Registration state of the mobile</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Not Registered; mobile is not currently searching for a new network to provide service</li> <li>– 1 - Registered with a network</li> <li>– 2 - Not registered, but mobile is currently searching for a new network to provide service</li> <li>– 3 - Registration denied by visible network</li> <li>– 4 - Registration state is unknown</li> </ul> </li> </ul>
<i>csAttachState</i>	<ul style="list-style-type: none"> <li>• CS Attach State - Circuit-switched domain attach state of the mobile</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Unknown or not applicable</li> <li>– 1 - Attached</li> <li>– 2 - Detached</li> </ul> </li> </ul>
<i>psAttachState</i>	<ul style="list-style-type: none"> <li>• PS Attach State - Packet-switched domain attach state of the mobile</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - Unknown or not applicable</li> <li>– 1 - Attached</li> <li>– 2 - Detached</li> </ul> </li> </ul>



<i>selNetwork</i>	<ul style="list-style-type: none"> <li>Selected Network - Type of selected radio access network</li> <li>Values: <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - 3GPP2 network</li> <li>2 - 3GPP network</li> </ul> </li> </ul>
<i>numRadio↔ Interfaces</i>	<ul style="list-style-type: none"> <li>In Use Radio Interfaces Number <ul style="list-style-type: none"> <li>Number of radio interfaces currently in use</li> <li>defaults to zero</li> </ul> </li> </ul>
<i>radioInterface</i>	<ul style="list-style-type: none"> <li>Radio Interface currently in use</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - RADIO_IF_NO_SVC - None(no service)</li> <li>0x01 - RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>0x02 - RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO)</li> <li>0x03 - RADIO_IF_AMPS - AMPS</li> <li>0x04 - RADIO_IF_GSM - GSM</li> <li>0x05 - RADIO_IF_UMTS - UMTS</li> <li>0x08 - RADIO_IF_LTE - LTE</li> </ul> </li> </ul>

## 8.115.2 Field Documentation

8.115.2.1 uint8\_t nas\_servSystem::csAttachState

8.115.2.2 uint8\_t nas\_servSystem::numRadioInterfaces

8.115.2.3 uint8\_t nas\_servSystem::psAttachState

8.115.2.4 uint8\_t nas\_servSystem::radioInterface[32]

8.115.2.5 uint8\_t nas\_servSystem::regState

8.115.2.6 uint8\_t nas\_servSystem::selNetwork

## 8.116 nas\_SignalStrengthTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- int8\_t [signalStrength](#)
- uint32\_t [radioInterface](#)

### 8.116.1 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

## 8.116.2 Field Documentation

8.116.2.1 uint32\_t nas\_SignalStrengthTlv::radioInterface

8.116.2.2 int8\_t nas\_SignalStrengthTlv::signalStrength

8.116.2.3 uint8\_t nas\_SignalStrengthTlv::TlvPresent

## 8.117 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

### 8.117.1 Detailed Description

#### Parameters

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

## 8.117.2 Field Documentation

- 8.117.2.1 `uint8_t` `nas_SLQSSignalStrengthsIndReq::ecioDelta`
- 8.117.2.2 `int16_t` `nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]`
- 8.117.2.3 `uint8_t` `nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen`
- 8.117.2.4 `uint8_t` `nas_SLQSSignalStrengthsIndReq::ioDelta`
- 8.117.2.5 `uint8_t` `nas_SLQSSignalStrengthsIndReq::lteRsrpDelta`
- 8.117.2.6 `uint16_t` `nas_SLQSSignalStrengthsIndReq::lteSnrDelta`
- 8.117.2.7 `uint8_t` `nas_SLQSSignalStrengthsIndReq::rsrqDelta`
- 8.117.2.8 `uint8_t` `nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta`
- 8.117.2.9 `uint8_t` `nas_SLQSSignalStrengthsIndReq::sinrDelta`
- 8.117.2.10 `uint8_t` `nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]`
- 8.117.2.11 `uint8_t` `nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen`

## 8.118 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`

### 8.118.1 Detailed Description

#### Parameters

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

## 8.118.2 Field Documentation

8.118.2.1 `nas_ecioListElement` `nas_SLQSSignalStrengthsInformation::ecioInfo`

8.118.2.2 `nas_errorRateListElement` `nas_SLQSSignalStrengthsInformation::errorRateInfo`

8.118.2.3 `uint32_t` `nas_SLQSSignalStrengthsInformation::io`

8.118.2.4 `nas_lteRsrpInformation` `nas_SLQSSignalStrengthsInformation::lteRsrpInfo`

8.118.2.5 `nas_lteSnrInformation` `nas_SLQSSignalStrengthsInformation::lteSnrInfo`

8.118.2.6 `nas_rsrqInformation` `nas_SLQSSignalStrengthsInformation::rsrqInfo`

8.118.2.7 `nas_rxSignalStrengthListElement` `nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo`

8.118.2.8 `uint8_t` `nas_SLQSSignalStrengthsInformation::sinr`

## 8.119 `nas_SLQSSignalStrengthsTlv` Struct Reference

### Data Fields

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

### 8.119.1 Detailed Description

#### Parameters

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

## 8.119.2 Field Documentation

8.119.2.1 `nas_SLQSSignalStrengthsInformation` `nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo`

8.119.2.2 `uint8_t` `nas_SLQSSignalStrengthsTlv::TlvPresent`

## 8.120 `nas_SrvStatusInfo` Struct Reference

### Data Fields

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

### 8.120.1 Detailed Description

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

#### Parameters

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

### 8.120.2 Field Documentation

8.120.2.1 `uint8_t nas_SrvStatusInfo::isPrefDataPath`

8.120.2.2 `uint8_t nas_SrvStatusInfo::srvStatus`

## 8.121 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`

### 8.121.1 Detailed Description

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

## Parameters

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

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

## 8.121.2 Field Documentation

8.121.2.1 `uint8_t nas_sysInfoCommon::isSysForbidden`

8.121.2.2 `uint8_t nas_sysInfoCommon::isSysForbiddenValid`

8.121.2.3 `uint8_t nas_sysInfoCommon::roamStatus`

8.121.2.4 `uint8_t nas_sysInfoCommon::roamStatusValid`

8.121.2.5 `uint8_t nas_sysInfoCommon::srvCapability`

8.121.2.6 `uint8_t nas_sysInfoCommon::srvCapabilityValid`

8.121.2.7 `uint8_t nas_sysInfoCommon::srvDomain`

8.121.2.8 `uint8_t nas_sysInfoCommon::srvDomainValid`

## 8.122 `nas_TDSCDMAECIOThresh` Struct Reference

### Data Fields

- `uint8_t TDSCDMAECIOThreshListLen`
- `float * pTDSCDMAECIOThreshList`

### 8.122.1 Detailed Description

This structure contains TDSCDMA ECIO threshold related parameters.

#### Parameters

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

### 8.122.2 Field Documentation

8.122.2.1 `float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList`

8.122.2.2 `uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen`

## 8.123 `nas_TDSCDMARSCPThresh` Struct Reference

### Data Fields

- `uint8_t TDSCDMARSCPThreshListLen`
- `int16_t * pTDSCDMARSCPThreshList`

### 8.123.1 Detailed Description

This structure contains TDSCDMA RSCP threshold related parameters.



## Parameters

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

## 8.123.2 Field Documentation

8.123.2.1 int16\_t\* nas\_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.123.2.2 uint8\_t nas\_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.124 nas\_TDSCDMARSSIThresh Struct Reference

## Data Fields

- uint8\_t TDSCDMARSSIThreshListLen
- float \* pTDSCDMARSSIThreshList

## 8.124.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

## Parameters

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

## 8.124.2 Field Documentation

8.124.2.1 float\* nas\_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.124.2.2 uint8\_t nas\_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.125 nas\_TDSCDMASINRThresh Struct Reference

## Data Fields

- uint8\_t [TDSCDMASINRThreshListLen](#)
- float \* [pTDSCDMASINRThreshList](#)

### 8.125.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

#### Parameters

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

### 8.125.2 Field Documentation

8.125.2.1 float\* nas\_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.125.2.2 uint8\_t nas\_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.126 nas\_timeInfo Struct Reference

## Data Fields

- uint16\_t [year](#)
- uint8\_t [month](#)
- uint8\_t [day](#)
- uint8\_t [hour](#)
- uint8\_t [minute](#)
- uint8\_t [second](#)
- uint8\_t [dayOfWeek](#)
- int8\_t [timeZone](#)
- uint8\_t [dayLtSavingAdj](#)
- uint8\_t [radioInterface](#)
- uint8\_t [TlvPresent](#)

### 8.126.1 Detailed Description

This structure contains the parameters for Network Time.

## Parameters

<i>year</i>	<ul style="list-style-type: none"> <li>• Year</li> </ul>
<i>month</i>	<ul style="list-style-type: none"> <li>• Month</li> <li>• 1 is January and 12 is December</li> </ul>
<i>day</i>	<ul style="list-style-type: none"> <li>• Day</li> <li>• Range - 1 to 31</li> </ul>
<i>hour</i>	<ul style="list-style-type: none"> <li>• Hour</li> <li>• Range - 0 to 59</li> </ul>
<i>minute</i>	<ul style="list-style-type: none"> <li>• Minute</li> <li>• Range - 0 to 59</li> </ul>
<i>second</i>	<ul style="list-style-type: none"> <li>• Second</li> <li>• Range - 0 to 59</li> </ul>
<i>dayOfWeek</i>	<ul style="list-style-type: none"> <li>• Day of the week</li> <li>• 0 is Monday and 6 is Sunday</li> </ul>
<i>timeZone</i>	<ul style="list-style-type: none"> <li>• Offset from Universal time</li> <li>• The difference between local time and Universal time, in increments of 15 min</li> <li>• Signed Value</li> </ul>
<i>dayLtSavingAdj</i>	<ul style="list-style-type: none"> <li>• Daylight saving adjustment in hours</li> <li>• Possible values - 0, 1, and 2.</li> <li>• This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVDO</li> </ul>
<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface from which the information comes</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x04 - NAS_RADIO_IF_GSM - GSM</li> <li>– 0x05 - NAS_RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - NAS_RADIO_IF_LTE - LTE</li> <li>– 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.126.2 Field Documentation

8.126.2.1 `uint8_t nas_timeInfo::day`

8.126.2.2 `uint8_t nas_timeInfo::dayLtSavingAdj`

8.126.2.3 `uint8_t nas_timeInfo::dayOfWeek`

8.126.2.4 `uint8_t nas_timeInfo::hour`

8.126.2.5 `uint8_t nas_timeInfo::minute`

8.126.2.6 `uint8_t nas_timeInfo::month`

8.126.2.7 `uint8_t nas_timeInfo::radioInterface`

8.126.2.8 `uint8_t nas_timeInfo::second`

8.126.2.9 `int8_t nas_timeInfo::timeZone`

8.126.2.10 `uint8_t nas_timeInfo::TlvPresent`

8.126.2.11 `uint16_t nas_timeInfo::year`

## 8.127 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]

## 8.127.1 Detailed Description

This structure contains information about the UMTS Network.

## Parameters

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

## 8.127.2 Field Documentation

- 8.127.2.1    `uint16_t nas_UMTSInfo::cellID`
- 8.127.2.2    `int16_t nas_UMTSInfo::ecio`
- 8.127.2.3    `uint8_t nas_UMTSInfo::geranInst`
- 8.127.2.4    `nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]`
- 8.127.2.5    `uint16_t nas_UMTSInfo::lac`
- 8.127.2.6    `uint8_t nas_UMTSInfo::plmn[3]`
- 8.127.2.7    `uint16_t nas_UMTSInfo::psc`
- 8.127.2.8    `int16_t nas_UMTSInfo::rscp`
- 8.127.2.9    `uint16_t nas_UMTSInfo::uarfcn`
- 8.127.2.10    `uint8_t nas_UMTSInfo::umtsInst`
- 8.127.2.11    `nas_UMTSInstInfo nas_UMTSInfo::UMTSInstInfo[255]`

## 8.128    `nas_UMTSInstInfo` Struct Reference

### Data Fields

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

### 8.128.1    Detailed Description

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

#### Parameters

<i>umtsUarfcn</i>	<ul style="list-style-type: none"> <li>• UTRA absolute RF channel number.</li> </ul>
<i>umtsPsc</i>	<ul style="list-style-type: none"> <li>• Primary scrambling code.</li> </ul>
<i>umtsRscp</i>	<ul style="list-style-type: none"> <li>• Received signal code power.</li> </ul>
<i>umtsEcio</i>	<ul style="list-style-type: none"> <li>• ECIO(Signal-to-Interference-ratio).</li> </ul>
	Generated by Doxygen

## 8.128.2 Field Documentation

8.128.2.1 int16\_t nas\_UMTSinstInfo::umtsEcio

8.128.2.2 uint16\_t nas\_UMTSinstInfo::umtsPsc

8.128.2.3 int16\_t nas\_UMTSinstInfo::umtsRscp

8.128.2.4 uint16\_t nas\_UMTSinstInfo::umtsUarfcn

## 8.129 nas\_umtsLTENbrCell Struct Reference

### Data Fields

- uint16\_t [earfcn](#)
- uint16\_t [pci](#)
- uint32\_t [rsrp](#)
- uint32\_t [rsrq](#)
- int16\_t [srxlev](#)
- uint8\_t [cellsTDD](#)

### 8.129.1 Detailed Description

This structure contains information about the UMTS LTE neighbour Cell.

#### Parameters

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

## 8.129.2 Field Documentation

8.129.2.1 `uint8_t nas_umtsLTENbrCell::cellsTDD`

8.129.2.2 `uint16_t nas_umtsLTENbrCell::earfcn`

8.129.2.3 `uint16_t nas_umtsLTENbrCell::pci`

8.129.2.4 `uint32_t nas_umtsLTENbrCell::rsrp`

8.129.2.5 `uint32_t nas_umtsLTENbrCell::rsrq`

8.129.2.6 `int16_t nas_umtsLTENbrCell::srxlev`

## 8.130 `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`

### 8.130.1 Detailed Description

This structure contains the parameters for Universal Time Information.

#### Parameters

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



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

### 8.130.2 Field Documentation

8.130.2.1 uint8\_t nas\_UniversalTime::day

8.130.2.2 uint8\_t nas\_UniversalTime::dayOfWeek

8.130.2.3 uint8\_t nas\_UniversalTime::hour

8.130.2.4 uint8\_t nas\_UniversalTime::minute

8.130.2.5 uint8\_t nas\_UniversalTime::month

8.130.2.6 uint8\_t nas\_UniversalTime::second

8.130.2.7 uint16\_t nas\_UniversalTime::year

## 8.131 nas\_wcdmaCellInfo Struct Reference

### Data Fields

- uint16\_t [psc](#)
- int16\_t [cpich\\_rscp](#)
- int16\_t [cpich\\_ecno](#)
- int16\_t [srxlev](#)

### 8.131.1 Detailed Description

This structure contains information about the WCDMA Cell.

#### Parameters

<i>p<sub>sc</sub></i>	<ul style="list-style-type: none"> <li>Primary scrambling code.</li> <li>Range: 0 to 511.</li> </ul>
<i>cpich<sub>rscp</sub></i>	<ul style="list-style-type: none"> <li>Absolute power level (in 1/10 dBm) of the common pilot channel as received by the UE.</li> <li>Range: -120.0 dBm to -25.0 dBm</li> </ul>
<i>cpich<sub>ecno</sub></i>	<ul style="list-style-type: none"> <li>CPICH Ec/No; ratio (in 1/10 dB) of the received energy per PN chip for the CPICH to the total received power spectral density at the UE antenna connector.</li> <li>Range: -50.0 dB to 0.</li> </ul>
<i>srxlev</i>	<ul style="list-style-type: none"> <li>Cell selection Rx level (Srxlev) value.</li> <li>Range: -128 to 128.</li> <li>This field is only valid when ue_in_idle is TRUE.</li> </ul>

### 8.131.2 Field Documentation

8.131.2.1 int16\_t nas\_wcdmaCellInfo::cpich\_ecno

8.131.2.2 int16\_t nas\_wcdmaCellInfo::cpich\_rscp

8.131.2.3 uint16\_t nas\_wcdmaCellInfo::psc

8.131.2.4 int16\_t nas\_wcdmaCellInfo::srxlev

## 8.132 nas\_WCDMAECIOThresh Struct Reference

#### Data Fields

- uint8\_t [WCDMAECIOThreshListLen](#)
- int16\_t \* [pWCDMAECIOThreshList](#)

### 8.132.1 Detailed Description

This structure contains WCDMA ECIO threshold related parameters.

#### Parameters

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

### 8.132.2 Field Documentation

8.132.2.1 `int16_t* nas_WCDMAECIOThresh::pWCDMAECIOThreshList`

8.132.2.2 `uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen`

## 8.133 nas\_WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

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

### 8.133.1 Detailed Description

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

#### Parameters

<i>wcdmaRRCTest</i> ↔ <i>State</i>	<ul style="list-style-type: none"> <li>WCDMA RRC states.</li> <li>Defined in 3GPP TS 25.331</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - NAS_WCDMA_RRC_STATE_DISCONNECTED * WCDMA RRC State is IDLE</li> <li>0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH * WCDMA RRC state is CELL_PCH</li> <li>0x02 - NAS_WCDMA_RRC_STATE_URA_PCH * WCDMA RRC state is URA_PCH</li> <li>0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH * WCDMA RRC state is CELL_FACH</li> <li>0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH * WCDMA RRC state is CELL_DCH</li> </ul> </li> </ul>
<i>umtsLTENbr</i> ↔ <i>CellLen</i>	<ul style="list-style-type: none"> <li>Number of sets of UMTS LTE Neighbors.</li> </ul>
<i>UMTSLTENbr</i> ↔ <i>Cell</i>	
Generated by Doxygen	<ul style="list-style-type: none"> <li>See <code>nas_umtsLTENbrCell</code> for more information.</li> </ul>

### 8.133.2 Field Documentation

8.133.2.1 `nas_umtsLTENbrCell` `nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]`

8.133.2.2 `uint8_t` `nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen`

8.133.2.3 `uint32_t` `nas_WCDMAInfoLTENeighborCell::wcdmaRRState`

## 8.134 `nas_WCDMARSSIThresh` Struct Reference

### Data Fields

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

### 8.134.1 Detailed Description

This structure contains WCDMA RSSI threshold related parameters.

#### Parameters

<i>WCDMARSSIThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the WCDMA RSSI threshold list parameter to follow</li> </ul>
<i>pWCDMARSSIThreshList</i>	<ul style="list-style-type: none"> <li>• Array of RSSI thresholds (in units of 0.1 dBm)</li> <li>• Maximum of 32 values.</li> <li>• Range for RSSI values: -121 to 0 (in dBm)</li> </ul>

### 8.134.2 Field Documentation

8.134.2.1 `int16_t *` `nas_WCDMARSSIThresh::pWCDMARSSIThreshList`

8.134.2.2 `uint8_t` `nas_WCDMARSSIThresh::WCDMARSSIThreshListLen`

## 8.135 `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](#)

### 8.135.1 Detailed Description

Structure for storing the WCDMA System Information.

#### Parameters

<i>sysInfoWCDMA</i>	<ul style="list-style-type: none"> <li>• See sysInfoCommon for more information.</li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> <li>• Only applies to 3GPP. <ul style="list-style-type: none"> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Not Available</li> </ul> </li> </ul>
<i>regRejectInfoValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the registration reject information is valid. <ul style="list-style-type: none"> <li>– 0x00 - Invalid</li> <li>– 0x01 - Valid</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>

<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>Type of service domain in which the registration is rejected. <ul style="list-style-type: none"> <li>0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>0x01 - Circuit-switched only</li> <li>0x02 - Packet-switched only</li> <li>0x03 - Circuit-switched and packet-switched</li> <li>0x04 - Camped</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>rejCause</i>	<ul style="list-style-type: none"> <li>Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. <ul style="list-style-type: none"> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>networkIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>MCC[PLMN_L↔ ENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[PLMN_L↔ ENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Network Code.</li> <li>MNC digits in ASCII characters</li> <li>An unused byte is set to 0xFF.</li> <li>In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.</li> </ul>
<i>hsCallStatus↔ Valid</i>	<ul style="list-style-type: none"> <li>Indicates whether the high-speed call status is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>hsCallStatus</i>	<ul style="list-style-type: none"> <li>Call status on high speed.</li> <li>Only applicable for WCDMA. <ul style="list-style-type: none"> <li>0x00 - HSDPA and HSUPA are unsupported</li> <li>0x01 - HSDPA is supported</li> <li>0x02 - HSUPA is supported</li> <li>0x03 - HSDPA and HSUPA are supported</li> <li>0x04 - HSDPA+ is supported</li> <li>0x05 - HSDPA+ and HSUPA are supported</li> <li>0x06 - Dual-cell HSDPA+ is supported</li> <li>0x07 - Dual-cell HSDPA+ and HSUPA are supported</li> <li>0xFF - Not Available</li> </ul> </li> </ul>

<i>hsIndValid</i>	<ul style="list-style-type: none"> <li>Indicates whether high-speed service indication is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>hsInd</i>	<ul style="list-style-type: none"> <li>High-speed service indication</li> <li>Only applicable for WCDMA. <ul style="list-style-type: none"> <li>0x00 - HSDPA and HSUPA are unsupported</li> <li>0x01 - HSDPA is supported</li> <li>0x02 - HSUPA is supported</li> <li>0x03 - HSDPA and HSUPA are supported</li> <li>0x04 - HSDPA+ is supported</li> <li>0x05 - HSDPA+ and HSUPA are supported</li> <li>0x06 - Dual-cell HSDPA+ is supported</li> <li>0x07 - Dual-cell HSDPA+ and HSUPA are supported</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>pscValid</i>	<ul style="list-style-type: none"> <li>Indicates whether primary scrambling code is valid. <ul style="list-style-type: none"> <li>0x00 - Invalid</li> <li>0x01 - Valid</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>psc</i>	<ul style="list-style-type: none"> <li>Primary scrambling code. <ul style="list-style-type: none"> <li>0xFFFF - Not Available</li> </ul> </li> </ul>

## 8.135.2 Field Documentation

8.135.2.1 `uint32_t nas_WCDMASysInfo::cellId`

8.135.2.2 `uint8_t nas_WCDMASysInfo::cellIdValid`

8.135.2.3 `uint8_t nas_WCDMASysInfo::hsCallStatus`

8.135.2.4 `uint8_t nas_WCDMASysInfo::hsCallStatusValid`

8.135.2.5 `uint8_t nas_WCDMASysInfo::hsInd`

8.135.2.6 `uint8_t nas_WCDMASysInfo::hsIndValid`

8.135.2.7 `uint16_t nas_WCDMASysInfo::lac`

8.135.2.8 `uint8_t nas_WCDMASysInfo::lacValid`

8.135.2.9    `uint8_t nas_WCDMASysInfo::MCC[3]`

8.135.2.10   `uint8_t nas_WCDMASysInfo::MNC[3]`

8.135.2.11   `uint8_t nas_WCDMASysInfo::networkIdValid`

8.135.2.12   `uint16_t nas_WCDMASysInfo::psc`

8.135.2.13   `uint8_t nas_WCDMASysInfo::pscValid`

8.135.2.14   `uint8_t nas_WCDMASysInfo::regRejectInfoValid`

8.135.2.15   `uint8_t nas_WCDMASysInfo::rejCause`

8.135.2.16   `uint8_t nas_WCDMASysInfo::rejectSrvDomain`

8.135.2.17   `nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA`

## 8.136   NASBandPreferenceTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint64_t` [band\\_pref](#)

### 8.136.1   Field Documentation

8.136.1.1   `uint64_t NASBandPreferenceTlv::band_pref`

8.136.1.2   `uint8_t NASBandPreferenceTlv::TlvPresent`

## 8.137   NASEmergencyModeTlv Struct Reference

### Data Fields

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



### 8.137.1 Field Documentation

8.137.1.1 `uint8_t` `NASEmergencyModeTlv::EmerMode`

8.137.1.2 `uint8_t` `NASEmergencyModeTlv::TlvPresent`

## 8.138 NasGetLTECphyCalInfo Struct Reference

### Data Fields

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

### 8.138.1 Field Documentation

8.138.1.1 `NASPhyCaAggPcellInfo` `NasGetLTECphyCalInfo::PhyCaAggPcellInfo`

8.138.1.2 `NASPhyCaAggScellIDIBw` `NasGetLTECphyCalInfo::PhyCaAggScellIDIBw`

8.138.1.3 `NASPhyCaAggScellIndex` `NasGetLTECphyCalInfo::PhyCaAggScellIndex`

8.138.1.4 `NASPhyCaAggScellIndType` `NasGetLTECphyCalInfo::PhyCaAggScellIndType`

8.138.1.5 `NASPhyCaAggScellInfo` `NasGetLTECphyCalInfo::PhyCaAggScellInfo`

## 8.139 NASGWAcqOrderPrefTlv Struct Reference

### Data Fields

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

### 8.139.1 Field Documentation

8.139.1.1 `uint32_t` `NASGWAcqOrderPrefTlv::GWAcqOrderPref`

8.139.1.2 `uint8_t` `NASGWAcqOrderPrefTlv::TlvPresent`

## 8.140 NASLTEBandPreferenceTlv Struct Reference

### Data Fields

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

### 8.140.1 Field Documentation

8.140.1.1 uint64\_t NASLTEBandPreferenceTlv::LTEBandPref

8.140.1.2 uint8\_t NASLTEBandPreferenceTlv::TlvPresent

## 8.141 NASLteNasReleaseInfoTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [nas\\_release](#)
- uint8\_t [nas\\_major](#)
- uint8\_t [nas\\_minor](#)

### 8.141.1 Field Documentation

8.141.1.1 uint8\_t NASLteNasReleaseInfoTlv::nas\_major

8.141.1.2 uint8\_t NASLteNasReleaseInfoTlv::nas\_minor

8.141.1.3 uint8\_t NASLteNasReleaseInfoTlv::nas\_release

8.141.1.4 uint8\_t NASLteNasReleaseInfoTlv::TlvPresent

## 8.142 NASModePreferenceTlv Struct Reference

### Data Fields

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

### 8.142.1 Field Documentation

8.142.1.1 uint16\_t NASModePreferenceTlv::ModePref

8.142.1.2 uint8\_t NASModePreferenceTlv::TlvPresent

## 8.143 NASNetSelPreferenceTlv Struct Reference

### Data Fields

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

### 8.143.1 Field Documentation

8.143.1.1 `uint8_t NASNetSelPreferenceTlv::NetSelPref`

8.143.1.2 `uint8_t NASNetSelPreferenceTlv::TlvPresent`

## 8.144 NASOTAMessageTlv Struct Reference

### Data Fields

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

### 8.144.1 Field Documentation

8.144.1.1 `uint8_t NASOTAMessageTlv::data_buf[2048]`

8.144.1.2 `uint16_t NASOTAMessageTlv::data_len`

8.144.1.3 `uint32_t NASOTAMessageTlv::message_type`

8.144.1.4 `uint8_t NASOTAMessageTlv::TlvPresent`

## 8.145 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`

### 8.145.1 Detailed Description

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

#### Parameters

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

### 8.145.2 Field Documentation

8.145.2.1 `LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value`

8.145.2.2 `uint32_t NASPhyCaAggPcellInfo::freq`

8.145.2.3 `uint32_t NASPhyCaAggPcellInfo::iLTEbandValue`

8.145.2.4 `uint32_t NASPhyCaAggPcellInfo::pci`

8.145.2.5 `uint8_t NASPhyCaAggPcellInfo::TlvPresent`

## 8.146 NASPhyCaAggScellIDIBw Struct Reference

#### Data Fields

- [LIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB dl\\_bw\\_value](#)
- [uint8\\_t TlvPresent](#)

### 8.146.1 Detailed Description

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

#### Parameters

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

## 8.146.2 Field Documentation

8.146.2.1 LIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB NASPhyCaAggScellIDBw::dl\_bw\_value

8.146.2.2 uint8\_t NASPhyCaAggScellIDBw::TlvPresent

## 8.147 NASPhyCaAggScellIndex Struct Reference

### Data Fields

- uint8\_t [scell\\_idx](#)
- uint8\_t [TlvPresent](#)

### 8.147.1 Detailed Description

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

#### Parameters

<i>scell_idx</i>	<ul style="list-style-type: none"><li>• Physical cell ID of the SCell Range.</li><li>• Range for ID values: 0 to 503.</li></ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Tlv Present.</li></ul>

## 8.147.2 Field Documentation

8.147.2.1 uint8\_t NASPhyCaAggScellIndex::scell\_idx

8.147.2.2 uint8\_t NASPhyCaAggScellIndex::TlvPresent

## 8.148 NASPhyCaAggScellIndType Struct Reference

### Data Fields

- uint32\_t [pci](#)
- uint32\_t [freq](#)
- LIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE [scell\\_state](#)
- uint8\_t [TlvPresent](#)

### 8.148.1 Detailed Description

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

#### Parameters

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

### 8.148.2 Field Documentation

8.148.2.1 `uint32_t NASPhyCaAggScellIndType::freq`

8.148.2.2 `uint32_t NASPhyCaAggScellIndType::pci`

8.148.2.3 `LIBPACK_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state`

8.148.2.4 `uint8_t NASPhyCaAggScellIndType::TlvPresent`

## 8.149 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`

### 8.149.1 Detailed Description

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

#### Parameters

<i>pci</i>	<ul style="list-style-type: none"> <li>Physical cell ID of the SCell Range.</li> <li>Range for ID values: 0 to 503.</li> </ul>
<i>freq</i>	<ul style="list-style-type: none"> <li>Frequency of the absolute cell Range.</li> <li>Range for ID values: 0 to 65535.</li> </ul>
<i>dl_bw_value</i>	<ul style="list-style-type: none"> <li>Downlink Bandwidth Values.</li> <li>See NAS_LTE_CPHY_CA_BW_NRB for more information.</li> </ul>
<i>iLTEbandValue</i>	<ul style="list-style-type: none"> <li>Band value.</li> <li>Range for LTE Band class 120 to 160.</li> </ul>
<i>scell_state</i>	<ul style="list-style-type: none"> <li>Scell state Values.</li> <li>See NAS_LTE_CPHY_SCELL_STATE for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

## 8.149.2 Field Documentation

8.149.2.1 LIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB NASPhyCaAggScellInfo::dl\_bw\_value

8.149.2.2 uint32\_t NASPhyCaAggScellInfo::freq

8.149.2.3 uint32\_t NASPhyCaAggScellInfo::iLTEbandValue

8.149.2.4 uint32\_t NASPhyCaAggScellInfo::pci

8.149.2.5 LIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE NASPhyCaAggScellInfo::scell\_state

8.149.2.6 uint8\_t NASPhyCaAggScellInfo::TlvPresent

## 8.150 NASPRLPreferenceTlv Struct Reference

### Data Fields

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

### 8.150.1 Field Documentation

8.150.1.1 `uint16_t NASPRLPreferenceTlv::PRLPref`

8.150.1.2 `uint8_t NASPRLPreferenceTlv::TlvPresent`

## 8.151 NASQmiCbkNasSwiOTAMessageInd Struct Reference

### Data Fields

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

### 8.151.1 Field Documentation

8.151.1.1 `NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv`

8.151.1.2 `NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv`

8.151.1.3 `NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv`

## 8.152 NASQmiCbkNasSystemSelPrefInd Struct Reference

### Data Fields

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



### 8.152.1 Field Documentation

8.152.1.1 **NASBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::BPTlv

8.152.1.2 **NASEmergencyModeTlv** NASQmiCbkNasSystemSelPrefInd::EMTlv

8.152.1.3 **NASGWAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

8.152.1.4 **NASLTEBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::LBPTlv

8.152.1.5 **NASModePreferenceTlv** NASQmiCbkNasSystemSelPrefInd::MPTlv

8.152.1.6 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.152.1.7 **NASPRLPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.152.1.8 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv

8.152.1.9 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

## 8.153 NASRoamPreferenceTlv Struct Reference

### Data Fields

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

### 8.153.1 Field Documentation

8.153.1.1 uint16\_t NASRoamPreferenceTlv::RoamPref

8.153.1.2 uint8\_t NASRoamPreferenceTlv::TlvPresent

## 8.154 NASServDomainPrefTlv Struct Reference

### Data Fields

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

### 8.154.1 Field Documentation

8.154.1.1 uint32\_t NAServDomainPrefTlv::SrvDomainPref

8.154.1.2 uint8\_t NAServDomainPrefTlv::TlvPresent

## 8.155 NAServingSystemInfo Struct Reference

### Data Fields

- uint8\_t [registrationState](#)
- uint8\_t [csAttachState](#)
- uint8\_t [psAttachState](#)
- uint8\_t [selectedNetwork](#)
- uint8\_t [radioInterfaceNo](#)
- uint8\_t [radioInterfaceList](#) [255]
- uint8\_t [hdrPersonality](#)

### 8.155.1 Detailed Description

This structure will hold the serving system parameters information

#### Parameters

<i>registrationState</i>	- Registration state of the mobile <ul style="list-style-type: none"> <li>• 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service</li> <li>• 1 - QMI_NAS_REGISTERED Registered with a network</li> <li>• 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service</li> <li>• 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network</li> <li>• 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown</li> </ul>
<i>csAttachState</i>	- Circuit Switch domain attach state of the mobile <ul style="list-style-type: none"> <li>• 0 - Unknown or not applicable</li> <li>• 1 - Attached</li> <li>• 2 - Detached</li> </ul>
<i>psAttachState</i>	- Packet domain attach state of the mobile <ul style="list-style-type: none"> <li>• 0 - Unknown or not applicable</li> <li>• 1 - Attached</li> <li>• 2 - Detached</li> </ul>
<i>selectedNetwork</i>	- Type of selected radio access network <ul style="list-style-type: none"> <li>• 0x00 - Unknown</li> <li>• 0x01 - 3GPP2 network</li> <li>• 0x02 - 3GPP network</li> </ul>
<i>radioInterfaceNo</i>	- Number of radio interfaces currently in use; this indicates how many radio_if identifiers follow this field

<i>radioInterfaceList</i>	- Radio interface currently in use (each is 1 byte) <ul style="list-style-type: none"> <li>• 0x00 - None (no service)</li> <li>• 0x01 - cdma2000 1X</li> <li>• 0x02 - cdma2000 HRPD (1xEV-DO)</li> <li>• 0x03 - AMPS</li> <li>• 0x04 - GSM</li> <li>• 0x05 - UMTS</li> <li>• 0x08 - LTE</li> </ul>
<i>hdrPersonality</i>	- HDR personality information (valid only for EVDO) <ul style="list-style-type: none"> <li>• 0x00 - Unknown</li> <li>• 0x01 - HRPD</li> <li>• 0x02 - eHRPD</li> </ul>

Note: None

## 8.155.2 Field Documentation

8.155.2.1 `uint8_t NAServingSystemInfo::csAttachState`

8.155.2.2 `uint8_t NAServingSystemInfo::hdrPersonality`

8.155.2.3 `uint8_t NAServingSystemInfo::psAttachState`

8.155.2.4 `uint8_t NAServingSystemInfo::radioInterfaceList[255]`

8.155.2.5 `uint8_t NAServingSystemInfo::radioInterfaceNo`

8.155.2.6 `uint8_t NAServingSystemInfo::registrationState`

8.155.2.7 `uint8_t NAServingSystemInfo::selectedNetwork`

## 8.156 NASTimeInfoTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t time`

### 8.156.1 Field Documentation

8.156.1.1 `uint64_t` `NASTimeInfoTlv::time`

8.156.1.2 `uint8_t` `NASTimeInfoTlv::TlvPresent`

## 8.157 `newMTMessageTlv` Struct Reference

### Data Fields

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

### 8.157.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>MTMessageInfo</i>	<ul style="list-style-type: none"> <li>• MT Message</li> <li>• See <a href="#">sMSMTMessageInfo</a> for more information</li> </ul>

### 8.157.2 Field Documentation

8.157.2.1 `sMSMTMessageInfo` `newMTMessageTlv::MTMessageInfo`

8.157.2.2 `uint8_t` `newMTMessageTlv::TlvPresent`

## 8.158 `pack_dms_ActivateAutomatic_t` Struct Reference

### Data Fields

- `uint8_t` [actCode](#) [81]

### 8.158.1 Detailed Description

This structure contains UIM activate automation information

#### Parameters

<i>spc[IN]</i>	<ul style="list-style-type: none"> <li>• Service programming code in ASCII format (digits 0 to 9 only)</li> </ul>
----------------	---

### 8.158.2 Field Documentation

8.158.2.1 uint8\_t pack\_dms\_ActivateAutomatic\_t::actCode[81]

## 8.159 pack\_dms\_GetCustFeaturesV2\_t Struct Reference

### Data Fields

- uint8\_t [cust\\_id](#) [64+1]
- uint8\_t [list\\_type](#)
- uint16\_t [Tlvresult](#)

### 8.159.1 Detailed Description

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

#### Parameters

<i>cust_id</i>	<ul style="list-style-type: none"> <li>• Customization ID (Maximum 64 bytes)</li> </ul>
<i>list_type</i>	<ul style="list-style-type: none"> <li>• list type requested</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack Result</li> </ul>

### 8.159.2 Field Documentation

8.159.2.1 uint8\_t pack\_dms\_GetCustFeaturesV2\_t::cust\_id[64+1]

8.159.2.2 uint8\_t pack\_dms\_GetCustFeaturesV2\_t::list\_type

8.159.2.3 uint16\_t pack\_dms\_GetCustFeaturesV2\_t::Tlvresult

## 8.160 pack\_dms\_ResetToFactoryDefaults\_t Struct Reference

### Data Fields

- uint8\_t [spc](#) [6]

### 8.160.1 Detailed Description

This structure contains UIM reset to factory default information

#### Parameters

<i>spc</i> [IN]	<ul style="list-style-type: none"> <li>• Service programming code in ASCII format (digits 0 to 9 only)</li> </ul>
-----------------	---

### 8.160.2 Field Documentation

8.160.2.1 uint8\_t pack\_dms\_ResetToFactoryDefaults\_t::spc[6]

## 8.161 pack\_dms\_SetActivationStatusCallback\_t Struct Reference

### Data Fields

- uint8\_t [activationState](#)

### 8.161.1 Detailed Description

This structure is used to store Service Activation Status parameter .

#### Parameters

<i>activationState</i>	<ul style="list-style-type: none"> <li>• Service activation state.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report activation state changes</li> </ul> </li> </ul>
------------------------	---

### 8.161.2 Field Documentation

8.161.2.1 uint8\_t pack\_dms\_SetActivationStatusCallback\_t::activationState

## 8.162 pack\_dms\_SetCrashAction\_t Struct Reference

### Data Fields

- uint8\_t [crashAction](#)

### 8.162.1 Detailed Description

Modem action in case of a crash

## Parameters

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

## 8.162.2 Field Documentation

8.162.2.1 uint8\_t pack\_dms\_SetCrashAction\_t::crashAction

## 8.163 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](#)

## 8.163.1 Field Documentation

8.163.1.1 uint8\_t pack\_dms\_SetCustFeature\_t::DHCPRelayEnabled

8.163.1.2 uint8\_t pack\_dms\_SetCustFeature\_t::DisableIMSI

8.163.1.3 uint32\_t pack\_dms\_SetCustFeature\_t::GpsEnable

8.163.1.4 uint8\_t pack\_dms\_SetCustFeature\_t::GPSLPM

8.163.1.5 uint8\_t pack\_dms\_SetCustFeature\_t::GPSSel

8.163.1.6 uint16\_t pack\_dms\_SetCustFeature\_t::IPFamSupport

8.163.1.7 uint8\_t pack\_dms\_SetCustFeature\_t::IsVoiceEnabled

8.163.1.8 uint8\_t pack\_dms\_SetCustFeature\_t::RMAutoConnect

8.163.1.9 uint8\_t pack\_dms\_SetCustFeature\_t::SMSSupport

## 8.164 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](#)

### 8.164.1 Detailed Description

This structure contains customization settings set to modem pack

#### Parameters

<i>cust_id</i>	<ul style="list-style-type: none"> <li>Customization ID (Maximum 64 bytes)</li> </ul>
<i>value_length</i>	<ul style="list-style-type: none"> <li>length of cust_value field</li> </ul>
<i>cust_value</i>	<ul style="list-style-type: none"> <li>Customization Setting Value (Maximum 8 bytes)</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Pack Result</li> </ul>

### 8.164.2 Field Documentation

8.164.2.1 `uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]`

8.164.2.2 `uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]`

8.164.2.3 `uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult`

8.164.2.4 `uint16_t pack_dms_SetCustFeaturesV2_t::value_length`

## 8.165 pack\_dms\_SetEventReport\_t Struct Reference

### Data Fields

- `uint8_t mode`

### 8.165.1 Field Documentation

8.165.1.1 `uint8_t pack_dms_SetEventReport_t::mode`

## 8.166 pack\_dms\_SetPower\_t Struct Reference

### Data Fields

- `uint32_t mode`
- `uint16_t Tlvresult`



### 8.166.1 Field Documentation

8.166.1.1 uint32\_t pack\_dms\_SetPower\_t::mode

8.166.1.2 uint16\_t pack\_dms\_SetPower\_t::Tlvresult

## 8.167 pack\_dms\_SetUSBComp\_t Struct Reference

### Data Fields

- uint8\_t [USBComp](#)
- uint16\_t [Tlvresult](#)

### 8.167.1 Field Documentation

8.167.1.1 uint16\_t pack\_dms\_SetUSBComp\_t::Tlvresult

8.167.1.2 uint8\_t pack\_dms\_SetUSBComp\_t::USBComp

## 8.168 pack\_dms\_SLQSDmsSwiIndicationRegister\_t Struct Reference

### Data Fields

- uint8\_t [resetInfoInd](#)

### 8.168.1 Detailed Description

#### Parameters

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

### 8.168.2 Field Documentation

8.168.2.1 uint8\_t pack\_dms\_SLQSDmsSwiIndicationRegister\_t::resetInfoInd

## 8.169 pack\_dms\_SLQSSwiGetCrashInfo\_t Struct Reference

### Data Fields

- uint8\_t [clear](#)

### 8.169.1 Detailed Description

This structure contains [crashInfoParams](#)

#### Parameters

<i>clear</i>	<ul style="list-style-type: none"> <li>• 0 - do not clear crash data after response</li> <li>• 1 - clear crash data after response</li> </ul>
--------------	---

### 8.169.2 Field Documentation

8.169.2.1 `uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear`

## 8.170 `pack_dms_SLQSSwiSetDyingGaspCfg_t` Struct Reference

#### Data Fields

- `uint8_t * pDestSMSNum`
- `uint8_t * pDestSMSContent`

### 8.170.1 Detailed Description

#### Parameters

<i>pDestSMS↵ Num[IN]</i>	<ul style="list-style-type: none"> <li>• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>• Optional parameter.</li> </ul>
<i>pDestSMS↵ Content[IN]</i>	<ul style="list-style-type: none"> <li>• SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>• Optional parameter.</li> </ul>

### 8.170.2 Field Documentation

8.170.2.1 `uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent`

8.170.2.2 `uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum`

## 8.171 `pack_dms_SLQSSwiSetHostDevInfo_t` Struct Reference

#### Data Fields

- `char manString [255]`
- `char modelString [255]`
- `char swVerString [255]`
- `char plasmaIDString [255]`

### 8.171.1 Detailed Description

This structure contains SWI set host device info unpack information

#### Parameters

<i>manString</i> [IN]	<ul style="list-style-type: none"> <li>optional parameter, host device manufacture</li> </ul>
<i>modelString</i> [IN]	<ul style="list-style-type: none"> <li>optional parameter, host device model</li> </ul>
<i>swVerString</i> [IN]	<ul style="list-style-type: none"> <li>optional parameter, host device software version</li> </ul>
<i>plasmaID</i> ↔ <i>String</i> [IN]	<ul style="list-style-type: none"> <li>optional parameter, host device plasma ID</li> </ul>

### 8.171.2 Field Documentation

8.171.2.1 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::manString[255]

8.171.2.2 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::modelString[255]

8.171.2.3 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::plasmaIDString[255]

8.171.2.4 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::swVerString[255]

## 8.172 pack\_dms\_SLQSSwiSetOSInfo\_t Struct Reference

### Data Fields

- char [nameString](#) [255]
- char [versionString](#) [255]

### 8.172.1 Detailed Description

This structure contains SWI set host OS info pack information

#### Parameters

<i>nameString</i> [IN]	<ul style="list-style-type: none"> <li>optional parameter, host device manufacture</li> </ul>
<i>VersionString</i> [IN]	<ul style="list-style-type: none"> <li>optional parameter, host device model</li> </ul>

### 8.172.2 Field Documentation

8.172.2.1 `char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]`

8.172.2.2 `char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]`

## 8.173 `pack_dms_UIMChangePIN_t` Struct Reference

### Data Fields

- `uint8_t id`
- `uint8_t oldValue` [255]
- `uint8_t newValue` [255]

### 8.173.1 Detailed Description

This structure contains UIM Unblock PIN Information

#### Parameters

<i>id</i> [IN]	<ul style="list-style-type: none"> <li>• PIN ID             <ul style="list-style-type: none"> <li>– 1 ( PIN1 / CHV1 )</li> <li>– 2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>oldValue</i> [IN]	<ul style="list-style-type: none"> <li>• Old PIN value of PIN to change</li> </ul>
<i>newValue</i> [IN]	<ul style="list-style-type: none"> <li>• New PIN value of PIN to change</li> </ul>

### 8.173.2 Field Documentation

8.173.2.1 `uint8_t pack_dms_UIMChangePIN_t::id`

8.173.2.2 `uint8_t pack_dms_UIMChangePIN_t::newValue`[255]

8.173.2.3 `uint8_t pack_dms_UIMChangePIN_t::oldValue`[255]

## 8.174 `pack_dms_UIMGetControlKeyStatus_t` Struct Reference

### Data Fields

- `uint8_t facility`

### 8.174.1 Detailed Description

This structure contains UIM get control key status information

#### Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> <li>• MT or network facility             <ul style="list-style-type: none"> <li>– 0 - Network Personalization (PN)</li> <li>– 1 - Network Subset Personalization (PU)</li> <li>– 2 - Service Provider Personalization (PP)</li> <li>– 3 - Corporate Personalization (PC)</li> <li>– 4 - UIM Personalization (PF)</li> </ul> </li> </ul>
----------------------	--

### 8.174.2 Field Documentation

8.174.2.1 uint8\_t pack\_dms\_UIMGetControlKeyStatus\_t::facility

## 8.175 pack\_dms\_UIMGetICCID\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.175.1 Detailed Description

This structure contains UIM Get ICCID pack

#### Parameters

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

### 8.175.2 Field Documentation

8.175.2.1 uint16\_t pack\_dms\_UIMGetICCID\_t::Tlvresult

## 8.176 pack\_dms\_UIMSetControlKeyProtection\_t Struct Reference

#### Data Fields

- uint8\_t [facility](#)
- uint8\_t [facilityState](#)
- uint8\_t [facilityCk](#) [8]

### 8.176.1 Detailed Description

This structure contains UIM Set control key protection information

#### Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> <li>• MT or network facility <ul style="list-style-type: none"> <li>– 0 - Network Personalization (PN)</li> <li>– 1 - Network Subset Personalization (PU)</li> <li>– 2 - Service Provider Personalization (PP)</li> <li>– 3 - Corporate Personalization (PC)</li> <li>– 4 - UIM Personalization (PF)</li> </ul> </li> </ul>
<i>facilityState</i> [IN]	<ul style="list-style-type: none"> <li>• UIM facility state</li> </ul>
<i>facilityCk</i> [IN]	<ul style="list-style-type: none"> <li>• Facility depersonalization control key, string in ASCII text (maximum 8 bytes)</li> </ul>

### 8.176.2 Field Documentation

8.176.2.1 `uint8_t pack_dms_UIMSetControlKeyProtection_t::facility`

8.176.2.2 `uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]`

8.176.2.3 `uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState`

## 8.177 `pack_dms_UIMSetPINProtection_t` Struct Reference

### Data Fields

- `uint8_t id`
- `uint8_t bEnable`
- `uint8_t value` [255]

### 8.177.1 Detailed Description

Set activation 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 This structure contains PIN Protection Information
	<i>id[IN]</i>	<ul style="list-style-type: none"> <li>PIN ID <ul style="list-style-type: none"> <li>1 ( PIN1 / CHV1 )</li> <li>2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
	<i>bEnable[IN]</i>	<ul style="list-style-type: none"> <li>Enable/disable PIN protection, 0 = Disable</li> </ul>
	<i>pValue[IN]</i>	<ul style="list-style-type: none"> <li>PIN value of the PIN to be enabled/disabled</li> </ul>

## 8.177.2 Field Documentation

8.177.2.1 uint8\_t pack\_dms\_UIMSetPINProtection\_t::bEnable

8.177.2.2 uint8\_t pack\_dms\_UIMSetPINProtection\_t::id

8.177.2.3 uint8\_t pack\_dms\_UIMSetPINProtection\_t::value[255]

## 8.178 pack\_dms\_UIMUnlockControlKey\_t Struct Reference

## Data Fields

- uint8\_t [facility](#)
- uint8\_t [facilityCk](#) [8]

## 8.178.1 Detailed Description

This structure contains UIM unblock Control Key information

## Parameters

<i>facility[IN]</i>	<ul style="list-style-type: none"> <li>MT or network facility <ul style="list-style-type: none"> <li>0 - Network Personalization (PN)</li> <li>1 - Network Subset Personalization (PU)</li> <li>2 - Service Provider Personalization (PP)</li> <li>3 - Corporate Personalization (PC)</li> <li>4 - UIM Personalization (PF)</li> </ul> </li> </ul>
<i>facilityCk[IN]</i>	<ul style="list-style-type: none"> <li>Facility depersonalization control key, string in ASCII text (maximum 8 bytes)</li> </ul>

## 8.178.2 Field Documentation

8.178.2.1 `uint8_t pack_dms_UIMUnblockControlKey_t::facility`

8.178.2.2 `uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]`

## 8.179 `pack_dms_UIMUnblockPIN_t` Struct Reference

### Data Fields

- `uint8_t id`
- `uint8_t pukValue` [255]
- `uint8_t newPin` [255]

### 8.179.1 Detailed Description

This structure contains UIM Unblock PIN Information

#### Parameters

<i>id</i> [IN]	<ul style="list-style-type: none"> <li>• PIN ID             <ul style="list-style-type: none"> <li>– 1 ( PIN1 / CHV1 )</li> <li>– 2 ( PIN2 / CHV2 )</li> </ul> </li> </ul>
<i>pukValue</i> [IN]	<ul style="list-style-type: none"> <li>• PUK value of PIN to be unblocked</li> </ul>
<i>newPin</i> [IN]	<ul style="list-style-type: none"> <li>• New PIN value for the PIN to be unblocked</li> </ul>

## 8.179.2 Field Documentation

8.179.2.1 `uint8_t pack_dms_UIMUnblockPIN_t::id`

8.179.2.2 `uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]`

8.179.2.3 `uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]`

## 8.180 `pack_dms_UIMVerifyPIN_t` Struct Reference

### Data Fields

- `uint8_t id`
- `uint8_t value` [255]



### 8.180.1 Detailed Description

This structure contains PIN Value Information

#### Parameters

<i>id</i> [IN]	<ul style="list-style-type: none"><li>• PIN ID<ul style="list-style-type: none"><li>– 1 ( PIN1 / CHV1 )</li><li>– 2 ( PIN2 / CHV2 )</li></ul></li></ul>
<i>pValue</i> [IN]	<ul style="list-style-type: none"><li>• PIN value of the PIN to be enabled/disabled</li></ul>

### 8.180.2 Field Documentation

8.180.2.1 uint8\_t pack\_dms\_UIMVerifyPIN\_t::id

8.180.2.2 uint8\_t pack\_dms\_UIMVerifyPIN\_t::value[255]

## 8.181 pack\_fms\_GetImagesPreference\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.181.1 Detailed Description

This structure contains the Get Image Preference information pack

#### Parameters

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

### 8.181.2 Field Documentation

8.181.2.1 uint16\_t pack\_fms\_GetImagesPreference\_t::Tlvresult

## 8.182 pack\_fms\_GetStoredImages\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.182.1 Detailed Description

This structure contains the Get Stored Images pack

#### Parameters

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

### 8.182.2 Field Documentation

8.182.2.1 uint16\_t pack\_fms\_GetStoredImages\_t::Tlvresult

## 8.183 pack\_fms\_SetImagesPreference\_t Struct Reference

#### Data Fields

- uint32\_t [imageListSize](#)
- [FMSPrefImageList](#) \* [pImageList](#)
- uint32\_t [bForceDownload](#)
- uint8\_t [modemindex](#)
- uint16\_t [Tlvresult](#)

### 8.183.1 Detailed Description

This structure contains the Set Images Preference pack

#### Parameters

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

### 8.183.2 Field Documentation

8.183.2.1 `uint32_t pack_fms_SetImagesPreference_t::bForceDownload`

8.183.2.2 `uint32_t pack_fms_SetImagesPreference_t::imageListSize`

8.183.2.3 `uint8_t pack_fms_SetImagesPreference_t::modemindex`

8.183.2.4 `FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList`

8.183.2.5 `uint16_t pack_fms_SetImagesPreference_t::Tlvresult`

## 8.184 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`

### 8.184.1 Detailed Description

This structure contains LOC delete assist data pack

#### Parameters

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

## 8.184.2 Field Documentation

8.184.2.1 `loc_BdsSVInfo*` `pack_loc_Delete_Assist_Data_t::pBdsSVInfo`

8.184.2.2 `loc_CellDb*` `pack_loc_Delete_Assist_Data_t::pCellDb`

8.184.2.3 `loc_ClkInfo*` `pack_loc_Delete_Assist_Data_t::pClkInfo`

8.184.2.4 `loc_GnssData*` `pack_loc_Delete_Assist_Data_t::pGnssData`

8.184.2.5 `loc_SVInfo*` `pack_loc_Delete_Assist_Data_t::pSVInfo`

8.184.2.6 `uint16_t` `pack_loc_Delete_Assist_Data_t::Tlvresult`

## 8.185 `pack_loc_EventRegister_t` Struct Reference

### Data Fields

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

### 8.185.1 Detailed Description

This structure contains the Parameter for RegisterEvents

## Parameters

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

**Note**

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

**Parameters**

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

**8.185.2 Field Documentation**

8.185.2.1 `uint64_t pack_loc_EventRegister_t::eventRegister`

8.185.2.2 `uint16_t pack_loc_EventRegister_t::Tlvresult`

**8.186 pack\_loc\_SetExtPowerState\_t Struct Reference****Data Fields**

- `uint32_t extPowerState`
- `uint16_t Tlvresult`

**8.186.1 Detailed Description**

This structure contains the Parameter External Power Source State pack.

**Parameters**

<i>extPowerState</i>	<ul style="list-style-type: none"> <li>• Specifies the Power state; injected by the control point.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Device is not connected to an external power source</li> <li>– 1 - Device is connected to an external power source</li> <li>– 2 - Unknown external power state</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack result.</li> </ul>

**8.186.2 Field Documentation**

8.186.2.1 `uint32_t pack_loc_SetExtPowerState_t::extPowerState`

8.186.2.2 `uint16_t pack_loc_SetExtPowerState_t::Tlvresult`

## 8.187 pack\_loc\_SetOperationMode\_t Struct Reference

### Data Fields

- uint32\_t [mode](#)
- uint16\_t [Tlvresult](#)

### 8.187.1 Detailed Description

This structure contains Set Operation Mode pack

#### Parameters

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

### 8.187.2 Field Documentation

8.187.2.1 uint32\_t pack\_loc\_SetOperationMode\_t::mode

8.187.2.2 uint16\_t pack\_loc\_SetOperationMode\_t::Tlvresult

## 8.188 pack\_loc\_SLQSLOCGetBestAvailPos\_t Struct Reference

### Data Fields

- uint32\_t [xid](#)
- uint16\_t [Tlvresult](#)

### 8.188.1 Detailed Description

This structure contains Set Operation Mode pack

#### Parameters

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

## 8.188.2 Field Documentation

8.188.2.1 `uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult`

8.188.2.2 `uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid`

## 8.189 `pack_loc_SLQSLOCInjectPosition_t` Struct Reference

### Data Fields

- double `latitude`
- int `has_latitude`
- double `longitude`
- int `has_longitude`
- float `horUncCircular`
- int `has_horUncCircular`
- `uint8_t` `horConfidence`
- int `has_horConfidence`
- `uint32_t` `horReliability`
- int `has_horReliability`
- float `altitudeWrtEllipsoid`
- int `has_altitudeWrtEllipsoid`
- float `altitudeWrtMeanSeaLevel`
- int `has_altitudeWrtMeanSeaLevel`
- float `vertUnc`
- int `has_vertUnc`
- `uint8_t` `vertConfidence`
- int `has_vertConfidence`
- `uint32_t` `vertReliability`
- int `has_vertReliability`
- `altSrcInfo_t` `altitudeSrcInfo`
- int `has_altitudeSrcInfo`
- `uint64_t` `timestampUtc`
- int `has_timestampUtc`
- `uint32_t` `timestampAge`
- int `has_timestampAge`
- `uint32_t` `positionSrc`
- int `has_positionSrc`
- float `rawHorUncCircular`
- int `has_rawHorUncCircular`
- `uint8_t` `rawHorConfidence`
- int `has_rawHorConfidence`



## 8.189.1 Detailed Description

This structure contains LOC Inject Position parameters Please check has\_<Param\_Name> field for presence of optional parameters

## Parameters

<i>latitude</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -90.0 to 90.0</li> <li>• Positive values indicate northern latitude</li> <li>• Negative values indicate southern latitude</li> <li>• Note - This field must be specified together with pLongitude and pHorUncCircular.</li> </ul>
<i>longitude</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -180.0 to 180.0</li> <li>• Positive values indicate eastern latitude</li> <li>• Negative values indicate western latitude</li> <li>• Note - This field must be specified together with pLatitude and pHorUncCircular.</li> </ul>
<i>horUncCircular</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal position uncertainty.</li> <li>• Units - Meters</li> <li>• Note - This field must be specified together with pLatitude and pLongitude.</li> </ul>
<i>horConfidence</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal confidence.</li> <li>• Units - Percent</li> <li>• Values <ul style="list-style-type: none"> <li>– Valid Values - 1 to 99</li> <li>– Invalid Values - 0, 101 to 255</li> <li>– If 100 is received, reinterpret to 99</li> </ul> </li> <li>• Note - This field must be specified together with horizontal uncertainty. If not specified when pHorUncCircular is set, the default value is 50.</li> </ul>
<i>horReliability</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> </ul>

- Values
  - 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

## Parameters

<i>altitudeWrt<sub>↔</sub> Ellipsoid</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Altitude With Respect to Ellipsoid.</li> <li>• Units - Meters</li> <li>• Values               <ul style="list-style-type: none"> <li>– Positive - height</li> <li>– Negative = depth</li> </ul> </li> </ul>
<i>altitudeWrt<sub>↔</sub> MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Altitude With Respect to Sea Level.</li> <li>• Units - Meters</li> </ul>
<i>vertUnc</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Vertical uncertainty.</li> <li>• Units - Meters</li> <li>• Note - This is mandatory if either pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified.</li> </ul>
<i>vertConfidence</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Vertical confidence.</li> <li>• Units - Percentage</li> <li>• Values               <ul style="list-style-type: none"> <li>– Valid Values - 0 to 99</li> <li>– Invalid Values - 0, 100-256</li> <li>– If 100 is received, reinterpret to 99</li> </ul> </li> <li>• Note - This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50.</li> </ul>
<i>vertReliability</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> </ul>

- Values
  - 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

## Parameters

<i>altitudeSrcInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> </ul>
------------------------	--

- Pointer to struct altitudeSrcInfo. See [altitudeSrcInfo](#) for more information

## Parameters

<i>timestampUtc</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> </ul>
<i>timestampAge</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Position age, which is an estimate of how long ago this fix was made.</li> <li>• Units - Milliseconds</li> </ul>
<i>positionSrc</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Source from which this position was obtained</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Position source is GNSS</li> <li>– 1 - Position source is Cell ID</li> <li>– 2 - Position source is Enhanced Cell ID</li> <li>– 3 - Position source is Wi-Fi</li> <li>– 4 - Position source is Terrestrial</li> <li>– 5 - Position source is GNSS Terrestrial Hybrid</li> <li>– 6 - Other sources</li> </ul> </li> <li>• Note - If altitude is specified and the altitude source is not specified, the engine assumes that the altitude was obtained using the specified position source. <ul style="list-style-type: none"> <li>– If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source.</li> </ul> </li> </ul>
<i>rawHorUnc</i> ↔ <i>Circular</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal position uncertainty (circular) without any optimization.</li> <li>• Units - Meters</li> </ul>
<i>rawHor</i> ↔ <i>Confidence</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal confidence associated with raw horizontal uncertainty</li> <li>• Units: Percent</li> <li>• Values <ul style="list-style-type: none"> <li>– Valid values - 1 to 99</li> <li>– Invalid values - 0, 101 to 255</li> <li>– If 100 is received, reinterpret to 99</li> </ul> </li> <li>• Note - This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50.</li> </ul>

## 8.189.2 Field Documentation

## 8.189.2.1 altSrcInfo\_t pack\_loc\_SLQSLOCInjectPosition\_t::altitudeSrcInfo

- 8.189.2.2 float pack\_loc\_SLQSLOCInjectPosition\_t::altitudeWrtEllipsoid
- 8.189.2.3 float pack\_loc\_SLQSLOCInjectPosition\_t::altitudeWrtMeanSeaLevel
- 8.189.2.4 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_altitudeSrcInfo
- 8.189.2.5 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_altitudeWrtEllipsoid
- 8.189.2.6 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_altitudeWrtMeanSeaLevel
- 8.189.2.7 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_horConfidence
- 8.189.2.8 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_horReliability
- 8.189.2.9 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_horUncCircular
- 8.189.2.10 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_latitude
- 8.189.2.11 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_longitude
- 8.189.2.12 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_positionSrc
- 8.189.2.13 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_rawHorConfidence
- 8.189.2.14 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_rawHorUncCircular
- 8.189.2.15 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_timestampAge
- 8.189.2.16 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_timestampUtc
- 8.189.2.17 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_vertConfidence
- 8.189.2.18 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_vertReliability
- 8.189.2.19 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_vertUnc
- 8.189.2.20 uint8\_t pack\_loc\_SLQSLOCInjectPosition\_t::horConfidence
- 8.189.2.21 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::horReliability
- 8.189.2.22 float pack\_loc\_SLQSLOCInjectPosition\_t::horUncCircular
- 8.189.2.23 double pack\_loc\_SLQSLOCInjectPosition\_t::latitude
- 8.189.2.24 double pack\_loc\_SLQSLOCInjectPosition\_t::longitude

- 8.189.2.25 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::positionSrc
- 8.189.2.26 uint8\_t pack\_loc\_SLQSLOCInjectPosition\_t::rawHorConfidence
- 8.189.2.27 float pack\_loc\_SLQSLOCInjectPosition\_t::rawHorUncCircular
- 8.189.2.28 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::timestampAge
- 8.189.2.29 uint64\_t pack\_loc\_SLQSLOCInjectPosition\_t::timestampUtc
- 8.189.2.30 uint8\_t pack\_loc\_SLQSLOCInjectPosition\_t::vertConfidence
- 8.189.2.31 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::vertReliability
- 8.189.2.32 float pack\_loc\_SLQSLOCInjectPosition\_t::vertUnc

## 8.190 pack\_loc\_SLQSLOCInjectSensorData\_t Struct Reference

### Data Fields

- int [has\\_opaqueId](#)
- uint32\_t [opaqueId](#)
- int [has\\_acceleroData](#)
- [sensorData\\_t](#) [acceleroData](#)
- int [has\\_gyroData](#)
- [sensorData\\_t](#) [gyroData](#)
- int [has\\_acceleroTimeSrc](#)
- uint32\_t [acceleroTimeSrc](#)
- int [has\\_gyroTimeSrc](#)
- uint32\_t [gyroTimeSrc](#)
- int [has\\_accelTemp](#)
- [tempData\\_t](#) [accelTemp](#)
- int [has\\_gyroTemp](#)
- [tempData\\_t](#) [gyroTemp](#)

### 8.190.1 Detailed Description

This structure contains parameters to inject sensor data into the GNSS location engine Please check [has\\_<↔ Param\\_Name>](#) field for presence of optional parameters

## Parameters

<i>opaqueId</i>	<ul style="list-style-type: none"> <li>• Opaque Identifier (Optional parameter)</li> <li>• An opaque identifier that is sent in by the client that will be echoed in the indication so the client can relate the indication to the request.</li> </ul>
<i>acceleroData</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct sensorData. See sensorData for more information</li> </ul>
<i>gyroData</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct sensorData. See sensorData for more information</li> </ul>
<i>acceleroTimeSrc</i>	<ul style="list-style-type: none"> <li>• 3-Axis Accelerometer Data Time Source (Optional parameter)</li> <li>• The location service uses this field to identify the time reference used in the accelerometer data time stamps.</li> <li>• If not specified, the location service assumes that the time source for the accelerometer data is unknown.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Sensor time source is unspecified</li> <li>– 1 - Time source is common between the sensors and the location engine</li> </ul> </li> </ul>
<i>gyroTimeSrc</i>	<ul style="list-style-type: none"> <li>• 3-Axis Gyroscope Data Time Source (Optional)</li> <li>• The location service uses this field to identify the time reference used in the gyroscope data time stamps.</li> <li>• If not specified, the location service assumes that the time source for the gyroscope data is unknown.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Sensor time source is unspecified</li> <li>– 1 - Time source is common between the sensors and the location engine</li> </ul> </li> </ul>
<i>accelTemp</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct tempratureData. See tempratureData for more information</li> </ul>
<i>gyroTemp</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct tempratureData. See tempratureData for more information</li> </ul>

## 8.190.2 Field Documentation

8.190.2.1 `sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData`8.190.2.2 `uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc`8.190.2.3 `tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp`

8.190.2.4 `sensorData_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroData`

8.190.2.5 `tempData_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroTemp`

8.190.2.6 `uint32_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc`

8.190.2.7 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc`

8.190.2.8 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp`

8.190.2.9 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_accleroData`

8.190.2.10 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroData`

8.190.2.11 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp`

8.190.2.12 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc`

8.190.2.13 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_opaqueld`

8.190.2.14 `uint32_t` `pack_loc_SLQSLOCInjectSensorData_t::opaqueld`

## 8.191 pack\_loc\_SLQSLOCInjectUTCTime\_t Struct Reference

### Data Fields

- `uint64_t` [timeMsec](#)
- `uint32_t` [timeUncMsec](#)

### 8.191.1 Detailed Description

#### Parameters

<i>timeMsec</i> [IN]	<ul style="list-style-type: none"> <li>• The UTC time since Jan. 1, 1970</li> </ul>
<i>timeUncMsec</i> [↔ IN]	<ul style="list-style-type: none"> <li>• The time Uncertainty</li> </ul>

### 8.191.2 Field Documentation

8.191.2.1 `uint64_t` `pack_loc_SLQSLOCInjectUTCTime_t::timeMsec`

8.191.2.2 `uint32_t` `pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec`

## 8.192 pack\_loc\_SLQSLOCSetCradleMountConfig\_t Struct Reference

### Data Fields

- uint32\_t [state](#)
- int [has\\_confidence](#)
- uint8\_t [confidence](#)

### 8.192.1 Detailed Description

This structure contains parameters to set current cradle mount configuration Please check has\_<Param\_Name> field for presence of optional parameters

#### Parameters

<i>state</i>	<ul style="list-style-type: none"> <li>• Cradle Mount State</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– 0 - Device is mounted on the cradle</li> <li>– 1 - Device is not mounted on the cradle</li> <li>– 2 - Unknown cradle mount state</li> </ul> </li> </ul>
<i>confidence</i>	<ul style="list-style-type: none"> <li>• Cradle Mount Confidence (Optional)</li> <li>• Confidence in the Cradle Mount state expressed as a percentage.</li> <li>• Range - 0 to 100</li> </ul>

### 8.192.2 Field Documentation

8.192.2.1 uint8\_t pack\_loc\_SLQSLOCSetCradleMountConfig\_t::confidence

8.192.2.2 int pack\_loc\_SLQSLOCSetCradleMountConfig\_t::has\_confidence

8.192.2.3 uint32\_t pack\_loc\_SLQSLOCSetCradleMountConfig\_t::state

## 8.193 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](#)



### 8.193.1 Detailed Description

This structure contains the LOC Start pack

#### Parameters

<i>SessionId</i>	<ul style="list-style-type: none"> <li>• ID of the session as identified by the control point.</li> <li>• Range: 0 to 255</li> </ul>
<i>pRecurrence↔ Type</i>	<ul style="list-style-type: none"> <li>• Specifies the type of session in which the control point is interested.</li> <li>• Defaults to SINGLE. -Values <ul style="list-style-type: none"> <li>– 1 - Request periodic position fixes</li> <li>– 2 - Request a single position fix</li> </ul> </li> </ul>
<i>pHorizontal↔ AccuracyLvl</i>	<ul style="list-style-type: none"> <li>• Specifies the horizontal accuracy level required by the control point.</li> <li>• Defaults to LOW</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Low accuracy</li> <li>– 2 - Medium accuracy</li> <li>– 3 - High accuracy</li> </ul> </li> </ul>
<i>pIntermediate↔ ReportState</i>	<ul style="list-style-type: none"> <li>• Specifies if the control point is interested in receiving intermediate reports.</li> <li>• ON by default.</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Intermediate reports are turned on</li> <li>– 2 - Intermediate reports are turned off</li> </ul> </li> </ul>
<i>appVersionValid</i>	<ul style="list-style-type: none"> <li>• Specifies whether the application version string contains a valid value</li> <li>• 0x00 (FALSE) – Application version string is invalid</li> <li>• 0x01 (TRUE) – Application version string is valid</li> </ul>
<i>LocApplication↔ Info</i>	<ul style="list-style-type: none"> <li>• LOC Application Parameters</li> <li>• See <a href="#">loc_LocApplicationInfo</a> for more information</li> </ul>
<i>pConfig↔ AltitudeAssumed</i>	<ul style="list-style-type: none"> <li>• Configuration for Altitude Assumed Info in GNSS SV Info Event</li> <li>• Defaults to ENABLED.</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Enable Altitude Assumed information in GNSS SV Info Event</li> <li>– 2 - Disable Altitude Assumed information in GNSS SV Info Event</li> </ul> </li> </ul>

### 8.193.2 Field Documentation

#### 8.193.2.1 loc\_LocApplicationInfo\* pack\_loc\_Start\_t::pApplicationInfo

8.193.2.2 uint32\_t\* pack\_loc\_Start\_t::pConfigAltitudeAssumed

8.193.2.3 uint32\_t\* pack\_loc\_Start\_t::pHorizontalAccuracyLvl

8.193.2.4 uint32\_t\* pack\_loc\_Start\_t::pIntermediateReportState

8.193.2.5 uint32\_t\* pack\_loc\_Start\_t::pMinIntervalTime

8.193.2.6 uint32\_t\* pack\_loc\_Start\_t::pRecurrenceType

8.193.2.7 uint8\_t pack\_loc\_Start\_t::SessionId

8.193.2.8 uint16\_t pack\_loc\_Start\_t::Tlvresult

## 8.194 pack\_loc\_Stop\_t Struct Reference

### Data Fields

- uint8\_t [SessionId](#)
- uint16\_t [Tlvresult](#)

### 8.194.1 Detailed Description

This structure contains Stop LOC pack

#### Parameters

<i>sessionId</i>	<ul style="list-style-type: none"> <li>• ID of the session as identified by the control point.</li> <li>• Range: 0 to 255</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> </ul>

### 8.194.2 Field Documentation

8.194.2.1 uint8\_t pack\_loc\_Stop\_t::SessionId

8.194.2.2 uint16\_t pack\_loc\_Stop\_t::Tlvresult

## 8.195 pack\_nas\_SetACCOLC\_t Struct Reference

### Data Fields

- int8\_t [spc](#) [6]
- uint8\_t [accolc](#)

## 8.195.1 Detailed Description

## Parameters

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

## 8.195.2 Field Documentation

8.195.2.1 uint8\_t pack\_nas\_SetACCOLC\_t::accolc

8.195.2.2 int8\_t pack\_nas\_SetACCOLC\_t::spc[6]

## 8.196 pack\_nas\_SetNetworkPreference\_t Struct Reference

## Data Fields

- uint32\_t [TechnologyPref](#)
- uint32\_t [Duration](#)
- uint16\_t [Tlvresult](#)

## 8.196.1 Detailed Description

## Parameters

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

## 8.196.2 Field Documentation

8.196.2.1 uint32\_t pack\_nas\_SetNetworkPreference\_t::Duration

8.196.2.2 uint32\_t pack\_nas\_SetNetworkPreference\_t::TechnologyPref

8.196.2.3 uint16\_t pack\_nas\_SetNetworkPreference\_t::Tlvresult

## 8.197 pack\_nas\_SLQSGetPLMNName\_t Struct Reference

### Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- uint8\_t \* [pMncPcsStatus](#)

### 8.197.1 Detailed Description

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"> <li>• A 16-bit integer representation of MCC. Range: 0 to 999</li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>• A 16-bit integer representation of MNC. Range: 0 to 999</li> </ul>
<i>pMncPcsStatus</i>	<ul style="list-style-type: none"> <li>• MNC PCS Digit Include Status</li> <li>• Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01).</li> <li>• Values             <ul style="list-style-type: none"> <li>– TRUE - MNC is a three-digit value. e.g. a reported value of 90 corresponds to an MNC value of 090</li> <li>– FALSE - MNC is a two-digit value. e.g. a reported value of 90 corresponds to an MNC value of 90</li> </ul> </li> </ul>

#### Note

If pMncPcsStatus is not present, an MNC smaller than 100 is assumed to be a two-digit value, and an MNC greater than or equal to 100 is assumed to be a three digit value.

## 8.197.2 Field Documentation

8.197.2.1 uint16\_t pack\_nas\_SLQSGetPLMNName\_t::mcc

8.197.2.2 uint16\_t pack\_nas\_SLQSGetPLMNName\_t::mnc

8.197.2.3 uint8\_t\* pack\_nas\_SLQSGetPLMNName\_t::pMncPcsStatus

## 8.198 pack\_nas\_SLQSIInitiateNetworkRegistration\_t Struct Reference

### Data Fields

- uint32\_t [regAction](#)
- [nas\\_MNRInfo](#) \* [pMNRInfo](#)
- uint32\_t \* [pChangeDuration](#)
- uint8\_t \* [pMncPcsDigitStatus](#)

### 8.198.1 Detailed Description

This structure contains Initiate Network Registration request parameters

#### Parameters

<i>regAction</i>	<ul style="list-style-type: none"> <li>• Specifies one of the following register actions :             <ul style="list-style-type: none"> <li>– AUTO_REGISTER - Device registers according to its provisioning and optional parameters supplied with the command are ignored.</li> <li>– MANUAL_REGISTER - Device registers to a specified network and the optional Manual Network Register Information parameter <a href="#">pMNRInfo</a> must also be included for the command to process successfully and supported only for 3GPP.</li> </ul> </li> </ul>
<i>pMNRInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• Pointer to structure MNRInfo             <ul style="list-style-type: none"> <li>– See <a href="#">nas_MNRInfo</a> for more information</li> </ul> </li> </ul>
<i>pChange↔ Duration</i>	[Optional] <ul style="list-style-type: none"> <li>• Duration of the change.             <ul style="list-style-type: none"> <li>– 0x00 - Power cycle - Remains active until the next device power cycle</li> <li>– 0x01 - Permanent - Remains active through power cycles until changed by the client</li> </ul> </li> </ul>
<i>pMncPcsDigit↔ Status</i>	[Optional] <ul style="list-style-type: none"> <li>• MNC PCS Digit Include Status             <ul style="list-style-type: none"> <li>– True - MNC is a 3-digit value.</li> <li>– False - MNC is a 2-digit value.</li> </ul> </li> </ul>

### 8.198.2 Field Documentation

8.198.2.1 uint32\_t\* pack\_nas\_SLQSIInitiateNetworkRegistration\_t::pChangeDuration

8.198.2.2 uint8\_t\* pack\_nas\_SLQSIInitiateNetworkRegistration\_t::pMncPcsDigitStatus

8.198.2.3 nas\_MNRInfo\* pack\_nas\_SLQSIInitiateNetworkRegistration\_t::pMNRInfo

8.198.2.4 uint32\_t pack\_nas\_SLQSIInitiateNetworkRegistration\_t::regAction

## 8.199 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\_HDRIOThresh \* pHDRIOThresh
- uint16\_t \* pHDRIODelta
- nas\_GSMRSSIThresh \* pGSMRSSIThresh
- uint16\_t \* pGSMRSSIDelta
- nas\_WCDMARSSIThresh \* pWCDMARSSIThresh
- uint16\_t \* pWCDMARSSIDelta
- nas\_WCDMAECIOThresh \* pWCDMAECIOThresh
- uint16\_t \* pWCDMAECIODelta
- nas\_LTERSSIThresh \* pLTERSSIThresh
- uint16\_t \* pLTERSSIDelta
- nas\_LTESNRThreshold \* pLTESNRThresh
- uint16\_t \* pLTESNRDelta
- nas\_LTERSRQThresh \* pLTERSRQThresh
- uint16\_t \* pLTERSRQDelta
- nas\_LTERSRPThresh \* pLTERSRPThresh
- uint16\_t \* pLTERSRPDelta
- nas\_LTESigRptConfig \* pLTESigRptConfig
- nas\_TDSCDMARSCPTThresh \* pTDSCDMARSCPTThresh
- uint16\_t \* pTDSCDMARSCPDelta
- nas\_TDSCDMARSSIThresh \* pTDSCDMARSSIThresh
- float \* pTDSCDMARSSIDelta
- nas\_TDSCDMAECIOThresh \* pTDSCDMAECIOThresh
- float \* pTDSCDMAECIODelta
- nas\_TDSCDMASINRThresh \* pTDSCDMASINRThresh
- float \* pTDSCDMASINRDelta

### 8.199.1 Detailed Description

## Parameters

<i>pCDMARSSI</i> <i>Thresh</i>	<ul style="list-style-type: none"> <li>CDMA RSSI threshold List</li> </ul>
<i>pCDMARSSI</i> <i>Delta</i>	<ul style="list-style-type: none"> <li>RSSI delta (in units of 0.1 dBm).</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pCDMAECIO</i> <i>Thresh</i>	<ul style="list-style-type: none"> <li>CDMA ECIO Threshold List</li> </ul>
<i>pCDMAECIO</i> <i>Delta</i>	<ul style="list-style-type: none"> <li>ECIO delta (in units of 0.1 dB).</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHRRSSI</i> <i>Thresh</i>	<ul style="list-style-type: none"> <li>HDR RSSI Threshold List</li> </ul>
<i>pHRRSSIDelta</i>	<ul style="list-style-type: none"> <li>RSSI delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHRECIOTresh</i>	<ul style="list-style-type: none"> <li>HDR ECIO Threshold List</li> </ul>
<i>pHRECIODelta</i>	<ul style="list-style-type: none"> <li>ECIO delta (in units of 0.1 dB)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHRSINR</i> <i>Thresh</i>	<ul style="list-style-type: none"> <li>HDR SINR Threshold List</li> </ul>
<i>pHRSINRDelta</i>	<ul style="list-style-type: none"> <li>SINR delta (in units of 1 SINR level)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pHDRIOTresh</i>	<ul style="list-style-type: none"> <li>HDR IO Threshold List</li> </ul>
<i>pHDRIODelta</i>	<ul style="list-style-type: none"> <li>IO delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pGSMRSSI</i> <i>Thresh</i>	<ul style="list-style-type: none"> <li>GSM RSSI Threshold List</li> <li>See GSMRSSIThresh for more details</li> </ul>
<i>pGSMRSSIDelta</i>	<ul style="list-style-type: none"> <li>RSSI delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>

<i>pWCDMARSS</i> ↔ <i>IThresh</i>	<ul style="list-style-type: none"> <li>WCDMA RSSI Threshold List</li> <li>See WCDMARSSIThresh for more details</li> </ul>
<i>pWCDMARSS</i> ↔ <i>IDelta</i>	<ul style="list-style-type: none"> <li>RSSI delta (in units of 0.1 dBm).</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pWCDMAECI</i> ↔ <i>OThresh</i>	<ul style="list-style-type: none"> <li>WCDMA ECIO Threshold List</li> </ul>
<i>pWCDMAECI</i> ↔ <i>ODelta</i>	<ul style="list-style-type: none"> <li>ECIO delta (in units of 0.1 dB)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSSI</i> ↔ <i>Thresh</i>	<ul style="list-style-type: none"> <li>LTE RSSI Threshold List</li> </ul>
<i>pLTERSSIDelta</i>	<ul style="list-style-type: none"> <li>RSSI delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSNR</i> ↔ <i>Thresh</i>	<ul style="list-style-type: none"> <li>LTE SNR Threshold List</li> </ul>
<i>pLTERSNRDelta</i>	<ul style="list-style-type: none"> <li>SNR delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSRQ</i> ↔ <i>Thresh</i>	<ul style="list-style-type: none"> <li>LTE RSRQ Threshold List</li> </ul>
<i>pLTERSRQ</i> ↔ <i>Delta</i>	<ul style="list-style-type: none"> <li>RSRQ delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSRP</i> ↔ <i>Thresh</i>	<ul style="list-style-type: none"> <li>LTE RSRP Threshold List</li> </ul>
<i>pLTERSRPDelta</i>	<ul style="list-style-type: none"> <li>RSRP delta (in units of 0.1 dBm).</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>
<i>pLTERSigRpt</i> ↔ <i>Config</i>	<ul style="list-style-type: none"> <li>LTE Signal Report Config</li> </ul>
<i>pTDSCDMAR</i> ↔ <i>SCPThresh</i>	<ul style="list-style-type: none"> <li>TDSCDMA RSCP Threshold List</li> </ul>
<i>pTDSCDMAR</i> ↔ <i>SCPDelta</i>	<ul style="list-style-type: none"> <li>RSCP delta (in units of 0.1 dBm)</li> <li>A value of 0 is rejected with a QMI_ERR_INVALID_ARG error.</li> </ul>



<i>pTDSCDMARSSIThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA RSSI Threshold List</li> </ul>
<i>pTDSCDMARSSIDelta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in dBm) used by TD-SCDMA.</li> </ul>
<i>pTDSCDMAECIOThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA ECIO Threshold List</li> </ul>
<i>pTDSCDMAECIODelta</i>	<ul style="list-style-type: none"> <li>• ECIO delta (in dB) used by TD-SCDMA</li> </ul>
<i>pTDSCDMASINRThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA SINR Threshold List</li> </ul>
<i>pTDSCDMASINRDelta</i>	<ul style="list-style-type: none"> <li>• SINR delta (in dB) used by TD-SCDMA.</li> </ul>

## 8.199.2 Field Documentation

8.199.2.1 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta`

8.199.2.2 `nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh`

8.199.2.3 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta`

8.199.2.4 `nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh`

8.199.2.5 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta`

8.199.2.6 `nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh`

8.199.2.7 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta`

8.199.2.8 `nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh`

8.199.2.9 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta`

8.199.2.10 `nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh`

8.199.2.11 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIDelta`

8.199.2.12 `nas_HDRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIThresh`

8.199.2.13 `uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRDelta`

- 8.199.2.14 `nas_HDRSINRThreshold*` `pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRThresh`
- 8.199.2.15 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta`
- 8.199.2.16 `nas_LTERSRPThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh`
- 8.199.2.17 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta`
- 8.199.2.18 `nas_LTERSRQThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh`
- 8.199.2.19 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta`
- 8.199.2.20 `nas_LTERSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh`
- 8.199.2.21 `nas_LTESigRptConfig*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig`
- 8.199.2.22 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta`
- 8.199.2.23 `nas_LTESNRThreshold*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh`
- 8.199.2.24 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta`
- 8.199.2.25 `nas_TDSCDMAECIOThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh`
- 8.199.2.26 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta`
- 8.199.2.27 `nas_TDSCDMARSCPThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh`
- 8.199.2.28 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta`
- 8.199.2.29 `nas_TDSCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh`
- 8.199.2.30 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta`
- 8.199.2.31 `nas_TDSCDMASINRThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh`
- 8.199.2.32 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta`
- 8.199.2.33 `nas_WCDMAECIOThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh`
- 8.199.2.34 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta`
- 8.199.2.35 `nas_WCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh`

## 8.200 `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](#)

### 8.200.1 Detailed Description

#### Parameters

<i>pSystem↔ SelectionInd</i>	[Optional] <ul style="list-style-type: none"> <li>System Selection Preference indication registration. The following callbacks would not be invoked if the indication is disabled. tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pDDTMInd</i>	[Optional] <ul style="list-style-type: none"> <li>DDTM (Data Dedicated Transmission Mode) indication registration. The following callbacks would not be invoked if the indication is disabled. tFNDDTM               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pServing↔ SystemInd</i>	[Optional] <ul style="list-style-type: none"> <li>Serving System indication registration. The following callbacks would not be invoked if the indication is disabled. tFNBandPreference               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pDualStandBy↔ PrefInd</i>	[Optional] <ul style="list-style-type: none"> <li>Dual Standby Preference indication registration. The following callbacks would not be invoked if the indication is disabled. tFNDualStandByPref               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSubscription↔ InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>Subscription Information indication registration. The following callbacks would not be invoked if the indication is disabled. tFNSubscriptionInfo               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pNetworkTime</i> ↔ <i>Ind</i>	[Optional] <ul style="list-style-type: none"> <li>Network Time indication registration. The following callbacks would not be invoked if the indication is disabled. tFNNetworkTime             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSysInfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>System Information indication registration. The following callbacks would not be invoked if the indication is disabled. tFNSysInfo             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSignal</i> ↔ <i>StrengthInd</i>	[Optional] <ul style="list-style-type: none"> <li>Signal Strength indication registration. The following callbacks would not be invoked if the indication is disabled. tFNSigInfo             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pErrorRateInd</i>	[Optional] <ul style="list-style-type: none"> <li>Error Rate indication registration. The following callbacks would not be invoked if the indication is disabled. tFNErrRate             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pHDRNewUA</i> ↔ <i>TIAssInd</i>	[Optional] <ul style="list-style-type: none"> <li>HDR New UATI Assigned indication registration. The following callbacks would not be invoked if the indication is disabled. tFNHDRUATIUpdate             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pHDRSession</i> ↔ <i>CloseInd</i>	[Optional] <ul style="list-style-type: none"> <li>HDR Session Closed indication registration. The following callbacks would not be invoked if the indication is disabled. tFNHDRSessionClose             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pManaged</i> ↔ <i>RoamingInd</i>	[Optional] <ul style="list-style-type: none"> <li>Managed Roaming indication registration. The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pLTECphyCa</i>	[Optional] <ul style="list-style-type: none"> <li>• LTE Physical Carrier Aggregation Information. The following callbacks would not be invoked if the indication is disabled.</li> </ul> tFNManagedRoaming <ul style="list-style-type: none"> <li>– 0x00 - Disable (default value)</li> <li>– 0x01 - Enable</li> </ul>
-------------------	--

## 8.200.2 Field Documentation

8.200.2.1 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pDDTMInd

8.200.2.2 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pDualStandByPrefInd

8.200.2.3 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pErrorRateInd

8.200.2.4 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pHDRNewUATIAssInd

8.200.2.5 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pHDRSessionCloseInd

8.200.2.6 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pLTECphyCa

8.200.2.7 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pManagedRoamingInd

8.200.2.8 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pNetworkTimeInd

8.200.2.9 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pServingSystemInd

8.200.2.10 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSignalStrengthInd

8.200.2.11 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSubscriptionInfoInd

8.200.2.12 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSysInfoInd

8.200.2.13 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSystemSelectionInd

## 8.201 pack\_nas\_SLQSNasSwlIndicationRegister\_t Struct Reference

### Data Fields

- uint8\_t lteEsmUI
- uint8\_t lteEsmDI
- uint8\_t lteEmmUI
- uint8\_t lteEmmDI
- uint8\_t gsmUmtsUI
- uint8\_t gsmUmtsDI
- uint8\_t \* pRankIndicatorInd
- uint8\_t \* pTimer

### 8.201.1 Detailed Description

This structure contains the OTA message indication.

#### Parameters

<i>lteEsmUI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report LTE ESM uplink messages</li> </ul>
<i>lteEsmDI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report LTE ESM downlink messages</li> </ul>
<i>lteEmmUI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report LTE EMM uplink messages</li> </ul>
<i>lteEmmDI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report GSM/UMTS uplink messages</li> </ul>
<i>gsmUmtsUI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report GSM/UMTS uplink messages</li> </ul>
<i>gsmUmtsDI</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report GSM/UMTS downlink messages</li> </ul>
<i>pRank↔ IndicatorInd</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report Rank Indicator messages</li> </ul>
<i>pTimer</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report Timer Indicator messages</li> </ul>

## 8.201.2 Field Documentation

8.201.2.1 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDI`

8.201.2.2 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUI`

8.201.2.3 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDI`

8.201.2.4 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUI`

8.201.2.5 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDI`

8.201.2.6 `uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUI`

8.201.2.7 `uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd`

8.201.2.8 uint8\_t\* pack\_nas\_SLQSNasSwiIndicationRegister\_t::pTimer

## 8.202 pack\_nas\_SLQSSetSignalStrengthsCallback\_t Struct Reference

### Data Fields

- uint8\_t [bEnable](#)
- [nas\\_SLQSSignalStrengthsIndReq](#) \* [pSigIndReq](#)

### 8.202.1 Detailed Description

#### Parameters

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

### 8.202.2 Field Documentation

8.202.2.1 uint8\_t pack\_nas\_SLQSSetSignalStrengthsCallback\_t::bEnable

8.202.2.2 [nas\\_SLQSSignalStrengthsIndReq](#)\* pack\_nas\_SLQSSetSignalStrengthsCallback\_t::pSigIndReq

## 8.203 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](#)

### 8.203.1 Detailed Description

Contain the system selection preferences.

#### Parameters

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



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

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

<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>- netSelectionPref</li> <li>• Optional parameter for specifying Network Selection Preference</li> <li>• Modem selects networks based on this parameter(if present).</li> <li>• see netSelectionPref for more information</li> </ul>
<i>pChgDuration</i>	<ul style="list-style-type: none"> <li>• Optional parameter specifying the duration of the change</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Power cycle - Remains active until the next device power cycle</li> <li>– 0x01 - Permanent - Remains active through power cycles until changed by client</li> <li>– Device will use "0x01 - permanent" as default if this parameter is omitted</li> </ul> </li> </ul>
<i>pMNCIncPCS↔ DigStat</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating if MNC includes PCS digit</li> <li>• Values: <ul style="list-style-type: none"> <li>– TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090</li> <li>– FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90</li> </ul> </li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Circuit switched only</li> <li>– 0x01 - Packet switched only</li> <li>– 0x02 - Circuit switched and packet switched</li> <li>– 0x03 - Packet switched attach</li> <li>– 0x04 - Packet switched detach</li> </ul> </li> </ul>
<i>pGWAcqOrder↔ Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic</li> <li>– 0x01 - GSM then WCDMA</li> <li>– 0x02 - WCDMA then GSM</li> </ul> </li> </ul>

## 8.203.2 Field Documentation

8.203.2.1 struct nas\_acqOrderPref\* pack\_nas\_SLQSSetSysSelectionPref\_t::pAcqOrderPref

8.203.2.2 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pBandPref

8.203.2.3 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pChgDuration

8.203.2.4 struct nas\_CSGID\* pack\_nas\_SLQSSetSysSelectionPref\_t::pCSGID

8.203.2.5 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pEmerMode

8.203.2.6 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pGWAcqOrderPref

8.203.2.7 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pLTEBandPref

8.203.2.8 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pMNCIncPCSDigStat

8.203.2.9 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pModePref

8.203.2.10 struct nas\_netSelectionPref\* pack\_nas\_SLQSSetSysSelectionPref\_t::pNetSelPref

8.203.2.11 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pPRLPref

8.203.2.12 unsigned char\* pack\_nas\_SLQSSetSysSelectionPref\_t::pRAT

8.203.2.13 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pRoamPref

8.203.2.14 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pSrvDomainPref

8.203.2.15 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pSrvRegRestriction

8.203.2.16 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pTdsdmaBandPref

## 8.204 pack\_qmi\_t Struct Reference

### Data Fields

- uint16\_t [xid](#)
- int [timeout](#)
- uint16\_t [msgid](#)
- uint8\_t [svc](#)

### 8.204.1 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

### 8.204.2 Field Documentation

8.204.2.1 uint16\_t pack\_qmi\_t::msgid

8.204.2.2    `uint8_t pack_qmi_t::svc`

8.204.2.3    `int pack_qmi_t::timeout`

8.204.2.4    `uint16_t pack_qmi_t::xid`

## 8.205   pack\_qos\_SLQSQosSwiReadApnExtraParams\_t Struct Reference

### Data Fields

- `uint32_t` [apnId](#)

### 8.205.1   Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

#### Parameters

<i>apnId</i> [IN]	<ul style="list-style-type: none"><li>• APN id</li></ul>
-------------------	--

### 8.205.2   Field Documentation

8.205.2.1    `uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId`

## 8.206   pack\_qos\_SLQSQosSwiReadDataStats\_t Struct Reference

### Data Fields

- `uint32_t` [apnId](#)

### 8.206.1   Detailed Description

Structure that contains the APN ID to obtain data statistics

#### Parameters

<i>apnId</i> [IN]	<ul style="list-style-type: none"><li>• APN id</li></ul>
-------------------	--

### 8.206.2   Field Documentation

8.206.2.1 `uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId`

## 8.207 `pack_qos_SLQSSetQosEventCallback_t` Struct Reference

### Data Fields

- `uint8_t enable`

#### 8.207.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

##### Parameters

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

#### 8.207.2 Field Documentation

8.207.2.1 `uint8_t pack_qos_SLQSSetQosEventCallback_t::enable`

## 8.208 `pack_sms_SendSMS_t` Struct Reference

### Data Fields

- `uint32_t messageFormat`
- `uint32_t messageSize`
- `uint8_t * pMessage`
- `uint8_t * pLinktimer`

#### 8.208.1 Detailed Description

##### Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>pLinktimer</i>	<ul style="list-style-type: none"> <li>• GW SMS link open for the specified number of second</li> </ul>
<i>pMessage</i>	<div>Generated by Doxygen</div> <ul style="list-style-type: none"> <li>• The message contents in PDU format contains SMS header and payload message</li> </ul>

### 8.208.2 Field Documentation

8.208.2.1 uint32\_t pack\_sms\_SendSMS\_t::messageFormat

8.208.2.2 uint32\_t pack\_sms\_SendSMS\_t::messageSize

8.208.2.3 uint8\_t\* pack\_sms\_SendSMS\_t::pLinktimer

8.208.2.4 uint8\_t\* pack\_sms\_SendSMS\_t::pMessage

## 8.209 pack\_sms\_SetNewSMSCallback\_t Struct Reference

### Data Fields

- enum [eqmiCbkJSetStatus status](#)

### 8.209.1 Detailed Description

#### Parameters

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

### 8.209.2 Field Documentation

8.209.2.1 enum [eqmiCbkJSetStatus](#) pack\_sms\_SetNewSMSCallback\_t::status

## 8.210 pack\_sms\_SLQSDeleteSMS\_t Struct Reference

### Data Fields

- uint32\_t [storageType](#)
- uint32\_t \* [pMessageIndex](#)
- uint32\_t \* [pMessageTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.210.1 Detailed Description

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"><li>• SMS message storage type<ul style="list-style-type: none"><li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li><li>– 1 - NV</li></ul></li></ul>
--------------------	---

<i>pMessageIndex</i>	<ul style="list-style-type: none"> <li>• (Optional) message index</li> </ul>
<i>pMessageTag</i>	<ul style="list-style-type: none"> <li>• (Optional) message tag <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> <li>– 2 - Mobile originated and sent</li> <li>– 3 - Mobile originated but not yet sent</li> </ul> </li> </ul>
<i>pMessageMode</i>	<ul style="list-style-type: none"> <li>• (Optional) message mode</li> <li>• this must be included if the device is capable of supporting more than one protocol</li> <li>• e.g. CDMA and GW <ul style="list-style-type: none"> <li>– 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>– 0x01 - GW, LTE (if network type is UMTS)</li> </ul> </li> </ul>

## 8.210.2 Field Documentation

8.210.2.1 `uint32_t* pack_sms_SLQSDDeleteSMS_t::pMessageIndex`

8.210.2.2 `uint8_t* pack_sms_SLQSDDeleteSMS_t::pMessageMode`

8.210.2.3 `uint32_t* pack_sms_SLQSDDeleteSMS_t::pMessageTag`

8.210.2.4 `uint32_t pack_sms_SLQSDDeleteSMS_t::storageType`

## 8.211 pack\_sms\_SLQSGetSMS\_t Struct Reference

### Data Fields

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

### 8.211.1 Detailed Description

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>• Message index</li> </ul>
<i>pMessageMode</i>	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>



### 8.211.2 Field Documentation

8.211.2.1 uint32\_t pack\_sms\_SLQSGetSMS\_t::messageIndex

8.211.2.2 uint8\_t\* pack\_sms\_SLQSGetSMS\_t::pMessageMode

8.211.2.3 uint32\_t pack\_sms\_SLQSGetSMS\_t::storageType

## 8.212 pack\_sms\_SLQSGetSMSList\_t Struct Reference

### Data Fields

- uint32\_t [storageType](#)
- uint32\_t \* [pRequestedTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.212.1 Detailed Description

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>requestedTag</i>	<ul style="list-style-type: none"> <li>• (Optional) Message tag <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> <li>– 2 - Mobile originated and sent</li> <li>– 3 - Mobile originated but not yet sent</li> </ul> </li> </ul>
<i>messageMode</i>	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>

### 8.212.2 Field Documentation

8.212.2.1 uint8\_t\* pack\_sms\_SLQSGetSMSList\_t::pMessageMode

8.212.2.2 uint32\_t\* pack\_sms\_SLQSGetSMSList\_t::pRequestedTag

8.212.2.3 uint32\_t pack\_sms\_SLQSGetSMSList\_t::storageType

## 8.213 pack\_sms\_SLQSMModifySMSStatus\_t Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)
- uint32\_t [messageTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.213.1 Detailed Description

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>• Message index</li> </ul>
<i>messageTag</i>	<ul style="list-style-type: none"> <li>• Message tag <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> </ul> </li> </ul>
<i>pMessageMode</i>	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>

### 8.213.2 Field Documentation

8.213.2.1 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::messageIndex

8.213.2.2 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::messageTag

8.213.2.3 uint8\_t\* pack\_sms\_SLQSMModifySMSStatus\_t::pMessageMode

8.213.2.4 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::storageType

## 8.214 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](#)

## 8.214.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

## Parameters

<i>function</i>	<ul style="list-style-type: none"> <li>Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> <li>0 - disabled</li> <li>1 - At bootup</li> <li>2 - When NMEA port is opened</li> </ul> </li> </ul>
<i>set_function</i>	<ul style="list-style-type: none"> <li>0 - do not set to modem</li> <li>1 - set to modem</li> </ul>
<i>fix_type</i>	<ul style="list-style-type: none"> <li>Type of GNSS fix: <ul style="list-style-type: none"> <li>1 - Default Engine mode</li> <li>2 - MS-Based</li> <li>3 - MS-Assisted</li> <li>4 - Standalone</li> </ul> </li> </ul>
<i>set_fix_type</i>	<ul style="list-style-type: none"> <li>0 - do not set to modem</li> <li>1 - set to modem</li> </ul>
<i>max_time</i>	<ul style="list-style-type: none"> <li>Maximum time allowed for the receiver to get a fix in seconds</li> <li>Valid range: 1-255</li> </ul>
<i>set_max_time</i>	<ul style="list-style-type: none"> <li>0 - do not set to modem</li> <li>1 - set to modem</li> </ul>
<i>max_dist</i>	<ul style="list-style-type: none"> <li>Maximum uncertainty of a fix measured by distance in meters</li> <li>Valid range: 1 - 4294967280</li> </ul>
<i>set_max_dist</i>	<ul style="list-style-type: none"> <li>0 - do not set to modem</li> <li>1 - set to modem</li> </ul>
<i>fix_rate</i>	<ul style="list-style-type: none"> <li>Time between fixes in seconds</li> <li>Valid range: 1–65535</li> </ul>
<i>set_fix_rate</i>	<ul style="list-style-type: none"> <li>0 - do not set to modem</li> <li>1 - set to modem</li> </ul>

## 8.214.2 Field Documentation

8.214.2.1 uint32\_t pack\_swiloc\_SwiLocSetAutoStart\_t::fix\_rate

8.214.2.2 uint8\_t pack\_swiloc\_SwiLocSetAutoStart\_t::fix\_type

8.214.2.3 uint8\_t pack\_swiloc\_SwiLocSetAutoStart\_t::function

8.214.2.4 uint32\_t pack\_swiloc\_SwiLocSetAutoStart\_t::max\_dist

8.214.2.5 uint8\_t pack\_swiloc\_SwiLocSetAutoStart\_t::max\_time

8.214.2.6 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_fix\_rate

8.214.2.7 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_fix\_type

8.214.2.8 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_function

8.214.2.9 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_max\_dist

8.214.2.10 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_max\_time

## 8.215 pack\_swiooma\_SLQSOMADMCancelSession\_t Struct Reference

### Data Fields

- uint32\_t [sessionType](#)

### 8.215.1 Detailed Description

Structure that contains the session type for OMA cancel session command

#### Parameters

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

## 8.215.2 Field Documentation

8.215.2.1 uint32\_t pack\_swiooma\_SLQSOMADMCancelSession\_t::sessionType

## 8.216 pack\_swisma\_SLQSOMADMGetSessionInfo\_t Struct Reference

### Data Fields

- uint32\_t [SessionType](#)

### 8.216.1 Detailed Description

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

#### Parameters

<i>SessionType</i> [IN]	<ul style="list-style-type: none"> <li>• Session type           <ul style="list-style-type: none"> <li>– 0x01 - FOTA</li> <li>– 0xFF - Any active OMADM session. If no active sessions are available, then previous O↔MADM session info is returned</li> </ul> </li> </ul>
-------------------------	--

### 8.216.2 Field Documentation

8.216.2.1 uint32\_t pack\_swisma\_SLQSOMADMGetSessionInfo\_t::SessionType

## 8.217 pack\_swisma\_SLQSOMADMSendSelection\_t Struct Reference

### Data Fields

- uint32\_t [selection](#)
- uint32\_t \* [pDeferTime](#)
- uint32\_t \* [pRejectReason](#)

### 8.217.1 Detailed Description

Structure containing the OMA DM selection

#### Parameters

<i>selection</i> [IN]	<ul style="list-style-type: none"> <li>• OMA-DM NIA Selection           <ul style="list-style-type: none"> <li>– 0x01 - Accept</li> <li>– 0x02 - Reject</li> <li>– 0x03 - Defer</li> </ul> </li> </ul>
<i>pDeferTime</i> [IN]	<ul style="list-style-type: none"> <li>• Defer time in minutes. A value of 0 will cause the prompt to be resent immediately.</li> <li>• This TLV is mandatory if selection is set to 0x03.</li> </ul>
<i>pRejectReason</i> [IN]	<ul style="list-style-type: none"> <li>• Reject Reason</li> <li>• This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default.</li> </ul>

## 8.217.2 Field Documentation

8.217.2.1 uint32\_t\* pack\_swima\_SLQSOMADMSendSelection\_t::pDeferTime

8.217.2.2 uint32\_t\* pack\_swima\_SLQSOMADMSendSelection\_t::pRejectReason

8.217.2.3 uint32\_t pack\_swima\_SLQSOMADMSendSelection\_t::selection

## 8.218 pack\_swima\_SLQSOMADMSetSettings\_t Struct Reference

### Data Fields

- uint8\_t [FOTAdownload](#)
- uint8\_t [FOTAUpdate](#)
- uint8\_t \* [pAutosdm](#)
- uint8\_t \* [pFwAutoCheck](#)

### 8.218.1 Detailed Description

Structure containing the OMA DM settings to be set on the device This maps to structure SLQSOMADMSettings↔ ReqParams3

#### Parameters

<i>FOTAdownload</i>	<ul style="list-style-type: none"> <li>• 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>– 0x00 - Firmware autodownload FALSE</li> <li>– 0x01 - Firmware autodownload TRUE</li> </ul> </li> </ul>
<i>FOTAUpdate</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>– 0x00 - Firmware autoupdate FALSE</li> <li>– 0x01 - Firmware autoupdate TRUE</li> </ul> </li> </ul>
<i>pAutosdm[IN]</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled Accept</li> <li>– 0x02 - Enabled Reject</li> </ul> </li> </ul>
<i>pFwAuto↔ Check[IN]</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> </ul> </li> </ul>

## 8.218.2 Field Documentation

8.218.2.1 uint8\_t pack\_swima\_SLQSOMADMSetSettings\_t::FOTAdownload

8.218.2.2 uint8\_t pack\_swima\_SLQSOMADMSetSettings\_t::FOTAUpdate

8.218.2.3 uint8\_t\* pack\_swima\_SLQSOMADMSetSettings\_t::pAutosdm

8.218.2.4 uint8\_t\* pack\_swima\_SLQSOMADMSetSettings\_t::pFwAutoCheck

## 8.219 pack\_swima\_SLQSOMADMStartSession\_t Struct Reference

### Data Fields

- uint32\_t [sessionType](#)

### 8.219.1 Detailed Description

Structure that contains the session type for OMA start session command

#### Parameters

<i>sessionType</i> [IN]	<ul style="list-style-type: none"><li>Session type<ul style="list-style-type: none"><li>0x01 - FOTA, to check availability of FW Update</li><li>0x02 - DM, to check availability of DM Update</li><li>0x03 - PRL, to check availability of PRL Update</li></ul></li></ul>
-------------------------	---

### 8.219.2 Field Documentation

8.219.2.1 uint32\_t pack\_swima\_SLQSOMADMStartSession\_t::sessionType

## 8.220 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](#)

### 8.220.1 Detailed Description

This structure contains information of the request parameters associated with a Change PIN API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See UIMSessionInformation for more information.</li> </ul>
<i>changePIN</i>	<ul style="list-style-type: none"> <li>• See changeUIMPIN for more information.</li> </ul>
<i>pKey</i> ↔ <i>Reference</i> ↔ <i>D(optional)</i>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<i>pIndication</i> ↔ <i>Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>

**Note**

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

**8.220.2 Field Documentation**

8.220.2.1 `uim_changeUIMPIN` `pack_uim_ChangePin_t::changePIN`

8.220.2.2 `uim_encryptedPIN1` `pack_uim_ChangePin_t::EncryptedPIN1`

8.220.2.3 `uint32_t*` `pack_uim_ChangePin_t::pIndicationToken`

8.220.2.4 `uint8_t*` `pack_uim_ChangePin_t::pKeyReferenceID`

8.220.2.5 `uim_sessionInformation` `pack_uim_ChangePin_t::sessionInfo`

8.220.2.6 `uint16_t` `pack_uim_ChangePin_t::Tlvresult`

**8.221 `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`



### 8.221.1 Detailed Description

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

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See UIMSessionInformation for more information.</li> </ul>
<i>fileIndex</i>	<ul style="list-style-type: none"> <li>• See fileInfo for more information.</li> </ul>
<i>readTransparent</i>	<ul style="list-style-type: none"> <li>• See readTransparentInfo for more information.</li> </ul>
<i>pIndication</i> $\leftrightarrow$ <i>Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>
<i>pEncrypt</i> $\leftrightarrow$ <i>Data(optional)</i>	<ul style="list-style-type: none"> <li>• Encrypt Data.</li> <li>• Indicates whether the data read from the card is to be encrypted.</li> </ul>

#### Note

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

### 8.221.2 Field Documentation

8.221.2.1 uim\_fileInfo pack\_uim\_ReadTransparent\_t::fileIndex

8.221.2.2 uint8\_t\* pack\_uim\_ReadTransparent\_t::pEncryptData

8.221.2.3 uint32\_t\* pack\_uim\_ReadTransparent\_t::pIndicationToken

8.221.2.4 uim\_readTransparentInfo pack\_uim\_ReadTransparent\_t::readTransparent

8.221.2.5 uim\_sessionInformation pack\_uim\_ReadTransparent\_t::sessionInfo

8.221.2.6 uint16\_t pack\_uim\_ReadTransparent\_t::Tlvresult

## 8.222 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](#)

### 8.222.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>pinProtection</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_setPINProtection</a> for more information.</li> </ul>
<i>pKey</i> ↔ <i>Reference</i> ↔ <i>D(optional)</i>	<ul style="list-style-type: none"> <li>Indicates the PIN key reference ID.</li> <li>Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<i>pIndication</i> ↔ <i>Token(optional)</i>	<ul style="list-style-type: none"> <li>Response in Indication.</li> <li>When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> <li>Valid Values <ul style="list-style-type: none"> <li>0 - Result of operation in response. Indication will not be generated by the modem</li> <li>Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>

#### Note

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

### 8.222.2 Field Documentation

8.222.2.1 `uim_encryptedPIN1` `pack_uim_SetPinProtection_t::EncryptedPIN1`

8.222.2.2 `uint32_t*` `pack_uim_SetPinProtection_t::pIndicationToken`

8.222.2.3 `uim_setPINProtection` `pack_uim_SetPinProtection_t::pinProtection`

8.222.2.4 `uint8_t*` `pack_uim_SetPinProtection_t::pKeyReferenceID`

8.222.2.5 `uim_sessionInformation` `pack_uim_SetPinProtection_t::sessionInfo`

8.222.2.6 `uint16_t` `pack_uim_SetPinProtection_t::Tlvresult`

## 8.223 `pack_uim_SLQSUIEventRegister_t` Struct Reference

#### Data Fields

- `uint32_t` [eventMask](#)

### 8.223.1 Detailed Description

#### Parameters

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

### 8.223.2 Field Documentation

8.223.2.1 `uint32_t pack_uim_SLQSUIEventRegister_t::eventMask`

## 8.224 pack\_uim\_SLQSUIPowerDown\_t Struct Reference

#### Data Fields

- `uint8_t slot`

### 8.224.1 Detailed Description

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

#### Parameters

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

### 8.224.2 Field Documentation

8.224.2.1 `uint8_t pack_uim_SLQSUIPowerDown_t::slot`

## 8.225 pack\_uim\_SLQSUIPowerUp\_t Struct Reference

#### Data Fields

- `uint8_t slot`
- `uint8_t * plgnoreHotSwapSwitch`

### 8.225.1 Detailed Description

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

#### Parameters

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

## 8.225.2 Field Documentation

8.225.2.1 `uint8_t* pack_uim_SLQSUIMPowerUp_t::plgnoreHotSwapSwitch`

8.225.2.2 `uint8_t pack_uim_SLQSUIMPowerUp_t::slot`

## 8.226 `pack_uim_SLQSUIMSwitchSlot_t` Struct Reference

### Data Fields

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

### 8.226.1 Detailed Description

This structure contains information of the request parameters associated with a Switch Slot.

#### Parameters

<i>bLogicalSlot</i>	<ul style="list-style-type: none"> <li>Indicates the slot to be used. <ul style="list-style-type: none"> <li>1 - Slot 1</li> <li>2 - Slot 2</li> <li>3 - Slot 3</li> <li>4 - Slot 4</li> <li>5 - Slot 5</li> </ul> </li> </ul>
<i>ulPhysicalSlot</i>	<ul style="list-style-type: none"> <li>1 - Slot 1</li> <li>2 - Slot 2</li> <li>3 - Slot 3</li> <li>4 - Slot 4</li> <li>5 - Slot 5</li> </ul>

## 8.226.2 Field Documentation

8.226.2.1 uint8\_t pack\_uim\_SLQSUIMSwitchSlot\_t::bLogicalSlot

8.226.2.2 uint32\_t pack\_uim\_SLQSUIMSwitchSlot\_t::ulPhysicalSlot

## 8.227 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](#)

### 8.227.1 Detailed Description

This structure contains information of the request parameters associated with a Unblock PIN API.

#### Parameters

<i>EncryptedPIN1</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</li> </ul>
<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>pinProtection</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_unblockUIMPIN</a> for more information.</li> </ul>
<i>pKey↔ Reference↔ D(optional)</i>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<i>pIndication↔ Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>

## 8.227.2 Field Documentation

8.227.2.1 uim\_encryptedPIN1 pack\_uim\_UnblockPin\_t::EncryptedPIN1

8.227.2.2 uint32\_t\* pack\_uim\_UnblockPin\_t::pIndicationToken

8.227.2.3 uim\_unblockUIMPIN pack\_uim\_UnblockPin\_t::pinProtection

8.227.2.4 uint8\_t\* pack\_uim\_UnblockPin\_t::pKeyReferenceID

8.227.2.5 uim\_sessionInformation pack\_uim\_UnblockPin\_t::sessionInfo

8.227.2.6 uint16\_t pack\_uim\_UnblockPin\_t::Tlvresult

## 8.228 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](#)

### 8.228.1 Detailed Description

This structure contains information of the request parameters associated with a verify PIN API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See UIMSessionInformation for more information.</li> </ul>
<i>verifyPIN</i>	<ul style="list-style-type: none"> <li>• See verifyUIMPIN for more information.</li> </ul>
<i>pEncryptedPIN1</i> <i>N1(optional)</i>	<ul style="list-style-type: none"> <li>• See encryptedPIN1 for more information.</li> </ul>
<i>pKeyReferenceID</i> <i>D(optional)</i>	<ul style="list-style-type: none"> <li>• Indicates the PIN key reference ID.</li> <li>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.</li> <li>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases.</li> </ul>
<i>pIndicationToken</i> <i>Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>
	Generated by Doxygen

## Note

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

## 8.228.2 Field Documentation

8.228.2.1 uim\_encryptedPIN1\* pack\_uim\_VerifyPin\_t::pEncryptedPIN1

8.228.2.2 uint32\_t\* pack\_uim\_VerifyPin\_t::pIndicationToken

8.228.2.3 uint8\_t\* pack\_uim\_VerifyPin\_t::pKeyReferenceID

8.228.2.4 uim\_sessionInformation pack\_uim\_VerifyPin\_t::sessionInfo

8.228.2.5 uint16\_t pack\_uim\_VerifyPin\_t::Tlvresult

8.228.2.6 uim\_verifyUIMPIN pack\_uim\_VerifyPin\_t::verifyPIN

## 8.229 pack\_wds\_DHCPv4ClientLeaseChange\_t Struct Reference

## Data Fields

- uint8\_t \* [pEnableNotification](#)

## 8.229.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

## Parameters

<i>pEnableNotification</i>	<ul style="list-style-type: none"><li>• Enable Notification or not</li></ul>
----------------------------	--

## 8.229.2 Field Documentation

8.229.2.1 uint8\_t\* pack\_wds\_DHCPv4ClientLeaseChange\_t::pEnableNotification

## 8.230 pack\_wds\_GetDefaultProfile\_t Struct Reference

## Data Fields

- uint32\_t [profiletype](#)

### 8.230.1 Detailed Description

#### Parameters

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

### 8.230.2 Field Documentation

8.230.2.1 uint32\_t pack\_wds\_GetDefaultProfile\_t::profiletype

## 8.231 pack\_wds\_GetDefaultProfileNum\_t Struct Reference

#### Data Fields

- uint8\_t [type](#)
- uint8\_t [family](#)

### 8.231.1 Detailed Description

#### Parameters

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

### 8.231.2 Field Documentation

8.231.2.1 uint8\_t pack\_wds\_GetDefaultProfileNum\_t::family

8.231.2.2 uint8\_t pack\_wds\_GetDefaultProfileNum\_t::type

## 8.232 pack\_wds\_GetDormancyState\_t Struct Reference

## 8.233 pack\_wds\_GetLastMobileIPError\_t Struct Reference

## 8.234 pack\_wds\_GetMobileIP\_t Struct Reference

## 8.235 pack\_wds\_GetMobileIPProfile\_t Struct Reference

#### Data Fields

- uint8\_t [index](#)



### 8.235.1 Detailed Description

#### Parameters

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

### 8.235.2 Field Documentation

8.235.2.1 uint8\_t pack\_wds\_GetMobileIPProfile\_t::index

## 8.236 pack\_wds\_GetPacketStatistics\_t Struct Reference

#### Data Fields

- uint32\_t \* [pStatMask](#)

### 8.236.1 Detailed Description

#### Parameters

<i>pStatMask</i>	<ul style="list-style-type: none"><li>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</li></ul>
------------------	--

### 8.236.2 Field Documentation

8.236.2.1 uint32\_t\* pack\_wds\_GetPacketStatistics\_t::pStatMask

## 8.237 pack\_wds\_GetPacketStatus\_t Struct Reference

#### Data Fields

- uint32\_t [statmask](#)

### 8.237.1 Detailed Description

#### Parameters

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

### 8.237.2 Field Documentation

8.237.2.1 uint32\_t pack\_wds\_GetPacketStatus\_t::statmask

## 8.238 pack\_wds\_GetSessionDuration\_t Struct Reference

## 8.239 pack\_wds\_RMSetTransferStatistics\_t Struct Reference

### Data Fields

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

### 8.239.1 Detailed Description

#### Parameters

<a href="#">rmTrasnfer↔ StaticsReq</a>	RM Transfer Statistics Indicator
--	----------------------------------

### 8.239.2 Field Documentation

8.239.2.1 [rmTrasnferStaticsReq](#) [pack\\_wds\\_RMSetTransferStatistics\\_t::RmTrasnferStaticsReq](#)

## 8.240 pack\_wds\_SetAutoconnect\_t Struct Reference

### Data Fields

- [uint8\\_t](#) [acsetting](#)
- [uint8\\_t](#) [acroamsetting](#)

### 8.240.1 Detailed Description

auto connect data session parameters.

#### Parameters

<i>setting</i>	<ul style="list-style-type: none"> <li>• NDIS autoconnect setting             <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> </ul>
----------------	---

### 8.240.2 Field Documentation

8.240.2.1 [uint8\\_t](#) [pack\\_wds\\_SetAutoconnect\\_t::acroamsetting](#)

8.240.2.2 [uint8\\_t](#) [pack\\_wds\\_SetAutoconnect\\_t::acsetting](#)

## 8.241 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](#)

### 8.241.1 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

### 8.241.2 Field Documentation

8.241.2.1 uint32\_t pack\_wds\_SetDefaultProfile\_t::authentication

8.241.2.2 uint32\_t pack\_wds\_SetDefaultProfile\_t::ipAddress

8.241.2.3 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pApnname

8.241.2.4 uint32\_t pack\_wds\_SetDefaultProfile\_t::pdpType

8.241.2.5 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pName

8.241.2.6 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pPassword

8.241.2.7 uint32\_t pack\_wds\_SetDefaultProfile\_t::primaryDNS

8.241.2.8 uint32\_t pack\_wds\_SetDefaultProfile\_t::profileType

8.241.2.9    `uint8_t*` `pack_wds_SetDefaultProfile_t::pUsername`

8.241.2.10   `uint32_t` `pack_wds_SetDefaultProfile_t::secondaryDNS`

## 8.242   `pack_wds_SetDefaultProfileNum_t` Struct Reference

### Data Fields

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

### 8.242.1   Field Documentation

8.242.1.1    `uint8_t` `pack_wds_SetDefaultProfileNum_t::family`

8.242.1.2    `uint8_t` `pack_wds_SetDefaultProfileNum_t::index`

8.242.1.3    `uint8_t` `pack_wds_SetDefaultProfileNum_t::type`

## 8.243   `pack_wds_SetMobileIP_t` Struct Reference

### Data Fields

- `uint32_t` [mode](#)

### 8.243.1   Detailed Description

#### Parameters

<i>mode</i>	<ul style="list-style-type: none"><li>• Mobile IP setting<ul style="list-style-type: none"><li>– 0 - Mobile IP off (simple IP only)</li><li>– 1 - Mobile IP preferred</li><li>– 2 - Mobile IP only</li></ul></li></ul>
-------------	--

### 8.243.2   Field Documentation

8.243.2.1    `uint32_t` `pack_wds_SetMobileIP_t::mode`

## 8.244   `pack_wds_SetMobileIPParameters_t` Struct Reference

### Data Fields

- `char *` [pSPC](#)

- uint32\_t \* [pMode](#)
- uint8\_t \* [pRetryLimit](#)
- uint8\_t \* [pRetryInterval](#)
- uint8\_t \* [pReRegPeriod](#)
- uint8\_t \* [pReRegTraffic](#)
- uint8\_t \* [pHAAuthenticator](#)
- uint8\_t \* [pHA2002bis](#)

### 8.244.1 Detailed Description

Mobile IP parameters information.

#### Parameters

<i>pSPC</i>	<ul style="list-style-type: none"> <li>• NULL-terminated string representing six digit service programming code.</li> </ul>
<i>pMode</i>	<ul style="list-style-type: none"> <li>• Mode to be set (optional) <ul style="list-style-type: none"> <li>– 0 - Mobile IP off (simple IP only)</li> <li>– 1 - Mobile IP preferred</li> <li>– 2 - Mobile IP only</li> </ul> </li> </ul>
<i>pRetryLimit</i>	<ul style="list-style-type: none"> <li>• Registration retry attempt limit (optional)</li> </ul>
<i>pRetryInterval</i>	<ul style="list-style-type: none"> <li>• Registration retry attempt interval used to determine the time between registration attempts (optional)</li> </ul>
<i>pReRegPeriod</i>	<ul style="list-style-type: none"> <li>• Period (in minutes) to attempt re-registration before current registration expires (optional)</li> </ul>
<i>pReRegTraffic</i>	<ul style="list-style-type: none"> <li>• Re-registration only if traffic since last attempt (optional) <ul style="list-style-type: none"> <li>– Zero - Disabled</li> <li>– NonZero - Enabled</li> </ul> </li> </ul>
<i>pHA<math>\leftrightarrow</math></i> <i>Authenticator</i>	<ul style="list-style-type: none"> <li>• MH-HA authenticator calculator (optional) <ul style="list-style-type: none"> <li>– Zero - Disabled</li> <li>– NonZero - Enabled</li> </ul> </li> </ul>
<i>pHA2002bis</i>	<ul style="list-style-type: none"> <li>• MH-HA RFC 2002bis authentication instead of RFC2002 (optional) <ul style="list-style-type: none"> <li>– Zero - Disabled</li> <li>– NonZero - Enabled</li> </ul> </li> </ul>

### 8.244.2 Field Documentation

- 8.244.2.1 `uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis`
- 8.244.2.2 `uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator`
- 8.244.2.3 `uint32_t* pack_wds_SetMobileIPParameters_t::pMode`
- 8.244.2.4 `uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod`
- 8.244.2.5 `uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic`
- 8.244.2.6 `uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval`
- 8.244.2.7 `uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit`
- 8.244.2.8 `char* pack_wds_SetMobileIPParameters_t::pSPC`

## 8.245 `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`

### 8.245.1 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

## 8.245.2 Field Documentation

8.245.2.1 `uint8_t pack_wds_SetMobileIPProfile_t::index`

8.245.2.2 `uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI`

8.245.2.3 `uint32_t* pack_wds_SetMobileIPProfile_t::pAddress`

8.245.2.4 `uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled`

8.245.2.5 `uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI`

8.245.2.6 `int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA`

8.245.2.7 `int8_t* pack_wds_SetMobileIPProfile_t::pMNHA`

8.245.2.8 `int8_t* pack_wds_SetMobileIPProfile_t::pNAI`

8.245.2.9 `uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA`

8.245.2.10 `uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling`

8.245.2.11 `uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA`

8.245.2.12 `int8_t pack_wds_SetMobileIPProfile_t::spc[10]`

## 8.246 pack\_wds\_SLQSCreateProfile\_t Struct Reference

### Data Fields

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

### 8.246.1 Detailed Description

#### Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> <li>• 1 to 16 for 3GPP profile (EM/MC73xx or earlier)</li> <li>• 1 to 24 for 3GPP profile (EM/MC74xx onwards)</li> <li>• 101 to 106 for 3GPP2 profile</li> </ul>
<i>ProfileType</i>	<ul style="list-style-type: none"> <li>• Identifies the technology type of the profile <ul style="list-style-type: none"> <li>– 0x00 - 3GPP</li> <li>– 0x01 - 3GPP2</li> </ul> </li> </ul>
Generated by Doxygen	<ul style="list-style-type: none"> <li>– NULL is not allowed</li> </ul>
<i>curProfile</i>	<ul style="list-style-type: none"> <li>• union of 3GPP and 3GPP2 profile</li> </ul>

## Note

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

## 8.246.2 Field Documentation

8.246.2.1 wds\_profileInfo\* pack\_wds\_SLQSCreateProfile\_t::pCurProfile

8.246.2.2 uint8\_t\* pack\_wds\_SLQSCreateProfile\_t::pProfileId

8.246.2.3 uint8\_t\* pack\_wds\_SLQSCreateProfile\_t::pProfileType

## 8.247 pack\_wds\_SLQSDeleteProfile\_t Struct Reference

### Data Fields

- uint8\_t [profileType](#)
- uint8\_t [profileIndex](#)

### 8.247.1 Detailed Description

#### Parameters

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

### 8.247.2 Field Documentation

8.247.2.1 uint8\_t pack\_wds\_SLQSDeleteProfile\_t::profileIndex

8.247.2.2 uint8\_t pack\_wds\_SLQSDeleteProfile\_t::profileType

## 8.248 pack\_wds\_SLQSGetCurrDataSystemStat\_t Struct Reference

## 8.249 pack\_wds\_SLQSGetDataBearerTechnology\_t Struct Reference

## 8.250 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](#)



## 8.250.1 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

## 8.250.2 Field Documentation

8.250.2.1 uint32\_t pack\_wds\_SLQSGetDUNCallInfo\_t::Mask

8.250.2.2 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportChannelRate

8.250.2.3 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportConnStatus

8.250.2.4 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportDataBearerTech

8.250.2.5 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportDormStatus

8.250.2.6 transferStatInd\* pack\_wds\_SLQSGetDUNCallInfo\_t::pTransferStatInd

## 8.251 pack\_wds\_SLQSGetProfileSettings\_t Struct Reference

## Data Fields

- uint8\_t [ProfileID](#)
- uint8\_t [ProfileType](#)

## 8.251.1 Detailed Description

## Parameters

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

**Note**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**8.251.2 Field Documentation**

8.251.2.1 uint8\_t pack\_wds\_SLQSGetProfileSettings\_t::ProfileId

8.251.2.2 uint8\_t pack\_wds\_SLQSGetProfileSettings\_t::ProfileType

**8.252 pack\_wds\_SLQSGetRuntimeSettings\_t Struct Reference****Data Fields**

- uint32\_t \* [pReqSettings](#)

**8.252.1 Detailed Description**

## Parameters

<i>pReqSettings</i>	<p>Requested Settings (Optional Parameter)</p> <ul style="list-style-type: none"> <li>• Set bits to 1, corresponding to requested information. All other bits must be set to 0.</li> <li>• If the values are not available, the corresponding TLVs are not returned in the response.</li> <li>• Absence of this mask TLV results in the device returning all of the available information corresponding to bits 0 through 12.</li> <li>• In cases where the information from bit 13 or greater is required, this TLV with all the necessary bits set must be present in the request.</li> <li>• Values <ul style="list-style-type: none"> <li>– Bit 0 - Profile identifier</li> <li>– Bit 1 - Profile name</li> <li>– Bit 2 - PDP type</li> <li>– Bit 3 - APN name</li> <li>– Bit 4 - DNS address</li> <li>– Bit 5 - UMTS/GPRS granted QoS</li> <li>– Bit 6 - Username</li> <li>– Bit 7 - Authentication Protocol</li> <li>– Bit 8 - IP address</li> <li>– Bit 9 - Gateway info (address and subnet mask)</li> <li>– Bit 10 - PCSCF address using PCO flag</li> <li>– Bit 11 - PCSCF server address list</li> <li>– Bit 12 - PCSCF domain name list</li> <li>– Bit 13 - MTU</li> <li>– Bit 14 - domain name list</li> <li>– Bit 15 - IP family</li> <li>– Bit 16 - IM_CM flag</li> <li>– Bit 17 - Technology name</li> <li>– Bit 18 - Operator reserved PCO</li> </ul> </li> </ul>
---------------------	--

## 8.252.2 Field Documentation

8.252.2.1 uint32\_t\* pack\_wds\_SLQSGetRuntimeSettings\_t::pReqSettings

## 8.253 pack\_wds\_SLQSModifyProfile\_t Struct Reference

## Data Fields

- uint8\_t \* [pProfileId](#)
- uint8\_t \* [pProfileType](#)
- [wds\\_profileInfo](#) curProfile

## 8.253.1 Detailed Description

## Parameters

<i>ProfileID</i>	<ul style="list-style-type: none"> <li>• 1 to 16 for 3GPP profile (EM/MC73xx or earlier)</li> <li>• 1 to 24 for 3GPP profile (EM/MC74xx onwards)</li> <li>• 101 to 106 for 3GPP2 profile</li> </ul>
<i>ProfileType</i>	<ul style="list-style-type: none"> <li>• Identifies the technology type of the profile <ul style="list-style-type: none"> <li>– 0x00 - 3GPP</li> <li>– 0x01 - 3GPP2</li> <li>– NULL is not allowed</li> </ul> </li> </ul>
<i>curProfile</i>	<ul style="list-style-type: none"> <li>• union of 3GPP and 3GPP2 profile</li> </ul>

## Note

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

## 8.253.2 Field Documentation

8.253.2.1 wds\_profileInfo pack\_wds\_SLQSMModifyProfile\_t::curProfile

8.253.2.2 uint8\_t\* pack\_wds\_SLQSMModifyProfile\_t::pProfileId

8.253.2.3 uint8\_t\* pack\_wds\_SLQSMModifyProfile\_t::pProfileType

## 8.254 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](#)

## 8.254.1 Detailed Description

## Parameters

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

## 8.254.2 Field Documentation

## 8.254.2.1 uint16\_t pack\_wds\_SLQSSet3GPPConfigItem\_t::LTEAttachProfileListLen

8.254.2.2 `uint8_t*` `pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease`

8.254.2.3 `uint8_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled`

8.254.2.4 `uint16_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile`

8.254.2.5 `uint16_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList`

8.254.2.6 `uint16_t*` `pack_wds_SLQSSet3GPPConfigItem_t::pProfileList`

## 8.255 `pack_wds_SLQSSetIPFamilyPreference_t` Struct Reference

### Data Fields

- `uint8_t` [IPFamilyPreference](#)

### 8.255.1 Detailed Description

#### Parameters

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

### 8.255.2 Field Documentation

8.255.2.1 `uint8_t` `pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference`

## 8.256 `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](#)

### 8.256.1 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

## 8.256.2 Field Documentation

8.256.2.1 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::currentDataBearer

8.256.2.2 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataBearer

8.256.2.3 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataSystemStatus

8.256.2.4 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dormancyStatus

8.256.2.5 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::interval

8.256.2.6 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::mobileIP

8.256.2.7 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::transferStats

## 8.257 pack\_wds\_SLQSSetDHCPv4ClientConfig\_t Struct Reference

## Data Fields

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

## 8.257.1 Detailed Description

## Parameters

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

## 8.257.2 Field Documentation

8.257.2.1 wdsDhcpv4ProfileId\* pack\_wds\_SLQSSetDHCPv4ClientConfig\_t::pProfileId

## 8.258 pack\_wds\_SLQSSetDHCPv4ClientConfig\_t Struct Reference

## Data Fields

- [wds\\_DHCPv4ProfileId](#) \* [pProfileId](#)
- [wds\\_DHCPv4HWConfig](#) \* [pHwConfig](#)
- [wds\\_DHCPv4OptionList](#) \* [pRequestOptionList](#)

### 8.258.1 Detailed Description

WDS SWI DHCPv4 Config Structure

#### Parameters

<i>pProfileId</i>	<ul style="list-style-type: none"> <li>• pointer to Profile Id structure</li> </ul>
<i>pHWConfig</i>	<ul style="list-style-type: none"> <li>• pointer to HW Config structure</li> </ul>
<i>pRequest↔ OptionList</i>	<ul style="list-style-type: none"> <li>• pointer to Option List structure to be sent in DHCP request</li> </ul>

### 8.258.2 Field Documentation

8.258.2.1 [wds\\_DHCPv4HWConfig](#)\* [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t::pHwConfig](#)

8.258.2.2 [wds\\_DHCPv4ProfileId](#)\* [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t::pProfileId](#)

8.258.2.3 [wds\\_DHCPv4OptionList](#)\* [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t::pRequestOptionList](#)

## 8.259 [pack\\_wds\\_SLQSSSetLoopback\\_t](#) Struct Reference

## Data Fields

- [uint8\\_t](#) [loopbackMode](#)
- [uint8\\_t](#) [loopbackMultiplier](#)

### 8.259.1 Detailed Description

#### Parameters

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



## 8.259.2 Field Documentation

8.259.2.1 uint8\_t pack\_wds\_SLQSSSetLoopback\_t::loopbackMode

8.259.2.2 uint8\_t pack\_wds\_SLQSSSetLoopback\_t::loopbackMultiplier

## 8.260 pack\_wds\_SLQSSStartDataSession\_t Struct Reference

### Data Fields

- uint8\_t \* [pTech](#)
- uint32\_t \* [pprofileid3gpp](#)
- uint32\_t \* [pprofileid3gpp2](#)
- uint32\_t \* [pAuth](#)
- char \* [pUser](#)
- char \* [pPass](#)

### 8.260.1 Detailed Description

#### Parameters

<i>pTech</i>	<ul style="list-style-type: none"> <li>• Indicates the technology preference               <ul style="list-style-type: none"> <li>– 1 - UMTS</li> <li>– 2 - CDMA</li> <li>– 3 - eMBMS</li> <li>– 4 - Modem Link Label. Modem Link is an interface for transferring data between entities on AP and modem.</li> </ul> </li> <li>• optional</li> </ul>
<i>pprofileid3gpp</i>	<ul style="list-style-type: none"> <li>• pointer to 3GPP profile id</li> <li>• optional</li> </ul>
<i>pprofileid3gpp2</i>	<ul style="list-style-type: none"> <li>• pointer to 3GPPs profile id</li> <li>• optional</li> </ul>
<i>pAuth</i>	<ul style="list-style-type: none"> <li>• Authentication type, it can be PAP or CHAP</li> <li>• optional</li> </ul>
<i>pUser</i>	<ul style="list-style-type: none"> <li>• username for authentication process</li> <li>• optional</li> </ul>
<i>pPass</i>	<ul style="list-style-type: none"> <li>• password for authentication process</li> <li>• optional</li> </ul>

## 8.260.2 Field Documentation

8.260.2.1 `uint32_t*` `pack_wds_SLQSSStartDataSession_t::pAuth`

8.260.2.2 `char*` `pack_wds_SLQSSStartDataSession_t::pPass`

8.260.2.3 `uint32_t*` `pack_wds_SLQSSStartDataSession_t::pprofileid3gpp`

8.260.2.4 `uint32_t*` `pack_wds_SLQSSStartDataSession_t::pprofileid3gpp2`

8.260.2.5 `uint8_t*` `pack_wds_SLQSSStartDataSession_t::pTech`

8.260.2.6 `char*` `pack_wds_SLQSSStartDataSession_t::pUser`

## 8.261 `pack_wds_SLQSSStopDataSession_t` Struct Reference

### Data Fields

- `uint32_t *` `psid`

### 8.261.1 Detailed Description

#### Parameters

<code>sid</code>	session id
------------------	------------

## 8.261.2 Field Documentation

8.261.2.1 `uint32_t*` `pack_wds_SLQSSStopDataSession_t::psid`

## 8.262 `pack_wds_SLQSWdsSetEventReport_t` Struct Reference

### Data Fields

- `uint8_t *` `pCurrChannelRateInd`
- `wds_TrStatInd *` `pTransferStatInd`
- `uint8_t *` `pDataBearerTechInd`
- `uint8_t *` `pDormancyStatusInd`
- `uint8_t *` `pMIPStatusInd`
- `uint8_t *` `pCurrDataBearerTechInd`
- `uint8_t *` `pDataCallStatusChangeInd`
- `uint8_t *` `pCurrPrefDataSysInd`
- `uint8_t *` `pEVDOPageMonPerChangeInd`
- `uint8_t *` `pDataSystemStatusChangeInd`

## 8.262.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

## Parameters

<i>pCurrChannel↔ RateInd</i>	(optional) <ul style="list-style-type: none"> <li>Current Channel Rate Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report channel rate when it changes</li> </ul> </li> </ul>
<i>pTransferStatInd</i>	(optional) <ul style="list-style-type: none"> <li>See TrStatInd for more information.</li> </ul>
<i>pDataBearer↔ TechInd</i>	(optional) <ul style="list-style-type: none"> <li>Data Bearer Technology Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report radio interface used for data transfer when it changes</li> </ul> </li> </ul>
<i>pDormancy↔ StatusInd</i>	(optional) <ul style="list-style-type: none"> <li>Dormancy Status indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report traffic channel state of interface used for data connection</li> </ul> </li> </ul>
<i>pMIPStatusInd</i>	(optional) <ul style="list-style-type: none"> <li>MIP Status Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report MIP status</li> </ul> </li> </ul>
<i>pCurrData↔ BearerTechInd</i>	(optional) <ul style="list-style-type: none"> <li>Current Data Bearer Technology Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report current data bearer technology when it changes</li> </ul> </li> </ul>
<i>pDataCall↔ StatusChange↔ Ind</i>	(optional) <ul style="list-style-type: none"> <li>Data Call Status Change Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report data call status change when it changes</li> </ul> </li> </ul>
<i>pCurrPrefData↔ SysInd</i>	(optional) <ul style="list-style-type: none"> <li>Current Preferred Data System Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report preferred data system when it changes</li> </ul> </li> </ul>
<i>pEVDOPage↔ MonPer↔ ChangeInd</i>	(optional) <ul style="list-style-type: none"> <li>EV-DO Page Monitor Period Change Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report EV-DO page monitor period change event</li> </ul> </li> </ul>
<i>pDataSystem↔ StatusChange↔ Ind</i>	(optional) <ul style="list-style-type: none"> <li>Data System Status Change Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report data system status change event</li> </ul> </li> </ul>

**Note**

At least one parameter should be present.

**8.262.2 Field Documentation**

8.262.2.1 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd`

8.262.2.2 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd`

8.262.2.3 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd`

8.262.2.4 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd`

8.262.2.5 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd`

8.262.2.6 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd`

8.262.2.7 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd`

8.262.2.8 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd`

8.262.2.9 `uint8_t*` `pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd`

8.262.2.10 `wds_TrStatInd*` `pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd`

**8.263 `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t` Struct Reference****Data Fields**

- `uint8_t` `contextId`
- `uint8_t` `contextType`

**8.263.1 Detailed Description****Parameters**

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

**8.263.2 Field Documentation**

8.263.2.1 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.263.2.2 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType`

## 8.264 PackCreateProfileOut Struct Reference

### Data Fields

- uint8\_t [ProfileType](#)
- uint8\_t [ProfileIndex](#)
- uint16\_t [ExtErrorCode](#)

### 8.264.1 Field Documentation

8.264.1.1 uint16\_t PackCreateProfileOut::ExtErrorCode

8.264.1.2 uint8\_t PackCreateProfileOut::ProfileIndex

8.264.1.3 uint8\_t PackCreateProfileOut::ProfileType

## 8.265 packgetDyingGaspCfg Struct Reference

### Data Fields

- uint8\_t \* [pDestSMSNum](#)
- uint8\_t \* [pDestSMSContent](#)

### 8.265.1 Detailed Description

#### Parameters

<i>pDestSMS↔ Num[IN]</i>	<ul style="list-style-type: none"> <li>• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>• Optional parameter.</li> </ul>
<i>pDestSMS↔ Content[IN]</i>	<ul style="list-style-type: none"> <li>• SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>• Optional parameter.</li> </ul>

### 8.265.2 Field Documentation

8.265.2.1 uint8\_t\* packgetDyingGaspCfg::pDestSMSContent

8.265.2.2 uint8\_t\* packgetDyingGaspCfg::pDestSMSNum

## 8.266 packgetDyingGaspStatistics Struct Reference

### Data Fields

- uint32\_t \* [pTimeStamp](#)
- uint8\_t \* [pSMSAttemptedFlag](#)

### 8.266.1 Detailed Description

#### Parameters

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

### 8.266.2 Field Documentation

8.266.2.1 uint8\_t\* packetDyingGaspStatistics::pSMSAttemptedFlag

8.266.2.2 uint32\_t\* packetDyingGaspStatistics::pTimeStamp

## 8.267 qmiSmsMessageList Struct Reference

#### Data Fields

- uint32\_t [messageIndex](#)
- uint32\_t [messageTag](#)

### 8.267.1 Detailed Description

#### Parameters

<i>messageIndex</i>	<ul style="list-style-type: none"> <li>Message index of each matched message</li> </ul>
<i>messageTag</i>	<ul style="list-style-type: none"> <li>Message tag</li> </ul>

### 8.267.2 Field Documentation

8.267.2.1 uint32\_t qmiSmsMessageList::messageIndex

8.267.2.2 uint32\_t qmiSmsMessageList::messageTag

## 8.268 qmiWSDDataBearerTechnology Struct Reference

#### Data Fields

- uint8\_t [currentNetwork](#)
- uint32\_t [ratMask](#)
- uint32\_t [soMask](#)

### 8.268.1 Detailed Description

#### Parameters

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

### 8.268.2 Field Documentation

8.268.2.1 `uint8_t qmiWSDDataBearerTechnology::currentNetwork`

8.268.2.2 `uint32_t qmiWSDDataBearerTechnology::ratMask`

8.268.2.3 `uint32_t qmiWSDDataBearerTechnology::soMask`

## 8.269 RFBandInfoElements Struct Reference

#### Data Fields

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

### 8.269.1 Detailed Description

#### Parameters

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

### 8.269.2 Field Documentation

8.269.2.1 `uint16_t RFBandInfoElements::activeBandClass`

8.269.2.2 `uint16_t RFBandInfoElements::activeChannel`

8.269.2.3 `uint8_t RFBandInfoElements::radioInterface`

## 8.270 rmTrasnferStaticsReq Struct Reference

#### Data Fields

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

### 8.270.1 Detailed Description

#### Parameters

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

### 8.270.2 Field Documentation

8.270.2.1 `uint8_t rmTrasnferStaticsReq::bResetStatistics`

8.270.2.2 `uint32_t rmTrasnferStaticsReq::ulMask`

## 8.271 `sensorData_t` Struct Reference

### Data Fields

- `uint32_t timeOfFirstSample`
- `uint8_t flags`
- `uint8_t sensorDataLen`
- `uint16_t timeOffset` [64]
- `uint32_t xAxis` [64]
- `uint32_t yAxis` [64]
- `uint32_t zAxis` [64]

### 8.271.1 Detailed Description

This structure specifies information regarding the 3-Axis Sensor Data. Please check has\_<Param\_Name> field for presence of optional parameters

#### Parameters

<i>timeOfFirstSample</i>	<ul style="list-style-type: none"> <li>• Denotes a full 32-bit time stamp of the first (oldest) sample in this message.</li> <li>• The time stamp is in the time reference scale that is used by the sensor time source.</li> <li>• Units - Milliseconds</li> </ul>
<i>flags</i>	<ul style="list-style-type: none"> <li>• Flags to indicate any deviation from the default measurement assumptions.</li> <li>• All unused bits in this field must be set to 0.</li> <li>• Valid bitmasks <ul style="list-style-type: none"> <li>– 0x01 - Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples</li> <li>– 0x02 - Bitmask to specify that the sensor time stamp is the same as the modem time stamp</li> </ul> </li> </ul>
<i>sensorDataLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– timeOffset</li> <li>– xAxis</li> <li>– yAxis</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– zAxis</li> </ul>



<i>timeOffset</i>	<ul style="list-style-type: none"> <li>• Sample time offset</li> <li>• Units - Milliseconds</li> </ul>
<i>xAxis</i>	<ul style="list-style-type: none"> <li>• Sensor x-axis sample.</li> <li>• Units Accelerometer - Meters/seconds square</li> <li>• Units Gyroscope - Radians/second</li> </ul>
<i>yAxis</i>	<ul style="list-style-type: none"> <li>• Sensor Y-axis sample.</li> <li>• Units Accelerometer - Meters/seconds square</li> <li>• Units Gyroscope - Radians/second</li> </ul>
<i>zAxis</i>	<ul style="list-style-type: none"> <li>• Sensor Z-axis sample.</li> <li>• Units Accelerometer - Meters/seconds square</li> <li>• Units Gyroscope - Radians/second</li> </ul>

## 8.271.2 Field Documentation

8.271.2.1 `uint8_t sensorData_t::flags`

8.271.2.2 `uint8_t sensorData_t::sensorDataLen`

8.271.2.3 `uint32_t sensorData_t::timeOfFirstSample`

8.271.2.4 `uint16_t sensorData_t::timeOffset[64]`

8.271.2.5 `uint32_t sensorData_t::xAxis[64]`

8.271.2.6 `uint32_t sensorData_t::yAxis[64]`

8.271.2.7 `uint32_t sensorData_t::zAxis[64]`

## 8.272 slot\_t Struct Reference

### Data Fields

- `uint32_t uPhyCardStatus`
- `uint32_t uPhySlotStatus`
- `uint8_t bLogicalSlot`
- `uint8_t bICCIDLength`
- `uint8_t bICCID [255]`

### 8.272.1 Detailed Description

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

#### Parameters

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

## 8.272.2 Field Documentation

8.272.2.1 `uint8_t slot_t::bICCID[255]`

8.272.2.2 `uint8_t slot_t::bICCIDLength`

8.272.2.3 `uint8_t slot_t::bLogicalSlot`

8.272.2.4 `uint32_t slot_t::uPhyCardStatus`

8.272.2.5 `uint32_t slot_t::uPhySlotStatus`

## 8.273 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]`

## 8.273.1 Detailed Description

This structure contains information about the SLOTS present.

## Parameters

<i>cardState</i>	<ul style="list-style-type: none"> <li>Indicates the state of the card for each slot. <ul style="list-style-type: none"> <li>0 - Absent</li> <li>1 - Present</li> <li>2 - Error</li> </ul> </li> </ul>
<i>upinState</i>	<ul style="list-style-type: none"> <li>Indicates the state of UPIN. <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Enabled and not verified</li> <li>2 - Enabled and verified</li> <li>3 - Disabled</li> <li>4 - Blocked</li> <li>5 - Permanently blocked</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>upinRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to verify the UPIN.</li> <li>If 0xFF, information not available.</li> </ul>
<i>upukRetries</i>	<ul style="list-style-type: none"> <li>Indicates the number of retries remaining to unblock the UPIN.</li> <li>If 0xFF, information not available.</li> </ul>
<i>errorState</i>	<ul style="list-style-type: none"> <li>Indicates the reason for the card error, and is valid only when the card state is Error <ul style="list-style-type: none"> <li>0 - Unknown</li> <li>1 - Power down</li> <li>2 - Poll error</li> <li>3 - No ATR received</li> <li>4 - Volt mismatch</li> <li>5 - Parity error</li> <li>6 - Unknown; possibly removed</li> <li>7 - Card returned technical problems</li> <li>0xFF - Not Available</li> </ul> </li> <li>Other values are possible and reserved for future use.</li> <li>When an unknown value is received, it is to be handled as "Unknown".</li> </ul>
<i>numApp</i>	<ul style="list-style-type: none"> <li>Indicates the number of applications available on the card.</li> <li>The following block is repeated for each application. i.e. AppStatus.</li> <li>If zero(0) then no AppStatus information exists.</li> </ul>
<i>AppStatus[MA↔ X_NO_OF_AP↔ PLICATIONS]</i>	<ul style="list-style-type: none"> <li>See <a href="#">appStats</a> for more information.</li> </ul>

### 8.273.2 Field Documentation

8.273.2.1 appStats slotInf::AppStatus[10]

8.273.2.2 uint8\_t slotInf::cardState

8.273.2.3 uint8\_t slotInf::errorState

8.273.2.4 uint8\_t slotInf::numApp

8.273.2.5 uint8\_t slotInf::upinRetries

8.273.2.6 uint8\_t slotInf::upinState

8.273.2.7 uint8\_t slotInf::upukRetries

## 8.274 slots\_t Struct Reference

### Data Fields

- [slot\\_t uimSlotStatus](#) [255]

### 8.274.1 Field Documentation

8.274.1.1 slot\_t slots\_t::uimSlotStatus[255]

## 8.275 sMSCAddress Struct Reference

### Data Fields

- uint8\_t [length](#)
- uint8\_t [data](#) [256]

### 8.275.1 Detailed Description

#### Parameters

<i>length</i>	<ul style="list-style-type: none"><li>• Number of sets of following element</li></ul>
<i>data</i>	<ul style="list-style-type: none"><li>• SMSC address</li></ul>

### 8.275.2 Field Documentation

8.275.2.1 `uint8_t sMSCAddress::data[256]`

8.275.2.2 `uint8_t sMSCAddress::length`

## 8.276 sMSCAddressTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `sMSCAddressInfo SMSCInfo`

### 8.276.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Boolean indicating the presence of the TLV in the QMI response</li></ul>
<i>SMSCInfo</i>	<ul style="list-style-type: none"><li>• SMSC Address</li><li>• See <a href="#">sMSCAddressInfo</a> for more information</li></ul>

### 8.276.2 Field Documentation

8.276.2.1 `sMSCAddressInfo sMSCAddressTlv::SMSCInfo`

8.276.2.2 `uint8_t sMSCAddressTlv::TlvPresent`

## 8.277 sMSEtwsMessage Struct Reference

### Data Fields

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

### 8.277.1 Detailed Description

## Parameters

<i>notificationType</i>	<ul style="list-style-type: none"> <li>• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS</li> </ul>
<i>length</i>	<ul style="list-style-type: none"> <li>• Number of sets of following elements</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>• Raw message data</li> </ul>

## 8.277.2 Field Documentation

8.277.2.1 `uint8_t sMSEtwsMessage::data[1254]`8.277.2.2 `uint16_t sMSEtwsMessage::length`8.277.2.3 `uint8_t sMSEtwsMessage::notificationType`

## 8.278 sMSEtwsMessageTlv Struct Reference

## Data Fields

- `uint8_t TlvPresent`
- `sMSEtwsMessageInfo EtwsMessageInfo`

## 8.278.1 Detailed Description

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>EtwsMessageInfo</i>	<ul style="list-style-type: none"> <li>• ETWS Message</li> <li>• See <a href="#">sMSEtwsMessageInfo</a> for more information</li> </ul>

## 8.278.2 Field Documentation

8.278.2.1 `sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo`8.278.2.2 `uint8_t sMSEtwsMessageTlv::TlvPresent`

## 8.279 sMSEtwsPlmn Struct Reference

## Data Fields

- uint16\_t [mobileCountryCode](#)
- uint16\_t [mobileNetworkCode](#)

### 8.279.1 Detailed Description

#### Parameters

<i>mobileCountry↔ Code</i>	<ul style="list-style-type: none"><li>• 16 bit representation of MCC value range : 0 -999</li></ul>
<i>mobile↔ NetworkCode</i>	<ul style="list-style-type: none"><li>• 16 bit representation of MNC value range : 0 -999</li></ul>

### 8.279.2 Field Documentation

8.279.2.1 uint16\_t sMSEtwsPlmn::mobileCountryCode

8.279.2.2 uint16\_t sMSEtwsPlmn::mobileNetworkCode

## 8.280 sMSMessageMode Struct Reference

## Data Fields

- uint8\_t [messageMode](#)

### 8.280.1 Detailed Description

#### Parameters

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

### 8.280.2 Field Documentation

8.280.2.1 uint8\_t sMSMessageMode::messageMode

## 8.281 sMSMTMessage Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)

### 8.281.1 Detailed Description

#### Parameters

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

### 8.281.2 Field Documentation

8.281.2.1 `uint32_t sMSMTMessage::messageIndex`

8.281.2.2 `uint32_t sMSMTMessage::storageType`

## 8.282 sMSOnIMS Struct Reference

### Data Fields

- `uint8_t` [sMSOnIMS](#)

### 8.282.1 Detailed Description

#### Parameters

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

### 8.282.2 Field Documentation

8.282.2.1 `uint8_t sMSOnIMS::sMSOnIMS`

## 8.283 sMSOnIMSTlv Struct Reference

### Data Fields

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

### 8.283.1 Detailed Description

#### Parameters

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



### 8.283.2 Field Documentation

8.283.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

8.283.2.2 uint8\_t sMSOnIMSTlv::TlvPresent

## 8.284 sMSTransferRouteMTMessage Struct Reference

### Data Fields

- uint8\_t [ackIndicator](#)
- uint32\_t [transactionID](#)
- uint8\_t [format](#)
- uint16\_t [length](#)
- uint8\_t [data](#) [256]

### 8.284.1 Detailed Description

#### Parameters

<i>ackIndicator</i>	<ul style="list-style-type: none"><li>• Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK</li></ul>
<i>transactionID</i>	<ul style="list-style-type: none"><li>• Transaction ID of the message</li></ul>
<i>format</i>	<ul style="list-style-type: none"><li>• Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC</li></ul>
<i>length</i>	<ul style="list-style-type: none"><li>• Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes</li></ul>
<i>data</i>	<ul style="list-style-type: none"><li>• Raw message data</li></ul>

### 8.284.2 Field Documentation

8.284.2.1 uint8\_t sMSTransferRouteMTMessage::ackIndicator

8.284.2.2 uint8\_t sMSTransferRouteMTMessage::data[256]

8.284.2.3 uint8\_t sMSTransferRouteMTMessage::format

8.284.2.4 uint16\_t sMSTransferRouteMTMessage::length

8.284.2.5 `uint32_t sMSTransferRouteMTMessage::transactionID`

## 8.285 `tdscdmaSigInfoExt` Struct Reference

### Data Fields

- float `rsi`
- float `rscp`
- float `ecio`
- float `sinr`

### 8.285.1 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.

### 8.285.2 Field Documentation

8.285.2.1 `float tdscdmaSigInfoExt::ecio`

8.285.2.2 `float tdscdmaSigInfoExt::rscp`

8.285.2.3 `float tdscdmaSigInfoExt::rsi`

8.285.2.4 `float tdscdmaSigInfoExt::sinr`

## 8.286 `tempData_t` Struct Reference

### Data Fields

- `uint32_t timeSource`
- `uint32_t timeOfFirstSample`
- `uint8_t temperatureDataLen`
- `uint16_t timeOffset` [64]
- `uint32_t temperature` [64]

### 8.286.1 Detailed Description

This structure specifies information regarding the Temperature Data. Please check `has_<Param_Name>` field for presence of optional parameters

#### Parameters

<i>timeSource</i>	<ul style="list-style-type: none"> <li>• Time source of the sensor data</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Sensor time source is unspecified</li> <li>– 1 - Time source is common between the sensors and the location engine</li> </ul> </li> </ul>
<i>timeOfFirstSample</i>	<ul style="list-style-type: none"> <li>• Denotes a full 32-bit time stamp of the first (oldest) sample in this message.</li> <li>• The time stamp is in the time reference scale that is used by the sensor time source.</li> <li>• Units - Milliseconds</li> </ul>
<i>temperatureDataLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– timeOffset</li> <li>– temperature</li> </ul> </li> </ul>
<i>timeOffset</i>	<ul style="list-style-type: none"> <li>• Sample time offset</li> <li>• Units - Milliseconds</li> </ul>
<i>temperature</i>	<ul style="list-style-type: none"> <li>• Sensor temperature.</li> <li>• Type - Floating point</li> <li>• Units - Degrees Celsius</li> <li>• Range -50 to +100.00</li> </ul>

## 8.286.2 Field Documentation

8.286.2.1 `uint32_t tempData_t::temperature[64]`

8.286.2.2 `uint8_t tempData_t::temperatureDataLen`

8.286.2.3 `uint32_t tempData_t::timeOfFirstSample`

8.286.2.4 `uint16_t tempData_t::timeOffset[64]`

8.286.2.5 `uint32_t tempData_t::timeSource`

## 8.287 transferRouteMessageTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo`

### 8.287.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>TransferRouteMTMessageInfo</i>	<ul style="list-style-type: none"> <li>• Transfer Route MT Message</li> <li>• See <a href="#">sMSTransferRouteMTMessageInfo</a> for more information</li> </ul>

## 8.287.2 Field Documentation

8.287.2.1 `uint8_t transferRouteMessageTlv::TlvPresent`

8.287.2.2 `sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo`

## 8.288 transferStatInd Struct Reference

### Data Fields

- `uint8_t StatsPeriod`
- `uint32_t StatsMask`

### 8.288.1 Detailed Description

#### Parameters

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

## 8.288.2 Field Documentation

8.288.2.1 `uint32_t transferStatInd::StatsMask`

8.288.2.2 `uint8_t transferStatInd::StatsPeriod`

## 8.289 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](#)

### 8.289.1 Detailed Description

This structure contains Application Status Information loaded on the card.

#### Parameters

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

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

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

### 8.289.2 Field Documentation

8.289.2.1 `uint8_t uim_appStatus::aidLength`

8.289.2.2 `uint8_t uim_appStatus::aidVal[255]`

8.289.2.3 `uint8_t uim_appStatus::appState`

8.289.2.4 `uint8_t uim_appStatus::appType`

8.289.2.5 `uint8_t uim_appStatus::persoFeature`

8.289.2.6 `uint8_t uim_appStatus::persoRetries`

8.289.2.7 `uint8_t uim_appStatus::persoState`

8.289.2.8 `uint8_t uim_appStatus::persoUnblockRetries`

8.289.2.9 `uint8_t uim_appStatus::pin1Retries`

8.289.2.10 `uint8_t uim_appStatus::pin1State`

8.289.2.11 `uint8_t uim_appStatus::pin2Retries`

8.289.2.12 `uint8_t uim_appStatus::pin2State`

8.289.2.13 `uint8_t uim_appStatus::puk1Retries`

8.289.2.14 `uint8_t uim_appStatus::puk2Retries`

8.289.2.15 `uint8_t uim_appStatus::univPin`

## 8.290 uim\_cardResult Struct Reference

### Data Fields

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

### 8.290.1 Detailed Description

This structure contains the information about the card result.

#### Parameters

<i>sw1</i>	<ul style="list-style-type: none"> <li>• SW1 received from the card.</li> </ul>
<i>sw2</i>	<ul style="list-style-type: none"> <li>• SW2 received from the card.</li> </ul>

### 8.290.2 Field Documentation

8.290.2.1 `uint8_t uim_cardResult::sw1`

8.290.2.2 `uint8_t uim_cardResult::sw2`

## 8.291 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]

### 8.291.1 Detailed Description

This structure contains Card Status Information.

#### Parameters

<i>indexGwPri</i>	<ul style="list-style-type: none"> <li>• Index of the primary GW provisioning application.</li> <li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li> <li>• The value 0xFFFF identifies when the session does not exist.</li> </ul>
<i>index1xPri</i>	<ul style="list-style-type: none"> <li>• Index of the primary 1X provisioning application.</li> <li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li> <li>• The value 0xFFFF identifies when the session does not exist.</li> </ul>



<i>indexGwSec</i>	<ul style="list-style-type: none"> <li>• Index of the secondary GW provisioning application.</li> <li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li> <li>• The value 0xFFFF identifies when the session does not exist.</li> </ul>
<i>index1xSec</i>	<ul style="list-style-type: none"> <li>• Index of the secondary GW provisioning application.</li> <li>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).</li> <li>• The value 0xFFFF identifies when the session does not exist.</li> </ul>
<i>numSlot</i>	<ul style="list-style-type: none"> <li>• Indicates the number of slots available on the device.</li> <li>• The following block is repeated for each slot. i.e. cardState</li> <li>• If zero(0) then no cardState information exists.</li> </ul>
<i>SlotInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_slotInfo</a> for more information.</li> </ul>

## 8.291.2 Field Documentation

8.291.2.1 `uint16_t uim_cardStatus::index1xPri`

8.291.2.2 `uint16_t uim_cardStatus::index1xSec`

8.291.2.3 `uint16_t uim_cardStatus::indexGwPri`

8.291.2.4 `uint16_t uim_cardStatus::indexGwSec`

8.291.2.5 `uint8_t uim_cardStatus::numSlot`

8.291.2.6 `uim_slotInfo uim_cardStatus::SlotInfo[5]`

## 8.292 uim\_changeUIMPIN Struct Reference

### Data Fields

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

### 8.292.1 Detailed Description

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

#### Parameters

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

## 8.292.2 Field Documentation

8.292.2.1 `uint8_t uim_changeUIMPIN::oldPINLen`

8.292.2.2 `uint8_t uim_changeUIMPIN::oldPINVal[255]`

8.292.2.3 `uint8_t uim_changeUIMPIN::pinID`

8.292.2.4 `uint8_t uim_changeUIMPIN::pinLen`

8.292.2.5 `uint8_t uim_changeUIMPIN::pinVal[255]`

## 8.293 uim\_encryptedPIN1 Struct Reference

### Data Fields

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

### 8.293.1 Detailed Description

This structure contains the encrypted PIN1 Information.

### Parameters

<i>pin1Len</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements ie encrypted PIN1 value.</li><li>• If zero(0), no information follows.</li></ul>
<i>pin1Val</i>	<ul style="list-style-type: none"><li>• Encrypted PIN1 value.</li></ul>

**Note**

This value is returned only when PIN1 is enabled successfully and the feature is supported.

**8.293.2 Field Documentation**

8.293.2.1 `uint8_t uim_encryptedPIN1::pin1Len`

8.293.2.2 `uint8_t uim_encryptedPIN1::pin1Val[255]`

**8.294 uim\_fileInfo Struct Reference****Data Fields**

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

**8.294.1 Detailed Description**

This structure contains paramaters for file Information

**Parameters**

<i>fileID</i>	<ul style="list-style-type: none"><li>• This is Identifier to SIM files; e.g. in UIM "6F07" is Identifier of IMSI File</li></ul>
<i>pathLen</i>	<ul style="list-style-type: none"><li>• Length of file Path</li></ul>
<i>path</i>	<ul style="list-style-type: none"><li>• Path value. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF).</li></ul>

**8.294.2 Field Documentation**

8.294.2.1 `uint16_t uim_fileInfo::fileID`

8.294.2.2 `uint16_t uim_fileInfo::path[255]`

8.294.2.3 `uint8_t uim_fileInfo::pathLen`

## 8.295 `uim_hotSwapStatus` Struct Reference

### Data Fields

- `uint8_t hotSwapLength`
- `uint8_t hotSwap [255]`

### 8.295.1 Detailed Description

This structure contains Hot Swap Status Information.

#### Parameters

<i>hotSwapLength</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements. i.e. <code>hot_swap</code></li> </ul>
<i>hotSwap</i>	<ul style="list-style-type: none"> <li>• Indicates the status of the hot-swap switch. <ul style="list-style-type: none"> <li>– 0 - Hot-swap is not supported</li> <li>– 1 - Hot-swap is supported, but the status of the switch is not supported</li> <li>– 2 - Switch indicates that the card is present</li> <li>– 3 - Switch indicates that the card is not present</li> </ul> </li> </ul>

### 8.295.2 Field Documentation

8.295.2.1 `uint8_t uim_hotSwapStatus::hotSwap[255]`

8.295.2.2 `uint8_t uim_hotSwapStatus::hotSwapLength`

## 8.296 `uim_readResult` Struct Reference

### Data Fields

- `uint16_t contentLen`
- `uint8_t content [255]`

### 8.296.1 Detailed Description

This structure contains the information for write operation.

#### Parameters

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

## 8.296.2 Field Documentation

8.296.2.1 `uint8_t uim_readResult::content[255]`

8.296.2.2 `uint16_t uim_readResult::contentLen`

## 8.297 uim\_readTransparentInfo Struct Reference

### Data Fields

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

### 8.297.1 Detailed Description

This structure contains the information for read operation.

#### Parameters

<i>offset</i>	<ul style="list-style-type: none"> <li>Offset for the read operation.</li> </ul>
<i>length</i>	<ul style="list-style-type: none"> <li>Length of the content to be read.</li> <li>The value 0 is used to read the complete file.</li> </ul>

### 8.297.2 Field Documentation

8.297.2.1 `uint16_t uim_readTransparentInfo::length`

8.297.2.2 `uint16_t uim_readTransparentInfo::offset`

## 8.298 uim\_remainingRetries Struct Reference

### Data Fields

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

### 8.298.1 Detailed Description

This structure contains the information about the retries remaining.

#### Parameters

<i>verifyLeft</i>	<ul style="list-style-type: none"> <li>• Number of remaining attempts to verify the PIN.</li> <li>• 0xFF, if unavailable.</li> </ul>
<i>unlockLeft</i>	<ul style="list-style-type: none"> <li>• Number of remaining attempts to unlock the PIN.</li> <li>• 0xFF, if unavailable.</li> </ul>

#### Note

This value is returned only when the enable/disable operation has failed. This information is not sent for a hidden key PIN type.

### 8.298.2 Field Documentation

8.298.2.1 `uint8_t uim_remainingRetries::unlockLeft`

8.298.2.2 `uint8_t uim_remainingRetries::verifyLeft`

## 8.299 uim\_sessionInformation Struct Reference

### Data Fields

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

### 8.299.1 Detailed Description

This structure contains the Session Information.

#### Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> <li>• Indicates the session type. <ul style="list-style-type: none"> <li>– 0 - Primary GW provisioning</li> <li>– 1 - Primary 1X provisioning</li> <li>– 2 - Secondary GW provisioning</li> <li>– 3 - Secondary 1X provisioning</li> <li>– 4 - Non-provisioning on slot 1</li> <li>– 5 - Non-provisioning on slot 2</li> <li>– 6 - Card on slot 1</li> <li>– 7 - Card on slot 2</li> <li>– 8 - Logical channel on slot 1</li> <li>– 9 - Logical channel on slot 2</li> </ul> </li> </ul>
	Generated by Doxygen

<i>aidLength</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. Application Identifier.</li> </ul>
<i>aid</i>	<ul style="list-style-type: none"> <li>Application identifier value or channel ID.</li> <li>This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases.</li> </ul>

## 8.299.2 Field Documentation

8.299.2.1 `uint8_t uim_sessionInformation::aid[255]`

8.299.2.2 `uint8_t uim_sessionInformation::aidLength`

8.299.2.3 `uint8_t uim_sessionInformation::sessionType`

## 8.300 uim\_setPINProtection Struct Reference

### Data Fields

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

### 8.300.1 Detailed Description

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

#### Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>Indicates the PIN ID to be enabled or disabled. <ul style="list-style-type: none"> <li>1 - PIN1 (also called PIN)</li> <li>2 - PIN2</li> <li>3 - Universal PIN</li> <li>4 - Hidden key</li> </ul> </li> </ul>
<i>pinOperation</i>	<ul style="list-style-type: none"> <li>Indicates whether the PIN is enabled or disabled. <ul style="list-style-type: none"> <li>0 - Disable the PIN</li> <li>1 - Enable the PIN</li> </ul> </li> </ul>
<i>pinLength</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. pin value.</li> </ul>
<i>pinValue</i> [MAX_PIN_LENGTH]	<ul style="list-style-type: none"> <li>PIN value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

### 8.300.2 Field Documentation

8.300.2.1 `uint8_t uim_setPINProtection::pinID`

8.300.2.2 `uint8_t uim_setPINProtection::pinLength`

8.300.2.3 `uint8_t uim_setPINProtection::pinOperation`

8.300.2.4 `uint8_t uim_setPINProtection::pinValue[255]`

## 8.301 `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]

### 8.301.1 Detailed Description

This structure contains information about the SLOTS present.

#### Parameters

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



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

## 8.301.2 Field Documentation

8.301.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.301.2.2 `uint8_t uim_slotInfo::cardState`

8.301.2.3 `uint8_t uim_slotInfo::errorState`

8.301.2.4 `uint8_t uim_slotInfo::numApp`

8.301.2.5 `uint8_t uim_slotInfo::upinRetries`

8.301.2.6 `uint8_t uim_slotInfo::upinState`

8.301.2.7 `uint8_t uim_slotInfo::upukRetries`

## 8.302 uim\_UIMSessionInformation Struct Reference

### Data Fields

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

### 8.302.1 Detailed Description

This structure contains the Session Information.

#### Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> <li>Indicates the session type. <ul style="list-style-type: none"> <li>0 - Primary GW provisioning</li> <li>1 - Primary 1X provisioning</li> <li>2 - Secondary GW provisioning</li> <li>3 - Secondary 1X provisioning</li> <li>4 - Non-provisioning on slot 1</li> <li>5 - Non-provisioning on slot 2</li> <li>6 - Card on slot 1</li> <li>7 - Card on slot 2</li> <li>8 - Logical channel on slot 1</li> <li>9 - Logical channel on slot 2</li> </ul> </li> </ul>
<i>aidLength</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. Application Identifier.</li> </ul>
<i>aid</i>	<ul style="list-style-type: none"> <li>Application identifier value or channel ID.</li> <li>This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases.</li> </ul>

### 8.302.2 Field Documentation

8.302.2.1 `uint8_t uim_UIMSessionInformation::aid[255]`

8.302.2.2 `uint8_t uim_UIMSessionInformation::aidLength`

8.302.2.3 `uint8_t uim_UIMSessionInformation::sessionType`

### 8.303 uim\_unblockUIMPIN Struct Reference

#### Data Fields

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

### 8.303.1 Detailed Description

This structure contains the information about the unblock pin parameters.

#### Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>Indicates the PIN ID to be changed. <ul style="list-style-type: none"> <li>1 - PIN1 (also called PIN)</li> <li>2 - PIN2</li> <li>3 - Universal PIN</li> </ul> </li> </ul>
<i>pukLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. puk value.</li> </ul>
<i>pukVal</i> [UIM_MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>PIN Unlock Key value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>
<i>newPINLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. new pin value.</li> </ul>
<i>newPINVal</i> [UIM_MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>New PIN value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

### 8.303.2 Field Documentation

8.303.2.1 `uint8_t uim_unblockUIMPIN::newPINLen`

8.303.2.2 `uint8_t uim_unblockUIMPIN::newPINVal[255]`

8.303.2.3 `uint8_t uim_unblockUIMPIN::pinID`

8.303.2.4 `uint8_t uim_unblockUIMPIN::pukLen`

8.303.2.5 `uint8_t uim_unblockUIMPIN::pukVal[255]`

## 8.304 uim\_verifyUIMPIN Struct Reference

### Data Fields

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

### 8.304.1 Detailed Description

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

#### Parameters

<i>pinID</i>	<ul style="list-style-type: none"> <li>Indicates the PIN ID to be verified. <ul style="list-style-type: none"> <li>1 - PIN1 (also called PIN)</li> <li>2 - PIN2</li> <li>3 - Universal PIN</li> <li>4 - Hidden key</li> </ul> </li> </ul>
<i>pinLen</i>	<ul style="list-style-type: none"> <li>Length of the following elements i.e. pin value.</li> </ul>
<i>pinVal</i> [ <i>MAX_DESCRIPTOR_LENGTH</i> ]	<ul style="list-style-type: none"> <li>PIN value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

### 8.304.2 Field Documentation

8.304.2.1 `uint8_t uim_verifyUIMPIN::pinID`

8.304.2.2 `uint8_t uim_verifyUIMPIN::pinLen`

8.304.2.3 `uint8_t uim_verifyUIMPIN::pinVal[255]`

## 8.305 `unpack_dms_GetActivationState_t` Struct Reference

### Data Fields

- `uint8_t state`

### 8.305.1 Detailed Description

#### Parameters

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

### 8.305.2 Field Documentation

8.305.2.1 uint8\_t unpack\_dms\_GetActivationState\_t::state

## 8.306 unpack\_dms\_GetBandCapability\_t Struct Reference

### Data Fields

- uint32\_t [BandCapability](#)
- uint16\_t [Tlvresult](#)

### 8.306.1 Field Documentation

8.306.1.1 uint32\_t unpack\_dms\_GetBandCapability\_t::BandCapability

8.306.1.2 uint16\_t unpack\_dms\_GetBandCapability\_t::Tlvresult

## 8.307 unpack\_dms\_GetCrashAction\_t Struct Reference

### Data Fields

- uint8\_t [DevCrashState](#)
- uint16\_t [Tlvresult](#)

### 8.307.1 Field Documentation

8.307.1.1 uint8\_t unpack\_dms\_GetCrashAction\_t::DevCrashState

8.307.1.2 uint16\_t unpack\_dms\_GetCrashAction\_t::Tlvresult

## 8.308 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](#)

### 8.308.1 Field Documentation

8.308.1.1 `uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled`

8.308.1.2 `uint8_t unpack_dms_GetCustFeature_t::DisableIMSI`

8.308.1.3 `uint32_t unpack_dms_GetCustFeature_t::GpsEnable`

8.308.1.4 `uint8_t unpack_dms_GetCustFeature_t::GPSLPM`

8.308.1.5 `uint8_t unpack_dms_GetCustFeature_t::GPSSel`

8.308.1.6 `uint16_t unpack_dms_GetCustFeature_t::IPFamSupport`

8.308.1.7 `uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled`

8.308.1.8 `uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect`

8.308.1.9 `uint8_t unpack_dms_GetCustFeature_t::SMSSupport`

8.308.1.10 `uint16_t unpack_dms_GetCustFeature_t::Tlvresult`

## 8.309 `unpack_dms_GetCustFeaturesV2_t` Struct Reference

### Data Fields

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

### 8.309.1 Detailed Description

This structure contains customization settings set to modem unpack

#### Parameters

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

### 8.309.2 Field Documentation

8.309.2.1 `DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2`

8.309.2.2 `uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult`

## 8.310 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](#)

### 8.310.1 Field Documentation

8.310.1.1 uint32\_t unpack\_dms\_GetDeviceCap\_t::DataServiceCapability

8.310.1.2 uint32\_t unpack\_dms\_GetDeviceCap\_t::MaxRXChannelRate

8.310.1.3 uint32\_t unpack\_dms\_GetDeviceCap\_t::MaxTXChannelRate

8.310.1.4 uint8\_t unpack\_dms\_GetDeviceCap\_t::Radiolfaces[64]

8.310.1.5 uint32\_t unpack\_dms\_GetDeviceCap\_t::RadiolfacesSize

8.310.1.6 uint32\_t unpack\_dms\_GetDeviceCap\_t::SimCapability

8.310.1.7 uint16\_t unpack\_dms\_GetDeviceCap\_t::Tlvresult

## 8.311 unpack\_dms\_GetDeviceCapabilities\_t Struct Reference

### Data Fields

- uint32\_t [maxTxChannelRate](#)
- uint32\_t [maxRxChannelRate](#)
- uint32\_t [dataServiceCaCapability](#)
- uint32\_t [simCapability](#)
- uint32\_t [radiolfacesSize](#)
- uint8\_t [Radiolfaces](#) [255]

### 8.311.1 Detailed Description

#### Parameters

<i>maxTx↔ ChannelRate</i>	Maximum Tx transmission rate in bits per second.
<i>maxRx↔ ChannelRate</i>	Maximum Rx transmission rate in bits per second
<i>dataServiceCa↔ Capability</i>	data service capability
<i>simCapability</i>	SIM Capability
<i>radiolfacesSize</i>	radio interface length
<i>Radiolfaces</i>	radio interfaces

### 8.311.2 Field Documentation

8.311.2.1 `uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability`

8.311.2.2 `uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate`

8.311.2.3 `uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate`

8.311.2.4 `uint8_t unpack_dms_GetDeviceCapabilities_t::Radiofaces[255]`

8.311.2.5 `uint32_t unpack_dms_GetDeviceCapabilities_t::radiofacesSize`

8.311.2.6 `uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability`

## 8.312 `unpack_dms_GetDeviceHardwareRev_t` Struct Reference

### Data Fields

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

### 8.312.1 Field Documentation

8.312.1.1 `char unpack_dms_GetDeviceHardwareRev_t::String[255]`

8.312.1.2 `uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize`

8.312.1.3 `uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult`

## 8.313 `unpack_dms_GetDeviceMfr_t` Struct Reference

### Data Fields

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



### 8.313.1 Field Documentation

8.313.1.1 char unpack\_dms\_GetDeviceMfr\_t::String[255]

8.313.1.2 uint8\_t unpack\_dms\_GetDeviceMfr\_t::stringSize

8.313.1.3 uint16\_t unpack\_dms\_GetDeviceMfr\_t::Tlvresult

## 8.314 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](#)

### 8.314.1 Field Documentation

8.314.1.1 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::esnSize

8.314.1.2 char unpack\_dms\_GetDeviceSerialNumbers\_t::ESNString[255]

8.314.1.3 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::imeiSize

8.314.1.4 char unpack\_dms\_GetDeviceSerialNumbers\_t::IMEIString[255]

8.314.1.5 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::imeiSvnSize

8.314.1.6 char unpack\_dms\_GetDeviceSerialNumbers\_t::ImeiSvnString[255]

8.314.1.7 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::meidSize

8.314.1.8 char unpack\_dms\_GetDeviceSerialNumbers\_t::MEIDString[255]

8.314.1.9 uint16\_t unpack\_dms\_GetDeviceSerialNumbers\_t::Tlvresult

## 8.315 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](#)

### 8.315.1 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"> <li>Package ID String.</li> <li>deprecated on EM/MC74xx(9x30) devices</li> </ul>
<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.

### 8.315.2 Field Documentation

8.315.2.1 `char unpack_dms_GetFirmwareInfo_t::appversion_str[85]`

8.315.2.2 `char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]`

8.315.2.3 `char unpack_dms_GetFirmwareInfo_t::carrier_str[20]`

8.315.2.4 `char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]`

8.315.2.5 `char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]`

8.315.2.6 `char unpack_dms_GetFirmwareInfo_t::modelid_str[20]`

8.315.2.7 `char unpack_dms_GetFirmwareInfo_t::packageid_str[85]`

8.315.2.8 `char unpack_dms_GetFirmwareInfo_t::priversion_str[16]`

8.315.2.9 `char unpack_dms_GetFirmwareInfo_t::sku_str[15]`

8.315.2.10 `uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult`

### 8.316 `unpack_dms_GetFirmwareRevision_t` Struct Reference

#### Data Fields

- `uint8_t amssSize`
- `char AMSSString [255]`
- `char PRIString [255]`
- `uint16_t Tlvresult`

### 8.316.1 Field Documentation

8.316.1.1 `uint8_t unpack_dms_GetFirmwareRevision_t::amssSize`

8.316.1.2 `char unpack_dms_GetFirmwareRevision_t::AMSSString[255]`

8.316.1.3 `char unpack_dms_GetFirmwareRevision_t::PRIString[255]`

8.316.1.4 `uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult`

## 8.317 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`

### 8.317.1 Detailed Description

#### Parameters

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

### 8.317.2 Field Documentation

8.317.2.1 `uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize`

8.317.2.2 `char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]`

8.317.2.3 `uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize`

8.317.2.4 `char unpack_dms_GetFirmwareRevisions_t::BootString[255]`

8.317.2.5 `uint8_t unpack_dms_GetFirmwareRevisions_t::priSize`

8.317.2.6 `char unpack_dms_GetFirmwareRevisions_t::PRIString[255]`

8.317.2.7 `uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult`

## 8.318 unpack\_dms\_GetFSN\_t Struct Reference

### Data Fields

- char [String](#) [255]
- uint16\_t [Tlvresult](#)

### 8.318.1 Field Documentation

8.318.1.1 char unpack\_dms\_GetFSN\_t::String[255]

8.318.1.2 uint16\_t unpack\_dms\_GetFSN\_t::Tlvresult

## 8.319 unpack\_dms\_GetHardwareRevision\_t Struct Reference

### Data Fields

- char [hwVer](#) [255]

### 8.319.1 Detailed Description

#### Parameters

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

### 8.319.2 Field Documentation

8.319.2.1 char unpack\_dms\_GetHardwareRevision\_t::hwVer[255]

## 8.320 unpack\_dms\_GetIMSI\_t Struct Reference

### Data Fields

- char [imsi](#) [255]
- uint16\_t [Tlvresult](#)

### 8.320.1 Field Documentation

8.320.1.1 char unpack\_dms\_GetIMSI\_t::imsi[255]

8.320.1.2 uint16\_t unpack\_dms\_GetIMSI\_t::Tlvresult

## 8.321 unpack\_dms\_GetManufacturer\_t Struct Reference

### Data Fields

- char [manufacturer](#) [255]
- uint16\_t [Tlvresult](#)

### 8.321.1 Detailed Description

This structure is used to store device manufacturer information.

#### Parameters

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

### 8.321.2 Field Documentation

8.321.2.1 char unpack\_dms\_GetManufacturer\_t::manufacturer[255]

8.321.2.2 uint16\_t unpack\_dms\_GetManufacturer\_t::Tlvresult

## 8.322 unpack\_dms\_GetModelID\_t Struct Reference

#### Data Fields

- char [modelid](#) [255]
- uint16\_t [Tlvresult](#)

### 8.322.1 Detailed Description

#### Parameters

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

### 8.322.2 Field Documentation

8.322.2.1 char unpack\_dms\_GetModelID\_t::modelid[255]

8.322.2.2 uint16\_t unpack\_dms\_GetModelID\_t::Tlvresult

## 8.323 unpack\_dms\_GetNetworkTime\_t Struct Reference

#### Data Fields

- uint16\_t [source](#)
- uint64\_t [timestamp](#)
- uint16\_t [Tlvresult](#)

### 8.323.1 Detailed Description

#### Parameters

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

#### Note

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

### 8.323.2 Field Documentation

8.323.2.1 `uint16_t unpack_dms_GetNetworkTime_t::source`

8.323.2.2 `uint64_t unpack_dms_GetNetworkTime_t::timestamp`

8.323.2.3 `uint16_t unpack_dms_GetNetworkTime_t::Tlvresult`

## 8.324 `unpack_dms_GetOfflineReason_t` Struct Reference

#### Data Fields

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

### 8.324.1 Detailed Description

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

## Parameters

<i>pReasonMask</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Bitmask of offline reasons <ul style="list-style-type: none"> <li>0x00000001 - Host image configuration issue</li> <li>0x00000002 - PRI image configuration issue</li> <li>0x00000004 - PRI version incompatible</li> <li>0x00000008 - PRI copy issue</li> <li>All others - Reserved</li> </ul> </li> </ul>
<i>pbPlatform</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Is the device offline due to a platform restriction? <ul style="list-style-type: none"> <li>0 - No</li> <li>1 - Yes</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

## 8.324.2 Field Documentation

8.324.2.1 uint32\_t\* unpack\_dms\_GetOfflineReason\_t::pbPlatform

8.324.2.2 uint32\_t\* unpack\_dms\_GetOfflineReason\_t::pReasonMask

8.324.2.3 uint16\_t unpack\_dms\_GetOfflineReason\_t::Tlvresult

## 8.325 unpack\_dms\_GetPower\_t Struct Reference

## Data Fields

- uint32\_t [OperationMode](#)
- uint32\_t [OfflineReason](#)
- uint32\_t [HardwareControlledMode](#)
- uint16\_t [Tlvresult](#)

## 8.325.1 Detailed Description

## Parameters

<i>OperationMode</i>	operating mode
<i>OfflineReason</i>	offline reason
<i>HardwareControlledMode</i>	hardware restricted mode

## 8.325.2 Field Documentation

8.325.2.1 uint32\_t unpack\_dms\_GetPower\_t::HardwareControlledMode

8.325.2.2 uint32\_t unpack\_dms\_GetPower\_t::OfflineReason

8.325.2.3 uint32\_t unpack\_dms\_GetPower\_t::OperationMode

8.325.2.4 uint16\_t unpack\_dms\_GetPower\_t::Tlvresult

## 8.326 unpack\_dms\_GetPRLVersion\_t Struct Reference

### Data Fields

- uint8\_t [u8PRLPreference](#)
- uint16\_t [u16PRLVersion](#)
- uint16\_t [Tlvresult](#)

### 8.326.1 Field Documentation

8.326.1.1 uint16\_t unpack\_dms\_GetPRLVersion\_t::Tlvresult

8.326.1.2 uint16\_t unpack\_dms\_GetPRLVersion\_t::u16PRLVersion

8.326.1.3 uint8\_t unpack\_dms\_GetPRLVersion\_t::u8PRLPreference

## 8.327 unpack\_dms\_GetSerialNumbers\_t Struct Reference

### Data Fields

- char [esn](#) [255]
- char [imei\\_no](#) [255]
- char [meid](#) [255]
- char [imeisv\\_svn](#) [255]

### 8.327.1 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

### 8.327.2 Field Documentation

8.327.2.1 char unpack\_dms\_GetSerialNumbers\_t::esn[255]



8.327.2.2 char unpack\_dms\_GetSerialNumbers\_t::imei\_no[255]

8.327.2.3 char unpack\_dms\_GetSerialNumbers\_t::imeisv\_svn[255]

8.327.2.4 char unpack\_dms\_GetSerialNumbers\_t::meid[255]

## 8.328 unpack\_dms\_GetUSBComp\_t Struct Reference

### Data Fields

- uint8\_t [USBComp](#) [255]
- uint8\_t [NumSupUSBComps](#)
- uint8\_t [SupUSBComps](#)
- uint16\_t [Tlvresult](#)

### 8.328.1 Field Documentation

8.328.1.1 uint8\_t unpack\_dms\_GetUSBComp\_t::NumSupUSBComps

8.328.1.2 uint8\_t unpack\_dms\_GetUSBComp\_t::SupUSBComps

8.328.1.3 uint16\_t unpack\_dms\_GetUSBComp\_t::Tlvresult

8.328.1.4 uint8\_t unpack\_dms\_GetUSBComp\_t::USBComp[255]

## 8.329 unpack\_dms\_GetVoiceNumber\_t Struct Reference

### Data Fields

- uint8\_t [voiceNumberSize](#)
- char [VoiceNumber](#) [255]
- uint8\_t [minSize](#)
- char [MIN](#) [255]
- uint16\_t [Tlvresult](#)

### 8.329.1 Field Documentation

8.329.1.1 char unpack\_dms\_GetVoiceNumber\_t::MIN[255]

8.329.1.2 uint8\_t unpack\_dms\_GetVoiceNumber\_t::minSize

8.329.1.3 uint16\_t unpack\_dms\_GetVoiceNumber\_t::Tlvresult

8.329.1.4 char unpack\_dms\_GetVoiceNumber\_t::VoiceNumber[255]

8.329.1.5 uint8\_t unpack\_dms\_GetVoiceNumber\_t::voiceNumberSize

## 8.330 unpack\_dms\_ResetToFactoryDefaults\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.330.1 Detailed Description

This structure contains reset to factory default unpack

#### Parameters

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

### 8.330.2 Field Documentation

8.330.2.1 uint16\_t unpack\_dms\_ResetToFactoryDefaults\_t::Tlvresult

## 8.331 unpack\_dms\_SetActivationStatusCallback\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.331.1 Detailed Description

unpack structure for Activation status callback

#### Parameters

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

### 8.331.2 Field Documentation

8.331.2.1 uint16\_t unpack\_dms\_SetActivationStatusCallback\_t::Tlvresult

## 8.332 unpack\_dms\_SetCrashAction\_t Struct Reference

#### Data Fields

- uint8\_t [notused](#)

### 8.332.1 Detailed Description

Modem response. Not used

### 8.332.2 Field Documentation

8.332.2.1 uint8\_t unpack\_dms\_SetCrashAction\_t::notused

## 8.333 unpack\_dms\_SetCustFeature\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.333.1 Field Documentation

8.333.1.1 uint16\_t unpack\_dms\_SetCustFeature\_t::Tlvresult

## 8.334 unpack\_dms\_SetCustFeaturesV2\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.334.1 Detailed Description

This structure contains customization settings set to modem unpack

#### Parameters

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

### 8.334.2 Field Documentation

8.334.2.1 uint16\_t unpack\_dms\_SetCustFeaturesV2\_t::Tlvresult

## 8.335 unpack\_dms\_SetEventReport\_ind\_t Struct Reference

### Data Fields

- [dms\\_ActivationStatusTlv](#) ActivationStatusTlv
- [dms\\_OperatingModeTlv](#) OperatingModeTlv
- uint16\_t [Tlvresult](#)

### 8.335.1 Detailed Description

DMS Event Report indication structure

#### Parameters

<i>Activation↔ StatusTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_ActivationStatusTlv</a></li> </ul>
<i>Operating↔ ModeTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_OperatingModeTlv</a></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

### 8.335.2 Field Documentation

8.335.2.1 `dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv`

8.335.2.2 `dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv`

8.335.2.3 `uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult`

## 8.336 `unpack_dms_SetEventReport_t` Struct Reference

#### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.336.1 Field Documentation

8.336.1.1 `uint16_t unpack_dms_SetEventReport_t::Tlvresult`

## 8.337 `unpack_dms_SetFirmwarePreference_t` Struct Reference

#### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.337.1 Field Documentation

8.337.1.1 `uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult`

## 8.338 `unpack_dms_SetPower_t` Struct Reference

#### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.338.1 Field Documentation

8.338.1.1 uint16\_t unpack\_dms\_SetPower\_t::Tlvresult

## 8.339 unpack\_dms\_SetUSBComp\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.339.1 Field Documentation

8.339.1.1 uint16\_t unpack\_dms\_SetUSBComp\_t::Tlvresult

## 8.340 unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t Struct Reference

### Data Fields

- uint8\_t [type](#)
- uint8\_t [source](#)
- uint16\_t [Tlvresult](#)

### 8.340.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

#### Parameters

--	--

### 8.340.2 Field Documentation

8.340.2.1 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t::source

8.340.2.2 uint16\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t::Tlvresult

8.340.2.3 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t::type

## 8.341 unpack\_dms\_SLQSDmsSwiGetResetInfo\_t Struct Reference

### Data Fields

- uint8\_t [type](#)
- uint8\_t [source](#)
- uint16\_t [Tlvresult](#)

### 8.341.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

#### Parameters

--	--

### 8.341.2 Field Documentation

8.341.2.1 `uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source`

8.341.2.2 `uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult`

8.341.2.3 `uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type`

## 8.342 `unpack_dms_SLQSDmsSwiIndicationRegister_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.342.1 Detailed Description

This structure contains set registration state for different indication unpack

#### Parameters

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

### 8.342.2 Field Documentation

8.342.2.1 `uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::Tlvresult`

## 8.343 `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](#)

## 8.343.1 Detailed Description

This structure contains the Band Capabilities response.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

## Parameters

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

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



<i>TdsBand</i> ↔ <i>Capability[OUT]</i>	Bitmask of TDS bands supported by the device. <ul style="list-style-type: none"><li>• Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz</li><li>• Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz</li><li>• Bit 2 - TDS Band C 1910 to 1930 MHz</li><li>• Bit 3 - TDS Band D 2570 to 2620 MHz</li><li>• Bit 4 - TDS Band E 2300 to 2400 MHz</li><li>• Bit 5 - TDS Band F 1880 to 1920 MHz</li></ul>
--	---

### 8.343.2 Field Documentation

8.343.2.1 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::bandCapability

8.343.2.2 int unpack\_dms\_SLQSGetBandCapability\_t::is\_LteBandCapability\_Available

8.343.2.3 int unpack\_dms\_SLQSGetBandCapability\_t::is\_TdsBandCapability\_Available

8.343.2.4 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::LteBandCapability

8.343.2.5 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::TdsBandCapability

## 8.344 unpack\_dms\_SLQSGetERIFile\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- [eriDataparams](#) eriFile

### 8.344.1 Detailed Description

This structure contains Get ERI file unpack

#### Parameters

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

### 8.344.2 Field Documentation

8.344.2.1 [eriDataparams](#) unpack\_dms\_SLQSGetERIFile\_t::eriFile

8.344.2.2 uint16\_t unpack\_dms\_SLQSGetERIFile\_t::Tlvresult

## 8.345 unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.345.1 Detailed Description

This structure contains Clear Dying GASP unpack

#### Parameters

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

### 8.345.2 Field Documentation

8.345.2.1 uint16\_t unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t::Tlvresult

## 8.346 unpack\_dms\_SLQSSwiGetCrashInfo\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- [crashInfoParams](#) [crashInfoParam](#)

### 8.346.1 Detailed Description

This structure contains SWI get crash information unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>crashInfoParam</i> <i>Param[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">crashInfoParams</a></li> </ul>

### 8.346.2 Field Documentation

8.346.2.1 crashInfoParams unpack\_dms\_SLQSSwiGetCrashInfo\_t::crashInfoParam

8.346.2.2 uint16\_t unpack\_dms\_SLQSSwiGetCrashInfo\_t::Tlvresult

## 8.347 unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t Struct Reference

### Data Fields

- [packgetDyingGaspCfg](#) \* [pGetDyingGaspCfg](#)
- [uint16\\_t](#) [Tlvresult](#)

### 8.347.1 Detailed Description

This structure contains Get Dying GASP Config unpack

#### Parameters

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

### 8.347.2 Field Documentation

8.347.2.1 [packgetDyingGaspCfg](#)\* [unpack\\_dms\\_SLQSSwiGetDyingGaspCfg\\_t::pGetDyingGaspCfg](#)

8.347.2.2 [uint16\\_t](#) [unpack\\_dms\\_SLQSSwiGetDyingGaspCfg\\_t::Tlvresult](#)

## 8.348 unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t Struct Reference

### Data Fields

- [packgetDyingGaspStatistics](#) \* [pGetDyingGaspStatistics](#)
- [uint16\\_t](#) [Tlvresult](#)

### 8.348.1 Detailed Description

This structure contains Get Dying GASP Statistics.

#### Parameters

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

### 8.348.2 Field Documentation

8.348.2.1 [packgetDyingGaspStatistics](#)\* [unpack\\_dms\\_SLQSSwiGetDyingGaspStatistics\\_t::pGetDyingGaspStatistics](#)

8.348.2.2 `uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

## 8.349 `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]`

### 8.349.1 Detailed Description

#### Parameters

<i>numEntries</i> [↔ N/OUT]	<ul style="list-style-type: none"> <li>• Number of entries in the image list to follow</li> <li>• The size of the list <code>pCurrImgInfo</code> must be specified when calling the API</li> </ul>
<i>pCurrImgInfo</i> [↔ OUT]	<ul style="list-style-type: none"> <li>• Currently Active Image List</li> </ul>
<i>priver</i> [OUT]	<ul style="list-style-type: none"> <li>• PRI version of the currently running firmware</li> </ul>
<i>pkgver</i> [OUT]	<ul style="list-style-type: none"> <li>• Package version of the currently running firmware</li> </ul>
<i>fwvers</i> [OUT]	<ul style="list-style-type: none"> <li>• firmware version of the currently running firmware</li> </ul>
<i>carrier</i> [OUT]	<ul style="list-style-type: none"> <li>• Carrier string of the currently running firmware</li> </ul>

### 8.349.2 Field Documentation

8.349.2.1 `char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]`

8.349.2.2 `char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]`

8.349.2.3 `uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries`

8.349.2.4 `image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo`

8.349.2.5 `char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]`

8.349.2.6 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::priver[16]

## 8.350 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](#)

### 8.350.1 Detailed Description

This structure is used to store Firmware Update Status

#### Parameters

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

<i>logString</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

## 8.350.2 Field Documentation

8.350.2.1 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType`

8.350.2.2 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]`

8.350.2.3 `uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData`

8.350.2.4 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]`

8.350.2.5 `uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode`

8.350.2.6 `uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult`

## 8.351 `unpack_dms_SLQSSwiGetHostDevInfo_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `char` [manString](#) [255]
- `char` [modelString](#) [255]
- `char` [swVerString](#) [255]
- `char` [plasmaIDString](#) [255]

### 8.351.1 Detailed Description

This structure contains SWI get host device info unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Result</li> </ul>
<i>manString[OUT]</i>	<ul style="list-style-type: none"> <li>optional parameter, host device manufacture</li> </ul>
<i>modelString[O↔UT]</i>	<ul style="list-style-type: none"> <li>optional parameter, host device model</li> </ul>
<i>swVerString[O↔UT]</i>	<ul style="list-style-type: none"> <li>optional parameter, host device software version</li> </ul>
<i>plasmaID↔</i>	

### 8.351.2 Field Documentation

8.351.2.1 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::manString[255]

8.351.2.2 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::modelString[255]

8.351.2.3 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::plasmaIDString[255]

8.351.2.4 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::swVerString[255]

8.351.2.5 uint16\_t unpack\_dms\_SLQSSwiGetHostDevInfo\_t::Tlvresult

## 8.352 unpack\_dms\_SLQSSwiGetOSInfo\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- char [nameString](#) [255]
- char [versionString](#) [255]

### 8.352.1 Detailed Description

This structure contains SWI get host os info unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• pack Result</li></ul>
<i>nameString</i> [ <i>OPTIONAL</i> ]	<ul style="list-style-type: none"><li>• optional parameter, host operating system name</li></ul>
<i>VersionString</i> [ <i>OPTIONAL</i> ]	<ul style="list-style-type: none"><li>• optional parameter, host operating system version</li></ul>

### 8.352.2 Field Documentation

8.352.2.1 char unpack\_dms\_SLQSSwiGetOSInfo\_t::nameString[255]

8.352.2.2 uint16\_t unpack\_dms\_SLQSSwiGetOSInfo\_t::Tlvresult

8.352.2.3 char unpack\_dms\_SLQSSwiGetOSInfo\_t::versionString[255]

## 8.353 unpack\_dms\_SLQSSwiGetSerialNoExt\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)
- char [meidString](#) [8]

### 8.353.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• pack Result</li></ul>
<i>meidString[OUT]</i>	<ul style="list-style-type: none"><li>• optional parameter, mobile equipment identifier</li></ul>

### 8.353.2 Field Documentation

8.353.2.1 char `unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]`

8.353.2.2 uint16\_t `unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult`

## 8.354 unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.354.1 Detailed Description

This structure contains set Dying GASP Config unpack

#### Parameters

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

### 8.354.2 Field Documentation

8.354.2.1 uint16\_t `unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult`

## 8.355 unpack\_dms\_SLQSSwiSetHostDevInfo\_t Struct Reference



## Data Fields

- uint16\_t [Tlvresult](#)

### 8.355.1 Detailed Description

This structure contains SWI set host dev info unpack

#### Parameters

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

### 8.355.2 Field Documentation

8.355.2.1 uint16\_t unpack\_dms\_SLQSSwiSetHostDevInfo\_t::Tlvresult

## 8.356 unpack\_dms\_SLQSSwiSetOSInfo\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.356.1 Detailed Description

This structure contains SWI set host OS info unpack

#### Parameters

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

### 8.356.2 Field Documentation

8.356.2.1 uint16\_t unpack\_dms\_SLQSSwiSetOSInfo\_t::Tlvresult

## 8.357 unpack\_dms\_SLQSUIMGetState\_t Struct Reference

## Data Fields

- uint8\_t [state](#)
- uint16\_t [Tlvresult](#)

### 8.357.1 Detailed Description

This structure contains UIM get state unpack information

#### Parameters

<i>state[OUT]</i>	<ul style="list-style-type: none"> <li>UIM state</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

### 8.357.2 Field Documentation

8.357.2.1 uint8\_t unpack\_dms\_SLQSUIMGetState\_t::state

8.357.2.2 uint16\_t unpack\_dms\_SLQSUIMGetState\_t::Tlvresult

## 8.358 unpack\_dms\_UIMGetControlKeyStatus\_t Struct Reference

### Data Fields

- uint8\_t [facilityState](#)
- uint8\_t [verifyRetriesLeft](#)
- uint8\_t [unblockRetriesLeft](#)
- uint16\_t [Tlvresult](#)

### 8.358.1 Detailed Description

This structure contains PIN retries status

#### Parameters

<i>facilityState[Optional UT]</i>	<ul style="list-style-type: none"> <li>Control key status <ul style="list-style-type: none"> <li>0 - Deactivated</li> <li>1 - Activated</li> <li>2 - Blocked</li> </ul> </li> </ul>
<i>verifyRetriesLeft[Optional Left[OUT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>unblockRetriesLeft[Optional Left[OUT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
	Generated by Doxygen
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

### 8.358.2 Field Documentation

8.358.2.1 uint8\_t unpack\_dms\_UIMGetControlKeyStatus\_t::facilityState

8.358.2.2 uint16\_t unpack\_dms\_UIMGetControlKeyStatus\_t::Tlvresult

8.358.2.3 uint8\_t unpack\_dms\_UIMGetControlKeyStatus\_t::unlockRetriesLeft

8.358.2.4 uint8\_t unpack\_dms\_UIMGetControlKeyStatus\_t::verifyRetriesLeft

## 8.359 unpack\_dms\_UIMGetICCID\_t Struct Reference

### Data Fields

- uint8\_t [stringSize](#)
- uint8\_t [String](#) [255]
- uint16\_t [Tlvresult](#)

### 8.359.1 Detailed Description

This structure contains Get ICCID pack

#### Parameters

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

### 8.359.2 Field Documentation

8.359.2.1 uint8\_t unpack\_dms\_UIMGetICCID\_t::String[255]

8.359.2.2 uint8\_t unpack\_dms\_UIMGetICCID\_t::stringSize

8.359.2.3 uint16\_t unpack\_dms\_UIMGetICCID\_t::Tlvresult

## 8.360 unpack\_dms\_UIMGetPINStatus\_t Struct Reference

### Data Fields

- uint8\_t [p1Status](#)

- uint8\_t [p1VerifyRetriesLeft](#)
- uint8\_t [p1UnblockRetriesLeft](#)
- uint8\_t [p2Status](#)
- uint8\_t [p2VerifyRetriesLeft](#)
- uint8\_t [p2UnblockRetriesLeft](#)
- uint16\_t [Tlvresult](#)

### 8.360.1 Detailed Description

This structure contains Get PIN Status unpack information

#### Parameters

<i>p1Status[OUT]</i>	<ul style="list-style-type: none"> <li>• PIN1 status(0xFFFFFFFF - Unknown) <ul style="list-style-type: none"> <li>– 0 - PIN not initialized</li> <li>– 1 - PIN enabled, not verified</li> <li>– 2 - PIN enabled, verified</li> <li>– 3 - PIN disabled</li> <li>– 4 - PIN blocked</li> <li>– 5 - PIN permanently blocked</li> </ul> </li> </ul>
<i>p1Verify↔ RetriesLeft[O↔ UT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of retries left of PIN1, after which PIN will be blocked. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>p1Unblock↔ RetriesLeft[O↔ UT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of unblock retries left of PIN1, after which the PIN will be permanently blocked i.e. UIM is unusable. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>p2Status[OUT]</i>	<ul style="list-style-type: none"> <li>• PIN2 status(0xFFFFFFFF - Unknown) <ul style="list-style-type: none"> <li>– 0 - PIN not initialized</li> <li>– 1 - PIN enabled, not verified</li> <li>– 2 - PIN enabled, verified</li> <li>– 3 - PIN disabled</li> <li>– 4 - PIN blocked</li> <li>– 5 - PIN permanently blocked</li> </ul> </li> </ul>
<i>p2Verify↔ RetriesLeft[O↔ UT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of retries left of PIN2, after which PIN will be blocked. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>

<i>p2UnblockRetriesLeft[Optional UT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Upon operational failure, this will indicate number of unblock retries left of PIN2, after which the PIN will be permanently blocked i.e. UIM is unusable. <ul style="list-style-type: none"> <li>0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

## 8.360.2 Field Documentation

8.360.2.1 uint8\_t unpack\_dms\_UIMGetPINStatus\_t::p1Status

8.360.2.2 uint8\_t unpack\_dms\_UIMGetPINStatus\_t::p1UnblockRetriesLeft

8.360.2.3 uint8\_t unpack\_dms\_UIMGetPINStatus\_t::p1VerifyRetriesLeft

8.360.2.4 uint8\_t unpack\_dms\_UIMGetPINStatus\_t::p2Status

8.360.2.5 uint8\_t unpack\_dms\_UIMGetPINStatus\_t::p2UnblockRetriesLeft

8.360.2.6 uint8\_t unpack\_dms\_UIMGetPINStatus\_t::p2VerifyRetriesLeft

8.360.2.7 uint16\_t unpack\_dms\_UIMGetPINStatus\_t::Tlvresult

## 8.361 unpack\_dms\_UIMSetControlKeyProtection\_t Struct Reference

### Data Fields

- uint8\_t [verifyRetriesLeft](#)
- uint16\_t [Tlvresult](#)

### 8.361.1 Detailed Description

This structure contains UIM Set control key protection unpack information

#### Parameters

<i>verifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>number of retries left after which the control key is blocked</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

### 8.361.2 Field Documentation

8.361.2.1 uint16\_t unpack\_dms\_UIMSetControlKeyProtection\_t::Tlvresult

8.361.2.2 uint8\_t unpack\_dms\_UIMSetControlKeyProtection\_t::verifyRetriesLeft

## 8.362 unpack\_dms\_UIMSetPINProtection\_t Struct Reference

### Data Fields

- uint8\_t [verifyRetriesLeft](#)
- uint8\_t [unblockRetriesLeft](#)
- uint16\_t [Tlvresult](#)

### 8.362.1 Detailed Description

This structure contains PIN retries status

#### Parameters

<i>verifyRetriesLeft</i> <i>Left[OUT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>unblockRetriesLeft</i> <i>Left[OUT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable. <ul style="list-style-type: none"> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

### 8.362.2 Field Documentation

8.362.2.1 uint16\_t unpack\_dms\_UIMSetPINProtection\_t::Tlvresult

8.362.2.2 uint8\_t unpack\_dms\_UIMSetPINProtection\_t::unblockRetriesLeft

8.362.2.3 uint8\_t unpack\_dms\_UIMSetPINProtection\_t::verifyRetriesLeft

## 8.363 unpack\_dms\_UIMUnblockControlKey\_t Struct Reference

### Data Fields

- uint8\_t [unblockRetriesLeft](#)
- uint16\_t [Tlvresult](#)

### 8.363.1 Detailed Description

This structure contains UIM Set control key protection unpack information

#### Parameters

<i>unlockRetriesLeft</i> <i>Left[OUT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>number of unlock retries left after which the control key is permanently blocked</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

### 8.363.2 Field Documentation

8.363.2.1 uint16\_t unpack\_dms\_UIMUnlockControlKey\_t::Tlvresult

8.363.2.2 uint8\_t unpack\_dms\_UIMUnlockControlKey\_t::unlockRetriesLeft

## 8.364 unpack\_fms\_GetImagesPreference\_t Struct Reference

### Data Fields

- uint32\_t [ImageListSize](#)
- FMSPrefImageList \* [pImageList](#)
- uint16\_t [Tlvresult](#)

### 8.364.1 Detailed Description

This structure contains the Get Image Preference information unpack

#### Parameters

<i>listSize</i>	<ul style="list-style-type: none"> <li>The number of elements in the image list</li> </ul>
<i>pListEntries</i>	<ul style="list-style-type: none"> <li>Array of Image entries with size provided by previous field</li> <li>See <a href="#">FMSImageElement</a></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack result</li> </ul>

### 8.364.2 Field Documentation

8.364.2.1 `uint32_t unpack_fms_GetImagesPreference_t::ImageListSize`

8.364.2.2 `FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList`

8.364.2.3 `uint16_t unpack_fms_GetImagesPreference_t::Tlvresult`

## 8.365 `unpack_fms_GetStoredImages_t` Struct Reference

### Data Fields

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

### 8.365.1 Detailed Description

This structure contains the Get Stored Images unpack

#### Parameters

<i>listSize</i>	<ul style="list-style-type: none"> <li>• The number of elements in the image list</li> </ul>
<i>imageList</i>	<ul style="list-style-type: none"> <li>• Array of Image entries with size provided by previous field</li> <li>• See <a href="#">FMSImageElement</a></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result</li> </ul>

### 8.365.2 Field Documentation

8.365.2.1 `FMSImageList unpack_fms_GetStoredImages_t::imageList`

8.365.2.2 `uint32_t unpack_fms_GetStoredImages_t::imagelistSize`

8.365.2.3 `uint16_t unpack_fms_GetStoredImages_t::Tlvresult`

## 8.366 `unpack_fms_SetImagesPreference_t` Struct Reference

### Data Fields

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



### 8.366.1 Detailed Description

This structure contains the Set Images Preference unpack

#### Parameters

<i>ImageTypesSize</i>	<ul style="list-style-type: none"> <li>Image Type Size</li> </ul>
<i>ImageTypes</i>	<ul style="list-style-type: none"> <li>Image Type</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack result</li> </ul>

### 8.366.2 Field Documentation

8.366.2.1 `uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]`

8.366.2.2 `uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize`

8.366.2.3 `uint16_t unpack_fms_SetImagesPreference_t::Tlvresult`

## 8.367 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](#)

### 8.367.1 Detailed Description

This structure contains Best Available Position

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Valid values: <ul style="list-style-type: none"> <li>– eQMI_LOC_SUCCESS (0) - Request was completed successfully</li> <li>– eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure</li> <li>– eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported</li> <li>– eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters</li> <li>– eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy</li> <li>– eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline</li> <li>– eQMI_LOC_TIMEOUT (6) - Request failed because it timed out</li> <li>– eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested</li> <li>– eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request</li> <li>– eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed</li> <li>– eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure</li> </ul> </li> </ul>
<i>xid</i>	Transaction ID that was specified in the Get Best Available Position request.
<i>pLatitude</i>	<ul style="list-style-type: none"> <li>• Latitude (specified in WGS84 datum)</li> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -90.0 to 90.0</li> <li>• Positive values indicate northern latitude</li> <li>• Negative values indicate southern latitude</li> </ul>
<i>pLongitude</i>	<ul style="list-style-type: none"> <li>• Longitude (specified in WGS84 datum)</li> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -180.0 to 180.0</li> <li>• Positive values indicate eastern latitude</li> <li>• Negative values indicate western latitude</li> </ul>

<i>pHorUncCircular</i>	<ul style="list-style-type: none"> <li>• Horizontal position uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pAltitudeWrt↔ Ellipsoid</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to WGS84 Ellipsoid.</li> <li>• Units - Meters</li> <li>• Range -500 to 15883</li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> </ul>
<i>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</li> </ul>
<i>pHorUnc↔ EllipseSemi↔ Minor</i>	<ul style="list-style-type: none"> <li>• Semi-minor axis of horizontal elliptical uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pHorUnc↔ EllipseSemi↔ Major</i>	<ul style="list-style-type: none"> <li>• Semi-major axis of horizontal elliptical uncertainty.</li> <li>• Units: Meters</li> </ul>
<i>pHorUnc↔ EllipseOrient↔ Azimuth</i>	<ul style="list-style-type: none"> <li>• Elliptical horizontal uncertainty azimuth of orientation.</li> <li>• Units - Decimal degrees</li> <li>• Range - 0 to 180</li> </ul>
<i>pHorCirConf</i>	<ul style="list-style-type: none"> <li>• Horizontal circular uncertainty confidence</li> <li>• Units: Percent</li> <li>• Range: 0 to 99</li> </ul>
<i>pHorEllpConf</i>	<ul style="list-style-type: none"> <li>• Horizontal elliptical uncertainty confidence</li> <li>• Units: Percent</li> <li>• Range: 0 to 99</li> </ul>
<i>pHorReliability</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Location reliability is not set.</li> <li>– 1 - Location reliability is very low; use it at your own risk</li> <li>– 2 - Location reliability is low; little or no cross-checking is possible.</li> <li>– 3 - Location reliability is medium; limited cross-check passed</li> <li>– 4 - Location reliability is high; strong cross-check passed</li> </ul> </li> </ul>

<i>pSpeed</i> ↔ <i>Horizontal</i>	<ul style="list-style-type: none"> <li>• Horizontal speed.</li> <li>• Units - Meters/second</li> </ul>
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> <li>• 3-D Speed uncertainty.</li> <li>• Units - Meters/second.</li> </ul>
<i>pAltitudeWrt</i> ↔ <i>MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to Sea Level.</li> <li>• Units - Meters</li> </ul>
<i>pVertConfidence</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty confidence.</li> <li>• Units - Percentage</li> <li>• Range 0 to 99</li> </ul>
<i>pVertReliability</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Location reliability is not set.</li> <li>– 1 - Location reliability is very low; use it at your own risk.</li> <li>– 2 - Location reliability is low; little or no cross-checking is possible</li> <li>– 3 - Location reliability is medium; limited cross-check passed</li> <li>– 4 - Location reliability is high; strong cross-check passed</li> </ul> </li> </ul>
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> <li>• Vertical speed.</li> <li>• Units - Meters/second</li> </ul>
<i>pSpeed</i> ↔ <i>VerticalUnc</i>	<ul style="list-style-type: none"> <li>• Vertical speed</li> <li>• Units: Meters/second</li> </ul>
<i>pHeading</i>	<ul style="list-style-type: none"> <li>• Heading.</li> <li>• Units - Degree</li> <li>• Range 0 to 359.999</li> </ul>
<i>pHeadingUnc</i>	<ul style="list-style-type: none"> <li>• Heading uncertainty.</li> <li>• Units - Degree</li> <li>• Range 0 to 359.999</li> </ul>
<i>pMagnetic</i> ↔ <i>Deviation</i>	<ul style="list-style-type: none"> <li>• Difference between the bearing to true north and the bearing shown on a magnetic compass. The deviation is positive when the magnetic north is east of true north.</li> </ul>

<i>pTechnology↔ Mask</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0x00000001 - Satellites were used to generate the fix</li> <li>– 0x00000002 - Cell towers were used to generate the fix</li> <li>– 0x00000004 - Wi-Fi access points were used to generate the fix</li> <li>– 0x00000008 - Sensors were used to generate the fix</li> <li>– 0x00000010 - Reference Location was used to generate the fix</li> <li>– 0x00000020 - Coarse position injected into the location engine was used to generate the fix</li> <li>– 0x00000040 - AFLT was used to generate the fix</li> <li>– 0x00000080 - GNSS and network-provided measurements were used to generate the fix</li> </ul> </li> </ul>
<i>-pPrecision↔ Dilution</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_precisionDilution</a> for more information</li> </ul>
<i>-pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gpsTime</a> for more information</li> </ul>
<i>pTimeSrc</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Invalid time.</li> <li>– 1 - Time is set by the 1X system.</li> <li>– 2 - Time is set by WCDMA/GSM time tagging.</li> <li>– 3 - Time is set by an external injection.</li> <li>– 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite.</li> <li>– 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites.</li> <li>– 6 - Both time of the week and the GPS week number are known.</li> <li>– 7 - Time is set by the position engine after the fix is obtained</li> <li>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.</li> <li>– 9 - Time is set after decoding GLO satellites.</li> <li>– 10- Time is set after transforming the GPS to GLO time</li> <li>– 11- Time is set by the sleep time tag provided by the WCDMA network.</li> <li>– 12- Time is set by the sleep time tag provided by the GSM network</li> <li>– 13- Source of the time is unknown</li> <li>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state</li> <li>– 15- Time is set after decoding QZSS satellites.</li> <li>– 16- Time is set after decoding BDS satellites.</li> </ul> </li> </ul>
<i>-pSensorData↔ Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_sensorDataUsage</a> for more information</li> </ul>
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_svUsedforFix</a> for more information</li> </ul>

## 8.367.2 Field Documentation

### 8.367.2.1 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pAltitudeWrtEllipsoid

- 8.367.2.2    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel`
- 8.367.2.3    `loc_gpsTime*` `unpack_loc_BestAvailPos_Ind_t::pGpsTime`
- 8.367.2.4    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHeading`
- 8.367.2.5    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHeadingUnc`
- 8.367.2.6    `uint8_t*` `unpack_loc_BestAvailPos_Ind_t::pHorCirConf`
- 8.367.2.7    `uint8_t*` `unpack_loc_BestAvailPos_Ind_t::pHorEllpConf`
- 8.367.2.8    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHorReliability`
- 8.367.2.9    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHorUncCircular`
- 8.367.2.10    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.367.2.11    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor`
- 8.367.2.12    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor`
- 8.367.2.13    `uint64_t*` `unpack_loc_BestAvailPos_Ind_t::pLatitude`
- 8.367.2.14    `uint64_t*` `unpack_loc_BestAvailPos_Ind_t::pLongitude`
- 8.367.2.15    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation`
- 8.367.2.16    `loc_precisionDilution*` `unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution`
- 8.367.2.17    `loc_sensorDataUsage*` `unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage`
- 8.367.2.18    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal`
- 8.367.2.19    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedUnc`
- 8.367.2.20    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedVertical`
- 8.367.2.21    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc`
- 8.367.2.22    `loc_svUsedforFix*` `unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix`
- 8.367.2.23    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pTechnologyMask`
- 8.367.2.24    `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pTimeSrc`

8.367.2.25 uint64\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pTimestampUtc

8.367.2.26 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pTimeUnc

8.367.2.27 uint8\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pVertConfidence

8.367.2.28 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pVertReliability

8.367.2.29 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pVertUnc

8.367.2.30 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pXid

8.367.2.31 uint32\_t unpack\_loc\_BestAvailPos\_Ind\_t::status

8.367.2.32 uint16\_t unpack\_loc\_BestAvailPos\_Ind\_t::Tlvresult

## 8.368 unpack\_loc\_Delete\_Assist\_Data\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.368.1 Detailed Description

This structure contains LOC delete assist data unpack

#### Parameters

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

### 8.368.2 Field Documentation

8.368.2.1 uint16\_t unpack\_loc\_Delete\_Assist\_Data\_t::Tlvresult

## 8.369 unpack\_loc\_DeleteAssistData\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [status](#)
- uint16\_t [Tlvresult](#)

### 8.369.1 Detailed Description

Contain the parameters passed for SetLocDeleteAssistDataCallback by the device.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Status of the Delete Assist Data request</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– eQMI_LOC_SUCCESS (0) - Request was completed successfully</li> <li>– eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure</li> <li>– eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported</li> <li>– eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters</li> <li>– eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy</li> <li>– eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline</li> <li>– eQMI_LOC_TIMEOUT (6) - Request failed because it timed out</li> <li>– eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested</li> <li>– eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request</li> <li>– eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed</li> <li>– eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.369.2 Field Documentation

8.369.2.1 uint32\_t unpack\_loc\_DeleteAssistData\_Ind\_t::status

8.369.2.2 uint16\_t unpack\_loc\_DeleteAssistData\_Ind\_t::Tlvresult

## 8.370 unpack\_loc\_EngineState\_Ind\_t Struct Reference

#### Data Fields

- uint32\_t [engineState](#)
- uint16\_t [Tlvresult](#)

### 8.370.1 Detailed Description

This structure contains LOC Engine State field.

#### Parameters



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

## 8.370.2 Field Documentation

8.370.2.1 uint32\_t unpack\_loc\_EngineState\_Ind\_t::engineState

8.370.2.2 uint16\_t unpack\_loc\_EngineState\_Ind\_t::Tlvresult

## 8.371 unpack\_loc\_EventRegister\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.371.1 Detailed Description

This structure contains Event Register unpack

#### Parameters

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

## 8.371.2 Field Documentation

8.371.2.1 uint16\_t unpack\_loc\_EventRegister\_t::Tlvresult

## 8.372 unpack\_loc\_GnssSvInfo\_Ind\_t Struct Reference

### Data Fields

- uint8\_t [altitudeAssumed](#)
- [loc\\_satelliteInfo](#) \* [pSatelliteInfo](#)
- uint16\_t [Tlvresult](#)

### 8.372.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

#### Parameters

<i>altitudeAssumed</i>	<ul style="list-style-type: none"> <li>Indicates whether altitude is assumed or calculated             <ul style="list-style-type: none"> <li>0x00 (FALSE) - Valid altitude is calculated</li> <li>0x01 (TRUE) - Valid altitude is assumed; there may not be enough satellites to determine precise altitude</li> </ul> </li> </ul>
<i>pSatelliteInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">loc_satelliteInfo</a> for more information.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack result</li> </ul>

### 8.372.2 Field Documentation

8.372.2.1 `uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed`

8.372.2.2 `loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo`

8.372.2.3 `uint16_t unpack_loc_GnssSvInfo_Ind_t::Tlvresult`

## 8.373 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](#)

### 8.373.1 Detailed Description

This structure contains Event Position Report Indication unpack

#### Parameters

<i>sessionStatus</i>	<ul style="list-style-type: none"> <li>• Values             <ul style="list-style-type: none"> <li>– 0 - Session was successful</li> <li>– 1 - Session is still in progress; further position reports will be generated until either the fix criteria specified by the client are met or the client response timeout occurs.</li> <li>– 2 - Session failed..</li> <li>– 3 - Fix request failed because the session timed out.</li> <li>– 4 - Fix request failed because the session was ended by the user.</li> <li>– 5 - Fix request failed due to bad parameters in the request.</li> <li>– 6 - Fix request failed because the phone is offline.</li> <li>– 7 - Fix request failed because the engine is locked</li> </ul> </li> </ul>
<i>sessionId</i>	<ul style="list-style-type: none"> <li>• ID of the session that was specified in the Start request</li> <li>• Range - 0 to 255</li> </ul>
<i>pLatitude</i>	<ul style="list-style-type: none"> <li>• Latitude (specified in WGS84 datum)</li> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -90.0 to 90.0</li> <li>• Positive values indicate northern latitude</li> <li>• Negative values indicate southern latitude</li> </ul>
<i>pLongitude</i>	<ul style="list-style-type: none"> <li>• Longitude (specified in WGS84 datum)</li> <li>• Type - Floating point</li> <li>• Units - Degrees</li> <li>• Range - -180.0 to 180.0</li> <li>• Positive values indicate eastern latitude</li> <li>• Negative values indicate western latitude</li> </ul>

<i>pHorUncCircular</i>	<ul style="list-style-type: none"> <li>Horizontal position uncertainty.</li> <li>Units - Meters</li> </ul>
<i>pHorUnc</i> ↔ <i>EllipseSemi</i> ↔ <i>Minor</i>	<ul style="list-style-type: none"> <li>Semi-minor axis of horizontal elliptical uncertainty.</li> <li>Units - Meters</li> </ul>
<i>pHorUnc</i> ↔ <i>EllipseSemi</i> ↔ <i>Major</i>	<ul style="list-style-type: none"> <li>Semi-major axis of horizontal elliptical uncertainty.</li> <li>Units: Meters</li> </ul>
<i>pHorUnc</i> ↔ <i>EllipseOrient</i> ↔ <i>Azimuth</i>	<ul style="list-style-type: none"> <li>Elliptical horizontal uncertainty azimuth of orientation.</li> <li>Units - Decimal degrees</li> <li>Range - 0 to 180</li> </ul>
<i>pHorConfidence</i>	<ul style="list-style-type: none"> <li>Horizontal uncertainty confidence.</li> <li>If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty.</li> <li>Units - Percentage</li> <li>Range 0-99</li> </ul>
<i>pHorReliability</i>	<ul style="list-style-type: none"> <li>Values <ul style="list-style-type: none"> <li>0 - Location reliability is not set.</li> <li>1 - Location reliability is very low; use it at your own risk</li> <li>2 - Location reliability is low; little or no cross-checking is possible.</li> <li>3 - Location reliability is medium; limited cross-check passed</li> <li>4 - Location reliability is high; strong cross-check passed</li> </ul> </li> </ul>
<i>pSpeed</i> ↔ <i>Horizontal</i>	<ul style="list-style-type: none"> <li>Horizontal speed.</li> <li>Units - Meters/second</li> </ul>
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> <li>3-D Speed uncertainty.</li> <li>Units - Meters/second.</li> </ul>
<i>pAltitudeWrt</i> ↔ <i>Ellipsoid</i>	<ul style="list-style-type: none"> <li>Altitude With Respect to WGS84 Ellipsoid.</li> <li>Units - Meters</li> <li>Range -500 to 15883</li> </ul>
<i>pAltitudeWrt</i> ↔ <i>MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>Altitude With Respect to Sea Level.</li> <li>Units - Meters</li> </ul>

<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty.</li> <li>• Units - Meters</li> </ul>
<i>pVertConfidence</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty confidence.</li> <li>• Units - Percentage</li> <li>• Range 0 to 99</li> </ul>
<i>pVertReliability</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Location reliability is not set.</li> <li>– 1 - Location reliability is very low; use it at your own risk.</li> <li>– 2 - Location reliability is low; little or no cross-checking is possible</li> <li>– 3 - Location reliability is medium; limited cross-check passed</li> <li>– 4 - Location reliability is high; strong cross-check passed</li> </ul> </li> </ul>
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> <li>• Vertical speed.</li> <li>• Units - Meters/second</li> </ul>
<i>pHeading</i>	<ul style="list-style-type: none"> <li>• Heading.</li> <li>• Units - Degree</li> <li>• Range 0 to 359.999</li> </ul>
<i>pHeadingUnc</i>	<ul style="list-style-type: none"> <li>• Heading uncertainty.</li> <li>• Units - Degree</li> <li>• Range 0 to 359.999</li> </ul>
<i>pMagnetic↔ Deviation</i>	<ul style="list-style-type: none"> <li>• Difference between the bearing to true north and the bearing shown on a magnetic compass. The deviation is positive when the magnetic north is east of true north.</li> </ul>
<i>pTechnology↔ Mask</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0x00000001 - Satellites were used to generate the fix</li> <li>– 0x00000002 - Cell towers were used to generate the fix</li> <li>– 0x00000004 - Wi-Fi access points were used to generate the fix</li> <li>– 0x00000008 - Sensors were used to generate the fix</li> <li>– 0x00000010 - Reference Location was used to generate the fix</li> <li>– 0x00000020 - Coarse position injected into the location engine was used to generate the fix</li> <li>– 0x00000040 - AFLT was used to generate the fix</li> <li>– 0x00000080 - GNSS and network-provided measurements were used to generate the fix</li> </ul> </li> </ul>
<i>-pPrecision↔ Dilution</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_precisionDilution</a> for more information</li> </ul>

<i>pTimestampUtc</i>	<ul style="list-style-type: none"> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> </ul>
<i>pLeapSeconds</i>	<ul style="list-style-type: none"> <li>• Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds.</li> <li>• Units - Seconds</li> </ul>
<i>-pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gpsTime</a> for more information</li> </ul>
<i>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</li> </ul>
<i>pTimeSrc</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Invalid time.</li> <li>– 1 - Time is set by the 1X system.</li> <li>– 2 - Time is set by WCDMA/GSM time tagging.</li> <li>– 3 - Time is set by an external injection.</li> <li>– 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite.</li> <li>– 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites.</li> <li>– 6 - Both time of the week and the GPS week number are known.</li> <li>– 7 - Time is set by the position engine after the fix is obtained</li> <li>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.</li> <li>– 9 - Time is set after decoding GLO satellites.</li> <li>– 10- Time is set after transforming the GPS to GLO time</li> <li>– 11- Time is set by the sleep time tag provided by the WCDMA network.</li> <li>– 12- Time is set by the sleep time tag provided by the GSM network</li> <li>– 13- Source of the time is unknown</li> <li>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state</li> <li>– 15- Time is set after decoding QZSS satellites.</li> <li>– 16- Time is set after decoding BDS satellites.</li> </ul> </li> </ul>
<i>-pSensorData↔ Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_sensorDataUsage</a> for more information</li> </ul>
<i>pFixId</i>	<ul style="list-style-type: none"> <li>• Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session.</li> </ul>
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_svUsedforFix</a> for more information</li> </ul>
<i>pAltitude↔ Assumed</i>	<ul style="list-style-type: none"> <li>• Indicates whether altitude is assumed or calculated.</li> </ul>

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

### 8.373.2 Field Documentation

8.373.2.1 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pAltitudeAssumed

8.373.2.2 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pAltitudeWrtEllipsoid

8.373.2.3 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pAltitudeWrtMeanSeaLevel

8.373.2.4 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pFixId

8.373.2.5 loc\_gpsTime\* unpack\_loc\_PositionRpt\_Ind\_t::pGpsTime

8.373.2.6 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHeading

8.373.2.7 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHeadingUnc

8.373.2.8 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorConfidence

8.373.2.9 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorReliability

8.373.2.10 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncCircular

8.373.2.11 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncEllipseOrientAzimuth

8.373.2.12 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncEllipseSemiMajor

8.373.2.13 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncEllipseSemiMinor

8.373.2.14 uint64\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pLatitude

8.373.2.15 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pLeapSeconds

8.373.2.16 uint64\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pLongitude

8.373.2.17 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pMagneticDeviation

8.373.2.18 loc\_precisionDilution\* unpack\_loc\_PositionRpt\_Ind\_t::pPrecisionDilution

8.373.2.19 loc\_sensorDataUsage\* unpack\_loc\_PositionRpt\_Ind\_t::pSensorDataUsage

- 8.373.2.20    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal`
- 8.373.2.21    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pSpeedUnc`
- 8.373.2.22    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pSpeedVertical`
- 8.373.2.23    `loc_svUsedforFix*` `unpack_loc_PositionRpt_Ind_t::pSvUsedforFix`
- 8.373.2.24    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pTechnologyMask`
- 8.373.2.25    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pTimeSrc`
- 8.373.2.26    `uint64_t*` `unpack_loc_PositionRpt_Ind_t::pTimestampUtc`
- 8.373.2.27    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pTimeUnc`
- 8.373.2.28    `uint8_t*` `unpack_loc_PositionRpt_Ind_t::pVertConfidence`
- 8.373.2.29    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pVertReliability`
- 8.373.2.30    `uint32_t*` `unpack_loc_PositionRpt_Ind_t::pVertUnc`
- 8.373.2.31    `uint8_t` `unpack_loc_PositionRpt_Ind_t::sessionId`
- 8.373.2.32    `uint32_t` `unpack_loc_PositionRpt_Ind_t::sessionStatus`
- 8.373.2.33    `uint16_t` `unpack_loc_PositionRpt_Ind_t::Tlvresult`

## 8.374    `unpack_loc_SetExtPowerConfig_Ind_t` Struct Reference

### Data Fields

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



### 8.374.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

#### Parameters

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

### 8.374.2 Field Documentation

8.374.2.1 uint32\_t unpack\_loc\_SetExtPowerConfig\_Ind\_t::status

8.374.2.2 uint16\_t unpack\_loc\_SetExtPowerConfig\_Ind\_t::Tlvresult

## 8.375 unpack\_loc\_SetExtPowerState\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.375.1 Detailed Description

This structure contains Set Ext Power State unpack

#### Parameters

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

### 8.375.2 Field Documentation

8.375.2.1 uint16\_t unpack\_loc\_SetExtPowerState\_t::Tlvresult

## 8.376 unpack\_loc\_SetOperationMode\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [status](#)
- uint16\_t [Tlvresult](#)

### 8.376.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

#### Parameters

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

### 8.376.2 Field Documentation

8.376.2.1 uint32\_t unpack\_loc\_SetOperationMode\_Ind\_t::status

8.376.2.2 uint16\_t unpack\_loc\_SetOperationMode\_Ind\_t::Tlvresult

## 8.377 unpack\_loc\_SetOperationMode\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.377.1 Detailed Description

This structure contains Set Operation Mode unpack

#### Parameters

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

### 8.377.2 Field Documentation

8.377.2.1 uint16\_t unpack\_loc\_SetOperationMode\_t::Tlvresult

## 8.378 unpack\_loc\_SLQSLOCGetBestAvailPos\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.378.1 Detailed Description

This structure contains Set Operation Mode unpack

#### Parameters

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

### 8.378.2 Field Documentation

8.378.2.1 uint16\_t unpack\_loc\_SLQSLOCGetBestAvailPos\_t::Tlvresult

## 8.379 unpack\_loc\_Start\_t Struct Reference

#### Data Fields

- uint16\_t [Tlvresult](#)

### 8.379.1 Detailed Description

This structure contains Start LOC unpack

#### Parameters

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

## 8.379.2 Field Documentation

8.379.2.1 uint16\_t unpack\_loc\_Start\_t::Tlvresult

## 8.380 unpack\_loc\_Stop\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.380.1 Detailed Description

This structure contains Stop LOC unpack

#### Parameters

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

## 8.380.2 Field Documentation

8.380.2.1 uint16\_t unpack\_loc\_Stop\_t::Tlvresult

## 8.381 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](#)

### 8.381.1 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

### 8.381.2 Field Documentation

8.381.2.1 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Application

8.381.2.2 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Broadcast

8.381.2.3 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::CustomSCP

8.381.2.4 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::ForceRev0

8.381.2.5 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Protocol

8.381.2.6 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegForeignNID

8.381.2.7 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegForeignSID

8.381.2.8 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegHomeSID

8.381.2.9 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Roaming

8.381.2.10 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::SCI

8.381.2.11 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::SCM

## 8.382 unpack\_nas\_GetHomeNetwork\_t Struct Reference

#### Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- char [name](#) [255]
- uint16\_t [sid](#)
- uint16\_t [nid](#)

### 8.382.1 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

## 8.382.2 Field Documentation

8.382.2.1 uint16\_t unpack\_nas\_GetHomeNetwork\_t::mcc

8.382.2.2 uint16\_t unpack\_nas\_GetHomeNetwork\_t::mnc

8.382.2.3 char unpack\_nas\_GetHomeNetwork\_t::name[255]

8.382.2.4 uint16\_t unpack\_nas\_GetHomeNetwork\_t::nid

8.382.2.5 uint16\_t unpack\_nas\_GetHomeNetwork\_t::sid

## 8.383 unpack\_nas\_GetNetworkPreference\_t Struct Reference

### Data Fields

- uint32\_t [ActiveTechPref](#)
- uint32\_t [Duration](#)
- uint32\_t [PersistentTechPref](#)
- uint16\_t [Tlvresult](#)

### 8.383.1 Detailed Description

#### Parameters

<i>Technology</i> ↔ <i>Pref[OUT]</i>	<ul style="list-style-type: none"> <li>• Bitmask representing the radio technology preference set.</li> <li>• No bits set indicates to the device to automatically determine the technology to use</li> <li>• Values: <ul style="list-style-type: none"> <li>– Bit 0 - Technology is 3GPP2</li> <li>– Bit 1 - Technology is 3GPP</li> </ul> </li> <li>• Any combination of the following may be returned: <ul style="list-style-type: none"> <li>– Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP</li> <li>– Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP</li> <li>– Bit 4 - HDR</li> <li>– Bit 5 - LTE</li> <li>– Bits 6 to 15 - Reserved</li> </ul> </li> </ul>
---	--

<i>Duration</i> [OUT]	<ul style="list-style-type: none"> <li>Duration of active preference <ul style="list-style-type: none"> <li>0 - Permanent</li> <li>1 - Power cycle</li> <li>2 - Until the end of the next call or a power cycle</li> <li>3 - Until the end of the next call, a specified time, or a power cycle</li> <li>4 to 6 - Until the end of the next call</li> </ul> </li> </ul>
<i>Persistent</i> ↔ <i>Technology</i> ↔ <i>Pref</i> [OUT]	<ul style="list-style-type: none"> <li>Bit field representing persistent radio technology preference <ul style="list-style-type: none"> <li>Same representation as the pTechnologyPref parameter</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack result</li> </ul>

## 8.383.2 Field Documentation

8.383.2.1 uint32\_t unpack\_nas\_GetNetworkPreference\_t::ActiveTechPref

8.383.2.2 uint32\_t unpack\_nas\_GetNetworkPreference\_t::Duration

8.383.2.3 uint32\_t unpack\_nas\_GetNetworkPreference\_t::PersistentTechPref

8.383.2.4 uint16\_t unpack\_nas\_GetNetworkPreference\_t::Tlvresult

## 8.384 unpack\_nas\_GetRFInfo\_t Struct Reference

### Data Fields

- uint8\_t [instancesSize](#)
- [RFBandInfoElements](#) [RFBandInfoElements](#) [255]

## 8.384.1 Detailed Description

### Parameters

<i>instancesSize</i>	number of elements in RF info instances array.
<i><a href="#">RFBandInfo</a></i> ↔ <i><a href="#">Elements</a></i>	RF info instances array

## 8.384.2 Field Documentation

8.384.2.1 uint8\_t unpack\_nas\_GetRFInfo\_t::instancesSize

8.384.2.2 RFBandInfoElements unpack\_nas\_GetRFInfo\_t::RFBandInfoElements[255]

## 8.385 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]

### 8.385.1 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

### 8.385.2 Field Documentation

8.385.2.1 uint32\_t unpack\_nas\_GetServingNetwork\_t::CSDomain

8.385.2.2 uint8\_t unpack\_nas\_GetServingNetwork\_t::DataCaps[255]

8.385.2.3 uint8\_t unpack\_nas\_GetServingNetwork\_t::DataCapsLen

8.385.2.4 uint16\_t unpack\_nas\_GetServingNetwork\_t::MCC



- 8.385.2.5    uint16\_t unpack\_nas\_GetServingNetwork\_t::MNC
- 8.385.2.6    uint8\_t unpack\_nas\_GetServingNetwork\_t::Name[255]
- 8.385.2.7    uint8\_t unpack\_nas\_GetServingNetwork\_t::nameSize
- 8.385.2.8    uint32\_t unpack\_nas\_GetServingNetwork\_t::PSDomain
- 8.385.2.9    uint8\_t unpack\_nas\_GetServingNetwork\_t::Radiolfaces[255]
- 8.385.2.10    uint8\_t unpack\_nas\_GetServingNetwork\_t::RadiolfacesSize
- 8.385.2.11    uint32\_t unpack\_nas\_GetServingNetwork\_t::RAN
- 8.385.2.12    uint32\_t unpack\_nas\_GetServingNetwork\_t::RegistrationState
- 8.385.2.13    uint32\_t unpack\_nas\_GetServingNetwork\_t::Roaming

## 8.386    unpack\_nas\_GetServingNetworkCapabilities\_t Struct Reference

### Data Fields

- uint8\_t [DataCapsLen](#)
- uint8\_t [DataCaps](#) [255]

### 8.386.1    Detailed Description

#### Parameters

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

### 8.386.2    Field Documentation

- 8.386.2.1    uint8\_t unpack\_nas\_GetServingNetworkCapabilities\_t::DataCaps[255]
- 8.386.2.2    uint8\_t unpack\_nas\_GetServingNetworkCapabilities\_t::DataCapsLen

## 8.387    unpack\_nas\_GetSignalStrengths\_t Struct Reference

### Data Fields

- uint32\_t [len](#)
- signed char [rssi](#) [8]
- uint32\_t [radio](#) [8]

### 8.387.1 Detailed Description

#### Parameters

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

### 8.387.2 Field Documentation

8.387.2.1 `uint32_t unpack_nas_GetSignalStrengths_t::len`

8.387.2.2 `uint32_t unpack_nas_GetSignalStrengths_t::radio[8]`

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

## 8.388 `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`

### 8.388.1 Detailed Description

#### Parameters

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

### 8.388.2 Field Documentation

8.388.2.1 `nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances`

8.388.2.2 `uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize`

8.388.2.3 `nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance`

8.388.2.4 `uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize`

8.388.2.5 `nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATInstance`

8.388.2.6 uint8\_t\* unpack\_nas\_PerformNetworkScan\_t::pRATInstanceSize

8.388.2.7 uint32\_t\* unpack\_nas\_PerformNetworkScan\_t::pScanResult

## 8.389 unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t Struct Reference

### Data Fields

- uint8\_t [dataCapsSize](#)
- uint8\_t [dataCaps](#) [255]

### 8.389.1 Detailed Description

#### Parameters

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

### 8.389.2 Field Documentation

8.389.2.1 uint8\_t unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t::dataCaps[255]

8.389.2.2 uint8\_t unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t::dataCapsSize

## 8.390 unpack\_nas\_SetEventReportInd\_t Struct Reference

### Data Fields

- [nas\\_SignalStrengthTlv](#) SSTlv
- [nas\\_RFInfoTlv](#) RFTlv
- [nas\\_RejectReasonTlv](#) RRTlv
- [nas\\_SLQSSignalStrengthsTlv](#) SLQSSSTlv

### 8.390.1 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

### 8.390.2 Field Documentation

8.390.2.1 nas\_RFInfoTlv unpack\_nas\_SetEventReportInd\_t::RFTlv

8.390.2.2 `nas_RejectReasonTlv` `unpack_nas_SetEventReportInd_t::RRTlv`

8.390.2.3 `nas_SLQSSignalStrengthsTlv` `unpack_nas_SetEventReportInd_t::SLQSSSTlv`

8.390.2.4 `nas_SignalStrengthTlv` `unpack_nas_SetEventReportInd_t::SSTlv`

## 8.391 `unpack_nas_SetNasLTECphyCalndCallback_ind_t` Struct Reference

### Data Fields

- [nas\\_PhyCaAggScellIndType](#) `sPhyCaAggScellIndType`
- [nas\\_PhyCaAggScellDIBw](#) `sPhyCaAggScellDIBw`
- [nas\\_PhyCaAggScellInfo](#) `sPhyCaAggScellInfo`
- [nas\\_PhyCaAggPcellInfo](#) `sPhyCaAggPcellInfo`
- [nas\\_PhyCaAggScellIndex](#) `sPhyCaAggScellIndex`

### 8.391.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

#### Parameters

<i>pPhyCaAggScellIndType</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIndType</a> for more information.</li> </ul>
<i>sPhyCaAggScellDIBw</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellDIBw</a> for more information.</li> </ul>
<i>sPhyCaAggScellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellInfo</a> for more information.</li> </ul>
<i>sPhyCaAggPcellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggPcellInfo</a> for more information.</li> </ul>
<i>sPhyCaAggScellIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIndex</a> for more information.</li> </ul>

### 8.391.2 Field Documentation

8.391.2.1 `nas_PhyCaAggPcellInfo` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggPcellInfo`

8.391.2.2 `nas_PhyCaAggScellDIBw` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellDIBw`

8.391.2.3 `nas_PhyCaAggScellIndex` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndex`

8.391.2.4 `nas_PhyCaAggScellIndType` `unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndType`

8.391.2.5 nas\_PhyCaAggScellInfo unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t::sPhyCaAggScellInfo

## 8.392 unpack\_nas\_SetNetworkPreference\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.392.1 Detailed Description

#### Parameters

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

### 8.392.2 Field Documentation

8.392.2.1 uint16\_t unpack\_nas\_SetNetworkPreference\_t::Tlvresult

## 8.393 unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t Struct Reference

### Data Fields

- uint8\_t [roaming](#)

### 8.393.1 Detailed Description

#### Parameters

<i>roaming</i>	<ul style="list-style-type: none"> <li>• Roaming Indication <ul style="list-style-type: none"> <li>– 0 - Roaming</li> <li>– 1 - Home</li> <li>– 2 - Roaming partner</li> <li>– &gt;2 - Operator defined values</li> </ul> </li> </ul>
----------------	---

### 8.393.2 Field Documentation

8.393.2.1 `uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming`

## 8.394 `unpack_nas_SetServingSystemCallback_ind_t` Struct Reference

#### Data Fields

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

### 8.394.1 Detailed Description

#### Parameters

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

### 8.394.2 Field Documentation

8.394.2.1 `NAServingSystemInfo unpack_nas_SetServingSystemCallback_ind_t::SSInfo`

8.394.2.2 `uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult`

## 8.395 `unpack_nas_SlqsGetLTECphyCAInfo_t` Struct Reference

## Data Fields

- [NasGetLTECphyCAInfo](#) [LTECphyCAInfo](#)
- [uint16\\_t](#) [Tlvresult](#)

## 8.395.1 Detailed Description

## Parameters

<i>LTECphyCa</i>	<ul style="list-style-type: none"> <li>• Carrier aggregation event information <ul style="list-style-type: none"> <li>– See <a href="#">NasGetLTECphyCAInfo</a> for more details</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

## 8.395.2 Field Documentation

8.395.2.1 [NasGetLTECphyCAInfo](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::LTECphyCAInfo](#)8.395.2.2 [uint16\\_t](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::Tlvresult](#)

## 8.396 unpack\_nas\_SLQSGetNetworkTime\_t Struct Reference

## Data Fields

- [nas\\_timeInfo](#) \* [p3GPP2TimeInfo](#)
- [nas\\_timeInfo](#) \* [p3GPPTimeInfo](#)

## 8.396.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

## Parameters

<i>p3GPP2TimeInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_timeInfo</a> for more information</li> </ul>
<i>p3GPPTimeInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_timeInfo</a> for more information</li> </ul>

## 8.396.2 Field Documentation

8.396.2.1 [nas\\_timeInfo](#)\* [unpack\\_nas\\_SLQSGetNetworkTime\\_t::p3GPP2TimeInfo](#)

8.396.2.2 nas\_timeInfo\* unpack\_nas\_SLQSGetNetworkTime\_t::p3GPPTIMEInfo

## 8.397 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]

### 8.397.1 Field Documentation

- 8.397.1.1 char unpack\_nas\_SLQSGetPLMNName\_t::longName[255]
- 8.397.1.2 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameCI
- 8.397.1.3 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameEn
- 8.397.1.4 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameLen
- 8.397.1.5 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameSB
- 8.397.1.6 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortName[255]
- 8.397.1.7 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortNameCI
- 8.397.1.8 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortNameEn
- 8.397.1.9 char unpack\_nas\_SLQSGetPLMNName\_t::shortNameLen
- 8.397.1.10 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortNameSB
- 8.397.1.11 char unpack\_nas\_SLQSGetPLMNName\_t::spn[255]
- 8.397.1.12 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::spnEncoding
- 8.397.1.13 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::spnLength

## 8.398 unpack\_nas\_SLQSGetServingSystem\_t Struct Reference

### Data Fields

- [nas\\_servSystem](#) ServingSystem



- uint8\_t [RoamIndicatorVal](#)
- [nas\\_dataSrvCapabilities](#) DataSrvCapabilities
- [nas\\_currentPLMN](#) CurrentPLMN
- uint16\_t [SystemID](#)
- uint16\_t [NetworkID](#)
- uint16\_t [BasestationID](#)
- uint32\_t [BasestationLatitude](#)
- uint32\_t [BasestationLongitude](#)
- [nas\\_roamIndList](#) RoamingIndicatorList
- uint8\_t [DefaultRoamInd](#)
- [nas\\_qaQmi3Gpp2TimeZone](#) Gpp2TimeZone
- uint8\_t [CDMA\\_P\\_Rev](#)
- uint8\_t [GppTimeZone](#)
- uint8\_t [GppNetworkDSTAdjustment](#)
- uint16\_t [Lac](#)
- uint32\_t [CellID](#)
- uint8\_t [ConcSvcInfo](#)
- uint8\_t [PRLInd](#)
- uint8\_t [DTMInd](#)
- [nas\\_detailSvcInfo](#) DetailedSvcInfo
- [nas\\_CDMASysInfoExt](#) CDMASystemInfoExt
- uint8\_t [HdrPersonality](#)
- uint16\_t [TrackAreaCode](#)
- [nas\\_callBarStatus](#) CallBarStatus

### 8.398.1 Detailed Description

#### Parameters

<i>ServingSystem</i>	serving system info
<i>RoamIndicator</i> ↔ <i>Val</i>	roaming indicator value
<i>DataSrv</i> ↔ <i>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</i> ↔ <i>Latitude</i>	base station latitude
<i>Basestation</i> ↔ <i>Longitude</i>	base station longitude
<i>Roaming</i> ↔ <i>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>GppNetworkD</i> ↔ <i>STAdjustment</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 service info
<i>CDMASysInfoExt</i>	extra cdma system info
<i>HdrPersonality</i>	hdr personality
<i>TrackAreaCode</i>	track area code
<i>CallBarStatus</i>	call barring status

## 8.398.2 Field Documentation

8.398.2.1 `uint16_t unpack_nas_SLQSGetservingSystem_t::BasestationID`

8.398.2.2 `uint32_t unpack_nas_SLQSGetservingSystem_t::BasestationLatitude`

8.398.2.3 `uint32_t unpack_nas_SLQSGetservingSystem_t::BasestationLongitude`

8.398.2.4 `nas_callBarStatus unpack_nas_SLQSGetservingSystem_t::CallBarStatus`

8.398.2.5 `uint8_t unpack_nas_SLQSGetservingSystem_t::CDMA_P_Rev`

8.398.2.6 `nas_CDMASysInfoExt unpack_nas_SLQSGetservingSystem_t::CDMASystemInfoExt`

8.398.2.7 `uint32_t unpack_nas_SLQSGetservingSystem_t::CellID`

8.398.2.8 `uint8_t unpack_nas_SLQSGetservingSystem_t::ConcSvcInfo`

8.398.2.9 `nas_currentPLMN unpack_nas_SLQSGetservingSystem_t::CurrentPLMN`

8.398.2.10 `nas_dataSrvCapabilities unpack_nas_SLQSGetservingSystem_t::DataSrvCapabilities`

8.398.2.11 `uint8_t unpack_nas_SLQSGetservingSystem_t::DefaultRoamInd`

8.398.2.12 `nas_detailSvcInfo unpack_nas_SLQSGetservingSystem_t::DetailedSvcInfo`

8.398.2.13 `uint8_t unpack_nas_SLQSGetservingSystem_t::DTMInd`

8.398.2.14 `nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetservingSystem_t::Gpp2TimeZone`

8.398.2.15 `uint8_t unpack_nas_SLQSGetservingSystem_t::GppNetworkDSTAdjustment`

8.398.2.16 `uint8_t unpack_nas_SLQSGetservingSystem_t::GppTimeZone`

8.398.2.17 `uint8_t unpack_nas_SLQSGetservingSystem_t::HdrPersonality`

8.398.2.18 `uint16_t unpack_nas_SLQSGetservingSystem_t::Lac`

8.398.2.19 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::NetworkID

8.398.2.20 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::PRLInd

8.398.2.21 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::RoamIndicatorVal

8.398.2.22 nas\_roamIndList unpack\_nas\_SLQSGetServingSystem\_t::RoamingIndicatorList

8.398.2.23 nas\_servSystem unpack\_nas\_SLQSGetServingSystem\_t::ServingSystem

8.398.2.24 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::SystemID

8.398.2.25 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::TrackAreaCode

## 8.399 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](#)

### 8.399.1 Detailed Description

#### Parameters

<i>rxSignal↔ StrengthListLen</i>	number of elements in Receive Signal Strength List
<i>rxSignal↔ StrengthList</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

## 8.399.2 Field Documentation

- 8.399.2.1 `nas_ecioListElement` `unpack_nas_SLQSGetSignalStrength_t::ecioList[18]`
- 8.399.2.2 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::ecioListLen`
- 8.399.2.3 `nas_errorRateListElement` `unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]`
- 8.399.2.4 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::errorRateListLen`
- 8.399.2.5 `int32_t` `unpack_nas_SLQSGetSignalStrength_t::lo`
- 8.399.2.6 `int16_t` `unpack_nas_SLQSGetSignalStrength_t::ltsrpr`
- 8.399.2.7 `int16_t` `unpack_nas_SLQSGetSignalStrength_t::ltsnr`
- 8.399.2.8 `nas_rsrqInformation` `unpack_nas_SLQSGetSignalStrength_t::rsrqInfo`
- 8.399.2.9 `nas_rxSignalStrengthListElement` `unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]`
- 8.399.2.10 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen`
- 8.399.2.11 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask`
- 8.399.2.12 `uint8_t` `unpack_nas_SLQSGetSignalStrength_t::sinr`

## 8.400 `unpack_nas_SLQSGetSysInfo_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`

### 8.400.1 Detailed Description

#### Parameters

<i>pCDMASrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See SrvStatusInfo for more information.</li> </ul>
<i>pHDRSrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See SrvStatusInfo for more information.</li> </ul>
<i>pGSMSrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSrvStatusInfo for more information.</li> </ul>
<i>pWCDMASrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSrvStatusInfo for more information.</li> </ul>
<i>pLTESrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSrvStatusInfo for more information.</li> </ul>
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See CDMASysInfo for more information.</li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See HDRSysInfo for more information.</li> </ul>
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSysInfo for more information.</li> </ul>
<i>pWCDMASys↔ Info</i>	<ul style="list-style-type: none"> <li>• See WCDMASysInfo for more information.</li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See LTESysInfo for more information.</li> </ul>
<i>pAddCDMA↔ SysInfo</i>	<ul style="list-style-type: none"> <li>• See AddCDMASysInfo for more information.</li> </ul>
<i>pAddHDRSys↔ Info</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>
<i>pAddGSMSys↔ Info</i>	<ul style="list-style-type: none"> <li>• See AddSysInfo for more information.</li> </ul>
<i>pAddWCDMA↔ SysInfo</i>	<ul style="list-style-type: none"> <li>• See AddSysInfo for more information.</li> </ul>
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>
<i>pGSMCall↔ BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See CallBarringSysInfo for more information.</li> </ul>

<i>pWCDMA</i> <i>CallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <i>CallBarringSysInfo</i> for more information.</li> </ul>
<i>pLTE</i> <i>VoiceSupportSysInfo</i>	<ul style="list-style-type: none"> <li>• Indicates voice support status on LTE. <ul style="list-style-type: none"> <li>– 0x00 - Voice is not supported</li> <li>– 0x01 - Voice is supported</li> </ul> </li> </ul>
<i>pGSM</i> <i>CipherDomainSysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>
<i>pWCDMA</i> <i>CipherDomainSysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>

## 8.400.2 Field Documentation

8.400.2.1 **nas\_AddCDMASysInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pAddCDMASysInfo*

8.400.2.2 **nas\_AddSysInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pAddGSMSysInfo*

8.400.2.3 **uint16\_t\*** *unpack\_nas\_SLQSGetSysInfo\_t::pAddHDRSysInfo*

8.400.2.4 **uint16\_t\*** *unpack\_nas\_SLQSGetSysInfo\_t::pAddLTESysInfo*

8.400.2.5 **nas\_AddSysInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pAddWCDMASysInfo*

8.400.2.6 **nas\_SrvStatusInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pCDMASrvStatusInfo*

8.400.2.7 **nas\_CDMASysInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pCDMASysInfo*

8.400.2.8 **nas\_CallBarringSysInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pGSMCallBarringSysInfo*

8.400.2.9 **uint8\_t\*** *unpack\_nas\_SLQSGetSysInfo\_t::pGSMCipherDomainSysInfo*

8.400.2.10 **nas\_GSMSrvStatusInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pGSMSrvStatusInfo*

8.400.2.11 **nas\_GSMSysInfo\*** *unpack\_nas\_SLQSGetSysInfo\_t::pGSMSysInfo*

8.400.2.12 nas\_SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pHDRSrvStatusInfo

8.400.2.13 nas\_HDRSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pHDRSysInfo

8.400.2.14 nas\_GSMSrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pLTESrvStatusInfo

8.400.2.15 nas\_LTESysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pLTESysInfo

8.400.2.16 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pLTEVoiceSupportSysInfo

8.400.2.17 nas\_CallBarringSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMACallBarringSysInfo

8.400.2.18 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMACipherDomainSysInfo

8.400.2.19 nas\_GSMSrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMASrvStatusInfo

8.400.2.20 nas\_WCDMASysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMASysInfo

## 8.401 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](#)

### 8.401.1 Detailed Description

#### Parameters

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

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



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

<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating network selection preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic network selection</li> <li>– 0x01 - Manual network selection.</li> </ul> </li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Circuit switched only</li> <li>– 0x01 - Packet switched only</li> <li>– 0x02 - Circuit switched and packet switched</li> <li>– 0x03 - Packet switched attach</li> <li>– 0x04 - Packet switched detach</li> </ul> </li> </ul>
<i>pGWAcqOrderPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Automatic</li> <li>– 0x01 - GSM then WCDMA</li> <li>– 0x02 - WCDMA then GSM</li> </ul> </li> </ul>

## 8.401.2 Field Documentation

8.401.2.1 uint64\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pBandPref

8.401.2.2 uint8\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pEmerMode

8.401.2.3 uint32\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pGWAcqOrderPref

8.401.2.4 uint64\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pLTEBandPref

8.401.2.5 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pModePref

8.401.2.6 uint8\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pNetSelPref

8.401.2.7 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pPRLPref

8.401.2.8 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pRoamPref

8.401.2.9 uint32\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pSrvDomainPref

## 8.402 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](#)

### 8.402.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

#### Parameters

<i>pGERANInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GERANInfo</a> for more information.</li> </ul>
<i>pUMTSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_UMTSInfo</a> for more information.</li> </ul>
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMAInfo</a> for more information.</li> </ul>
<i>pLTEInfo↔ Intrafreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoIntrafreq</a> for more information.</li> </ul>
<i>pLTEInfo↔ Interfreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoInterfreq</a> for more information.</li> </ul>
<i>pLTEInfo↔ NeighboringG↔ SM</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringGSM</a> for more information.</li> </ul>
<i>pLTEInfo↔ NeighboringW↔ CDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringWCDMA</a> for more information.</li> </ul>
<i>pUMTSCellID</i>	<ul style="list-style-type: none"> <li>• Cell ID.</li> <li>• 0xFFFFFFFF indicates cell ID information is not present.</li> </ul>
<i>pWCDMAInfo↔ LTENeighbor↔ Cell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMAInfoLTENeighborCell</a> for more information.</li> </ul>

### 8.402.2 Field Documentation

8.402.2.1 [nas\\_CDMAInfo](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pCDMAInfo](#)

8.402.2.2 [nas\\_GERANInfo](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pGERANInfo](#)

8.402.2.3 `nas_LTEInfoInterfreq*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq`

8.402.2.4 `nas_LTEInfoIntrafreq*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq`

8.402.2.5 `nas_LTEInfoNeighboringGSM*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM`

8.402.2.6 `nas_LTEInfoNeighboringWCDMA*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA`

8.402.2.7 `uint32_t*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID`

8.402.2.8 `nas_UMTSInfo*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo`

8.402.2.9 `nas_WCDMAInfoLTENeighborCell*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell`

## 8.403 `unpack_nas_SLQSNasGetSigInfo_t` Struct Reference

### Data Fields

- [cdmaSSInfo](#) `CDMASSInfo`
- [hdrSSInfo](#) `HDRSSInfo`
- [int8\\_t](#) `GSMSSInfo`
- [cdmaSSInfo](#) `WCDMASSInfo`
- [lteSSInfo](#) `LTESSInfo`

### 8.403.1 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

### 8.403.2 Field Documentation

8.403.2.1 `cdmaSSInfo` `unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo`

8.403.2.2 `int8_t` `unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo`

8.403.2.3 `hdrSSInfo` `unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo`

8.403.2.4 `lteSSInfo` `unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo`

8.403.2.5 `cdmaSSInfo` `unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo`

## 8.404 unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t Struct Reference

### Data Fields

- [nas\\_UniversalTime](#) universalTime
- uint8\_t \* [pTimeZone](#)
- uint8\_t \* [pDayltSavAdj](#)
- uint8\_t \* [pRadioInterface](#)

### 8.404.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

#### Parameters

<i>universalTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_UniversalTime</a> for more information.</li> </ul>
<i>pTimeZone</i>	<ul style="list-style-type: none"> <li>• Time Zone.</li> <li>• Offset from Universal time, i.e., the difference between local time and Universal time, in increments of 15 min (signed value).</li> </ul>
<i>pDayltSavAdj</i>	<ul style="list-style-type: none"> <li>• Daylight Saving Adjustment.</li> <li>• Daylight saving adjustment in hr. <ul style="list-style-type: none"> <li>– Possible values: 0, 1, and 2.</li> </ul> </li> </ul>
<i>pRadioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface from which the information comes</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x04 - NAS_RADIO_IF_GSM - GSM</li> <li>– 0x05 - NAS_RADIO_IF_UMTS - UMTS</li> <li>– 0x08 - NAS_RADIO_IF_LTE - LTE</li> <li>– 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA</li> </ul> </li> </ul>

### 8.404.2 Field Documentation

8.404.2.1 uint8\_t\* unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::pDayltSavAdj

8.404.2.2 uint8\_t\* unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::pRadioInterface

8.404.2.3 uint8\_t\* unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::pTimeZone

8.404.2.4 nas\_UniversalTime unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::universalTime

## 8.405 unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t Struct Reference

### Data Fields

- [cdmaSSInfo](#) \* [pCDMASigInfo](#)
- [hdrSSInfo](#) \* [pHDRSigInfo](#)
- [int8\\_t](#) \* [pGSMSigInfo](#)
- [cdmaSSInfo](#) \* [pWCDMASigInfo](#)
- [lteSSInfo](#) \* [pLTESigInfo](#)
- [int8\\_t](#) \* [pRscp](#)
- [tdscdmaSigInfoExt](#) \* [pTDSCDMASigInfoExt](#)

### 8.405.1 Detailed Description

#### Parameters

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

### 8.405.2 Field Documentation

8.405.2.1 [cdmaSSInfo](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pCDMASigInfo](#)

8.405.2.2 [int8\\_t](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pGSMSigInfo](#)

8.405.2.3 [hdrSSInfo](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pHDRSigInfo](#)

8.405.2.4 [lteSSInfo](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pLTESigInfo](#)

8.405.2.5 [int8\\_t](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pRscp](#)

8.405.2.6 [tdscdmaSigInfoExt](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pTDSCDMASigInfoExt](#)

8.405.2.7 [cdmaSSInfo](#)\* [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pWCDMASigInfo](#)

## 8.406 unpack\_nas\_SLQSNasSwiModemStatus\_t Struct Reference

### Data Fields

- [nas\\_CommInfo](#) [commonInfo](#)
- [nas\\_LTEInfo](#) \* [pLTEInfo](#)

### 8.406.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

#### Parameters

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

### 8.406.2 Field Documentation

8.406.2.1 `nas_CommInfo` `unpack_nas_SLQSNasSwiModemStatus_t::commonInfo`

8.406.2.2 `nas_LTEInfo*` `unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo`

## 8.407 unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t Struct Reference

#### Data Fields

- [NASQmiCbK NasSwiOTAMessageInd Info](#)
- `uint16_t` [Tlvresult](#)

### 8.407.1 Detailed Description

#### Parameters

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

### 8.407.2 Field Documentation

8.407.2.1 `NASQmiCbK NasSwiOTAMessageInd` `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info`

8.407.2.2 `uint16_t` `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult`

## 8.408 unpack\_nas\_SLQSNasTimerCallback\_ind\_t Struct Reference

## Data Fields

- char [t3396\\_apn](#) [101]
- uint8\_t [t3396\\_plmn\\_id](#) [3]
- uint32\_t [t3396\\_val](#)

### 8.408.1 Detailed Description

Structure for Network Timer indication parameters.

#### Parameters

<i>t3396_apn</i>	<ul style="list-style-type: none"> <li>• apn</li> </ul>
<i>t3396_plmn_id</i>	<ul style="list-style-type: none"> <li>• plmn id</li> </ul>
<i>t3396_val</i>	<ul style="list-style-type: none"> <li>• timer value</li> </ul>

### 8.408.2 Field Documentation

8.408.2.1 char [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_t::t3396\\_apn](#)[101]

8.408.2.2 uint8\_t [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_t::t3396\\_plmn\\_id](#)[3]

8.408.2.3 uint32\_t [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_t::t3396\\_val](#)

## 8.409 unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t Struct Reference

## Data Fields

- [NASQmiCbkJasSystemSelPrefInd](#) Info
- uint16\_t [Tlvresult](#)

### 8.409.1 Detailed Description

#### Parameters

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



## 8.409.2 Field Documentation

8.409.2.1 NASQmiCbkNasSystemSelPrefInd unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t::Info

8.409.2.2 uint16\_t unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t::Tlvresult

## 8.410 unpack\_nas\_SLQSSwiGetLteCQI\_t Struct Reference

### Data Fields

- uint8\_t [ValidityCW0](#)
- uint8\_t [CQIValueCW0](#)
- uint8\_t [ValidityCW1](#)
- uint8\_t [CQIValueCW1](#)

### 8.410.1 Detailed Description

#### Parameters

<i>ValidityCW0</i> [O↔ UT]	<ul style="list-style-type: none"> <li>• Values           <ul style="list-style-type: none"> <li>– 0- Invalid.</li> <li>– 1- Valid.</li> </ul> </li> </ul>
<i>CQIValueC↔ W0</i> [OUT]	<ul style="list-style-type: none"> <li>• Values           <ul style="list-style-type: none"> <li>– Range 0~15</li> </ul> </li> </ul>
<i>ValidityCW1</i> [O↔ UT]	<ul style="list-style-type: none"> <li>• Values           <ul style="list-style-type: none"> <li>– 0- Invalid.</li> <li>– 1- Valid.</li> </ul> </li> </ul>
<i>CQIValueC↔ W1</i> [OUT]	<ul style="list-style-type: none"> <li>• Values           <ul style="list-style-type: none"> <li>– Range 0~15</li> </ul> </li> </ul>

## 8.410.2 Field Documentation

8.410.2.1 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::CQIValueCW0

8.410.2.2 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::CQIValueCW1

8.410.2.3 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::ValidityCW0

8.410.2.4 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::ValidityCW1

## 8.411 unpack\_nas\_SLQSSwiGetLteSccRxInfo\_t Struct Reference

### Data Fields

- [nas\\_SccRxInfo](#) \* [pSccRxInfo](#)

### 8.411.1 Detailed Description

#### Parameters

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

### 8.411.2 Field Documentation

8.411.2.1 [nas\\_SccRxInfo](#)\* [unpack\\_nas\\_SLQSSwiGetLteSccRxInfo\\_t::pSccRxInfo](#)

## 8.412 unpack\_nas\_SLQSSysInfoCallback\_ind\_t Struct Reference

### Data Fields

- [nas\\_SrvStatusInfo](#) \* [pCDMASrvStatusInfo](#)
- [nas\\_SrvStatusInfo](#) \* [pHDRSrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pGSMSrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pWCDMASrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pLTESrvStatusInfo](#)
- [nas\\_CDMASysInfo](#) \* [pCDMASysInfo](#)
- [nas\\_HDRSysInfo](#) \* [pHDRSysInfo](#)
- [nas\\_GSMSysInfo](#) \* [pGSMSysInfo](#)
- [nas\\_WCDMASysInfo](#) \* [pWCDMASysInfo](#)
- [nas\\_LTESysInfo](#) \* [pLTESysInfo](#)
- [nas\\_AddCDMASysInfo](#) \* [pAddCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddHDRSysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddGSMSysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddWCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddLTESysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pGSMCallBarringSysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)
- [uint8\\_t](#) \* [pLTEVoiceSupportSysInfo](#)
- [uint8\\_t](#) \* [pGSMCipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pWCDMACipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pSysInfoNoChange](#)

## 8.412.1 Detailed Description

## Parameters

<i>pCDMASrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See SrvStatusInfo for more information.</li> </ul>
<i>pHDRSrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See SrvStatusInfo for more information.</li> </ul>
<i>pGSMSrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSrvStatusInfo for more information.</li> </ul>
<i>pWCDMASrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSrvStatusInfo for more information.</li> </ul>
<i>pLTESrv↔ StatusInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSrvStatusInfo for more information.</li> </ul>
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See CDMASysInfo for more information.</li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See HDRSysInfo for more information.</li> </ul>
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See GSMSysInfo for more information.</li> </ul>
<i>pWCDMASys↔ Info</i>	<ul style="list-style-type: none"> <li>• See WCDMASysInfo for more information.</li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See LTESysInfo for more information.</li> </ul>
<i>pAddCDMA↔ SysInfo</i>	<ul style="list-style-type: none"> <li>• See AddCDMASysInfo for more information.</li> </ul>
<i>pAddHDRSys↔ Info</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>
<i>pAddGSMSys↔ Info</i>	<ul style="list-style-type: none"> <li>• See AddSysInfo for more information.</li> </ul>
<i>pAddWCDMA↔ SysInfo</i>	<ul style="list-style-type: none"> <li>• See AddSysInfo for more information.</li> </ul>
<i>pAddLTESysInfo</i>	<ul style="list-style-type: none"> <li>• System table index referencing the beginning of the geo in which the current serving system is present.</li> <li>• When the system index is not known, 0xFFFF is used.</li> </ul>
<i>pGSMCall↔ BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See CallBarringSysInfo for more information.</li> </ul>

<i>pWCDMA</i> <i>CallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <i>CallBarringSysInfo</i> for more information.</li> </ul>
<i>pLTEVoice</i> <i>SupportSysInfo</i>	<ul style="list-style-type: none"> <li>• Indicates voice support status on LTE. <ul style="list-style-type: none"> <li>– 0x00 - Voice is not supported</li> <li>– 0x01 - Voice is supported</li> </ul> </li> </ul>
<i>pGSMCipher</i> <i>DomainSysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>
<i>pWCDMA</i> <i>CipherDomainSysInfo</i>	<ul style="list-style-type: none"> <li>• Ciphering on the service domain. <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Circuit-switched only</li> <li>– 0x02 - Packet-switched only</li> <li>– 0x03 - Circuit-switched and packet-switched</li> </ul> </li> </ul>
<i>pSysInfoNo</i> <i>Change</i>	<ul style="list-style-type: none"> <li>• System Info No Change.</li> <li>• Flag used to notify clients that a request to select a network ended with no change in the PLMN. <ul style="list-style-type: none"> <li>– 0x01 - No change in system information</li> </ul> </li> </ul>

## 8.412.2 Field Documentation

8.412.2.1 **nas\_AddCDMASysInfo\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddCDMASysInfo*

8.412.2.2 **nas\_AddSysInfo\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddGSMSysInfo*

8.412.2.3 **uint16\_t\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddHDRSysInfo*

8.412.2.4 **uint16\_t\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddLTESysInfo*

8.412.2.5 **nas\_AddSysInfo\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddWCDMASysInfo*

8.412.2.6 **nas\_SrvStatusInfo\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pCDMASrvStatusInfo*

8.412.2.7 **nas\_CDMASysInfo\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pCDMASysInfo*

8.412.2.8 **nas\_CallBarringSysInfo\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSMCallBarringSysInfo*

8.412.2.9 **uint8\_t\*** *unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSMCipherDomainSysInfo*

- 8.412.2.10 nas\_GSMsRvStatusInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSMsRvStatusInfo
- 8.412.2.11 nas\_GSMsSysInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSMsSysInfo
- 8.412.2.12 nas\_SrvStatusInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pHDRsRvStatusInfo
- 8.412.2.13 nas\_HDRsSysInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pHDRsSysInfo
- 8.412.2.14 nas\_GSMsRvStatusInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLTESrvStatusInfo
- 8.412.2.15 nas\_LTEsSysInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLTEsSysInfo
- 8.412.2.16 uint8\_t\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLTEVoiceSupportSysInfo
- 8.412.2.17 uint8\_t\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pSysInfoNoChange
- 8.412.2.18 nas\_CallBarringSysInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMA\_CallBarringSysInfo
- 8.412.2.19 uint8\_t\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMACipherDomainSysInfo
- 8.412.2.20 nas\_GSMsRvStatusInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMASrvStatusInfo
- 8.412.2.21 nas\_WCDMASysInfo\* unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMASysInfo

## 8.413 unpack\_omaDmConfigTlv\_t Struct Reference

### Data Fields

- uint8\_t [state](#)
- uint8\_t [userInputReq](#)
- uint16\_t [userInputTimeout](#)
- uint16\_t [alertmsglength](#)
- uint8\_t [alertmsg](#) [256]

### 8.413.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

## Parameters

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

## 8.413.2 Field Documentation

8.413.2.1 `uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]`8.413.2.2 `uint16_t unpack_omaDmConfigTlv_t::alertmsglength`8.413.2.3 `uint8_t unpack_omaDmConfigTlv_t::state`8.413.2.4 `uint8_t unpack_omaDmConfigTlv_t::userInputReq`8.413.2.5 `uint16_t unpack_omaDmConfigTlv_t::userInputTimeout`8.414 `unpack_omaDmFotaTlv_t` Struct Reference

## Data Fields

- `uint8_t state`
- `uint8_t userInputReq`
- `uint16_t userInputTimeout`
- `uint32_t fwdloadsize`
- `uint32_t fwloadComplete`
- `uint16_t updateCompleteStatus`
- `uint8_t severity`
- `uint16_t versionlength`
- `uint8_t version [256]`
- `uint16_t namelength`
- `uint8_t package_name [256]`
- `uint16_t descriptionlength`
- `uint8_t description [256]`
- `uint8_t sessionType`

## 8.414.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

## Parameters

<i>state</i>	<ul style="list-style-type: none"> <li>• 0x01 - No Firmware available</li> <li>• 0x02 - Query Firmware Download</li> <li>• 0x03 - Firmware Downloading</li> <li>• 0x04 - Firmware downloaded</li> <li>• 0x05 - Query Firmware Update</li> <li>• 0x06 - Firmware updating</li> <li>• 0x07 - Firmware updated</li> </ul>
<i>user_input_req</i>	<ul style="list-style-type: none"> <li>- Bit mask of available user inputs</li> <li>• 0x00 - No user input required. Informational indication</li> <li>• 0x01 - Accept</li> <li>• 0x02 - Reject</li> </ul>
<i>user_input</i> ↔ <i>timeout</i>	<ul style="list-style-type: none"> <li>• Timeout for user input in minutes. A value of 0 means no time-out</li> </ul>
<i>fw_dload_size</i>	<ul style="list-style-type: none"> <li>• The size (in bytes) of the firmware update package</li> </ul>
<i>fw_dload</i> ↔ <i>complete</i>	<ul style="list-style-type: none"> <li>• The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment.</li> </ul>
<i>update</i> ↔ <i>complete_status</i>	<ul style="list-style-type: none"> <li>• See table below.</li> </ul>
<i>severity</i>	<ul style="list-style-type: none"> <li>• 0x01 - Mandatory</li> <li>• 0x02 - Optional</li> </ul>
<i>versionlength</i>	<ul style="list-style-type: none"> <li>• Length of FW Version string in bytes</li> </ul>
<i>version</i>	<ul style="list-style-type: none"> <li>• FW Version string in ASCII (Max 256 characters)</li> </ul>
<i>namelength</i>	<ul style="list-style-type: none"> <li>• Length Package Name string in bytes</li> </ul>
<i>package_name</i>	<ul style="list-style-type: none"> <li>• Package Name in UCS2 (Max 256 characters)</li> </ul>
<i>descriptionlength</i>	<ul style="list-style-type: none"> <li>• Length of description in bytes</li> </ul>
<i>description</i>	<ul style="list-style-type: none"> <li>• Description of Update Package in USC2 (Max 256 characters)</li> </ul>

<i>sessionType</i>	<ul style="list-style-type: none"><li>• 0x00 - Client initiated</li><li>• 0x01 - Network initiated</li></ul>
--------------------	--

## 8.414.2 Field Documentation

8.414.2.1 uint8\_t unpack\_omaDmFotaTlv\_t::description[256]

8.414.2.2 uint16\_t unpack\_omaDmFotaTlv\_t::descriptionlength

8.414.2.3 uint32\_t unpack\_omaDmFotaTlv\_t::fwdloadsize

8.414.2.4 uint32\_t unpack\_omaDmFotaTlv\_t::fwloadComplete

8.414.2.5 uint16\_t unpack\_omaDmFotaTlv\_t::namelength

8.414.2.6 uint8\_t unpack\_omaDmFotaTlv\_t::package\_name[256]

8.414.2.7 uint8\_t unpack\_omaDmFotaTlv\_t::sessionType

8.414.2.8 uint8\_t unpack\_omaDmFotaTlv\_t::severity

8.414.2.9 uint8\_t unpack\_omaDmFotaTlv\_t::state

8.414.2.10 uint16\_t unpack\_omaDmFotaTlv\_t::updateCompleteStatus

8.414.2.11 uint8\_t unpack\_omaDmFotaTlv\_t::userInputReq

8.414.2.12 uint16\_t unpack\_omaDmFotaTlv\_t::userInputTimeout

8.414.2.13 uint8\_t unpack\_omaDmFotaTlv\_t::version[256]

8.414.2.14 uint16\_t unpack\_omaDmFotaTlv\_t::versionlength

## 8.415 unpack\_omaDmNotificationsTlv\_t Struct Reference

### Data Fields

- uint8\_t [notification](#)
- uint16\_t [sessionStatus](#)



### 8.415.1 Field Documentation

8.415.1.1 uint8\_t unpack\_omaDmNotificationsTlv\_t::notification

8.415.1.2 uint16\_t unpack\_omaDmNotificationsTlv\_t::sessionStatus

## 8.416 unpack\_qmi\_t Struct Reference

### Data Fields

- enum [msgtype](#) type
- uint16\_t [msgid](#)
- uint16\_t [xid](#)

### 8.416.1 Detailed Description

qmi response context

#### Parameters

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

### 8.416.2 Field Documentation

8.416.2.1 uint16\_t unpack\_qmi\_t::msgid

8.416.2.2 enum [msgtype](#) unpack\_qmi\_t::type

8.416.2.3 uint16\_t unpack\_qmi\_t::xid

## 8.417 unpack\_qos\_dataRate\_t Struct Reference

### Data Fields

- uint32\_t [dataRateMax](#)
- uint32\_t [guaranteedRate](#)

### 8.417.1 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)

### 8.417.2 Field Documentation

8.417.2.1 `uint32_t unpack_qos_dataRate_t::dataRateMax`

8.417.2.2 `uint32_t unpack_qos_dataRate_t::guaranteedRate`

## 8.418 `unpack_qos_IPv4Addr_t` Struct Reference

### Data Fields

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

### 8.418.1 Detailed Description

This structure contains the IPv4 filter address

#### Parameters

<i>addr</i>	IPv4 address
<i>subnetMask</i>	A packet matches if: <ul style="list-style-type: none"> <li>• <math>(addr \text{ and } subnetMask) == (IP \text{ pkt } addr \ \&amp; \ subnetMask)</math> Callers to set up a filter with a range of source addresses, if needed; subnet mask of all 1s (255.255.255.255) specifies a single address value</li> </ul>

### 8.418.2 Field Documentation

8.418.2.1 `uint32_t unpack_qos_IPv4Addr_t::addr`

8.418.2.2 `uint32_t unpack_qos_IPv4Addr_t::subnetMask`

## 8.419 `unpack_qos_IPv6Addr_t` Struct Reference

### Data Fields

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

### 8.419.1 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.

### 8.419.2 Field Documentation

8.419.2.1 `uint8_t unpack_qos_IPv6Addr_t::addr[16]`

8.419.2.2 `uint8_t unpack_qos_IPv6Addr_t::prefixLen`

## 8.420 unpack\_qos\_IPv6TrafCls\_t Struct Reference

### Data Fields

- `uint8_t val`
- `uint8_t mask`

### 8.420.1 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: $(IPv6\_filter\_traffic\_class\_val \text{ and } IPv6\_filter\_traffic\_class\_mask) == (Traffic\ class\ value\ in\ the\ IP\ packet \& IPv6\_filter\_traffic\_class\_mask)$ Example: <ul style="list-style-type: none"> <li>• <code>IPv6_filter_tc_val = 00101000</code></li> <li>• <code>IPv6_filter_tc_mask = 11111100</code> Filter will compare only the first 6 bits in <code>IPv6_filter_traffic_↵</code> class with the first 6 bits in the traffic class field of the IP packet; first 6 bits in the traffic class field of the IP packet must be 001010 to match filter; last 2 bits can be anything, since they are ignored by filtering</li> </ul>

### 8.420.2 Field Documentation

8.420.2.1 `uint8_t unpack_qos_IPv6TrafCls_t::mask`

8.420.2.2 `uint8_t unpack_qos_IPv6TrafCls_t::val`

## 8.421 unpack\_qos\_pktErrRate\_t Struct Reference

### Data Fields

- uint16\_t [multiplier](#)
- uint16\_t [exponent](#)

### 8.421.1 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)

### 8.421.2 Field Documentation

8.421.2.1 uint16\_t unpack\_qos\_pktErrRate\_t::exponent

8.421.2.2 uint16\_t unpack\_qos\_pktErrRate\_t::multiplier

## 8.422 unpack\_qos\_Port\_t Struct Reference

### Data Fields

- uint16\_t [port](#)
- uint16\_t [range](#)

### 8.422.1 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

### 8.422.2 Field Documentation

8.422.2.1 uint16\_t unpack\_qos\_Port\_t::port

8.422.2.2 uint16\_t unpack\_qos\_Port\_t::range

## 8.423 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

### 8.423.1 Detailed Description

Structure with QoS flow details.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

#### Parameters

<i>QFlowState</i>	<ul style="list-style-type: none"> <li>• QoS flow state information, please check <a href="#">unpack_qos_QosFlowInfoState_t</a> for more information</li> </ul>
<i>is_TxQFlowGranted_Available</i>	<ul style="list-style-type: none"> <li>• TRUE if optional TxQFlowGranted is available</li> </ul>
<i>TxQFlowGranted</i>	<ul style="list-style-type: none"> <li>• The Tx Qos flow granted, please check <a href="#">unpack_qos_swiQosFlow_t</a> for more information</li> </ul>
<i>is_RxQFlowGranted_Available</i>	<ul style="list-style-type: none"> <li>• TRUE if optional RxQFlowGranted is available</li> </ul>
<i>RxQFlowGranted</i>	<ul style="list-style-type: none"> <li>• The Rx Qos flow granted, please check <a href="#">unpack_qos_swiQosFlow_t</a> for more information</li> </ul>
<i>NumTxFilters</i>	<ul style="list-style-type: none"> <li>• Number of Tx filters available</li> </ul>
<i>TxQFilter</i>	<ul style="list-style-type: none"> <li>• The Tx Qos filter, please check <a href="#">unpack_qos_swiQosFilter_t</a> for more information</li> <li>• See <a href="#">LIBPACK_MAX_QOS_FILTERS</a> for more information</li> </ul>
<i>NumRxFilters</i>	<ul style="list-style-type: none"> <li>• Number of Tx filters available</li> </ul>
<i>RxQFilter</i>	<ul style="list-style-type: none"> <li>• The Rx Qos filter, please check <a href="#">unpack_qos_swiQosFilter_t</a> for more information</li> <li>• See <a href="#">LIBPACK_MAX_QOS_FILTERS</a> for more information</li> </ul>

<i>BearerID</i>	<ul style="list-style-type: none"> <li>• The bearer ID</li> <li>• Bearer ID or Radio Link Protocol (RLP) ID of the activated flow.</li> <li>• Valid Values - 0 to 16</li> <li>• 0xFF - Invalid value.</li> </ul>
-----------------	--

## 8.423.2 Field Documentation

8.423.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`

8.423.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`

8.423.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`

8.423.2.4 `uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters`

8.423.2.5 `uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters`

8.423.2.6 `unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState`

8.423.2.7 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]`

8.423.2.8 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted`

8.423.2.9 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]`

8.423.2.10 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted`

## 8.424 `unpack_qos_QosFlowInfoState_t` Struct Reference

### Data Fields

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

### 8.424.1 Detailed Description

This structure contains QoS flow state

#### Parameters

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

### 8.424.2 Field Documentation

8.424.2.1 `uint32_t unpack_qos_QosFlowInfoState_t::id`

8.424.2.2 `uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow`

8.424.2.3 `uint8_t unpack_qos_QosFlowInfoState_t::state`

## 8.425 unpack\_qos\_SLQSQosGetNetworkStatus\_t Struct Reference

### Data Fields

- `uint8_t` [NWQoSStatus](#)

### 8.425.1 Detailed Description

Structure that contains the response to get NW QoS status command

#### Parameters

<i>NWQoSStatus</i>	<p>Network QoS support status</p> <ul style="list-style-type: none"><li>• 0 – No QoS support in network</li><li>• 1 – Network supports QoS</li></ul>
--------------------	--

### 8.425.2 Field Documentation

8.425.2.1 `uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus`

## 8.426 `unpack_qos_SLQSQosSwiReadApnExtraParams_t` Struct Reference

### Data Fields

- `uint32_t apnId`
- `uint8_t ambr_ul`
- `uint8_t ambr_dl`
- `uint8_t ambr_ul_ext`
- `uint8_t ambr_dl_ext`
- `uint8_t ambr_ul_ext2`
- `uint8_t ambr_dl_ext2`

### 8.426.1 Detailed Description

Structure that contains extra APN parameters

#### Parameters

<i>apnId</i>	<ul style="list-style-type: none"> <li>• APN id</li> <li>• ID identifying the APN that the client would like to query the AMBR params</li> </ul>
<i>ambr_ul</i>	<ul style="list-style-type: none"> <li>• APN AMBR uplink</li> <li>• APN AMBR uplink values from 1 kbps to 8640 kbps</li> </ul>
<i>ambr_dl</i>	<ul style="list-style-type: none"> <li>• APN AMBR downlink</li> <li>• APN AMBR downlink values from 1 kbps to 8640 kbps</li> </ul>
<i>ambr_ul_ext</i>	<ul style="list-style-type: none"> <li>• Extended APN AMBR uplink</li> <li>• APN AMBR uplink values from 8700 kbps to 256 Mbps</li> </ul>
<i>ambr_dl_ext</i>	<ul style="list-style-type: none"> <li>• Extended APN AMBR downlink</li> <li>• APN AMBR downlink values from 8700 kbps to 256 Mbps</li> </ul>
<i>ambr_ul_ext2</i>	<ul style="list-style-type: none"> <li>• Second extended APN AMBR uplink</li> <li>• APN AMBR uplink values from 256 Mbps to 65280 Mbps</li> </ul>
<i>ambr_dl_ext2</i>	<ul style="list-style-type: none"> <li>• Second extended APN AMBR downlink</li> <li>• APN AMBR downlink values from 256 Mbps to 65280 Mbps</li> </ul>



## 8.426.2 Field Documentation

8.426.2.1 `uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl`

8.426.2.2 `uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext`

8.426.2.3 `uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2`

8.426.2.4 `uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul`

8.426.2.5 `uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext`

8.426.2.6 `uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2`

8.426.2.7 `uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId`

## 8.427 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]

### 8.427.1 Detailed Description

Structure that contains APN data statistics

#### Parameters

<i>apnId</i>	<ul style="list-style-type: none"> <li>• APN id</li> <li>• ID identifying the connected APN that the client would like to query the data statistic for</li> </ul>
<i>total_tx_pkt</i>	<ul style="list-style-type: none"> <li>• sum of all packets sent</li> </ul>
<i>total_tx_pkt_drp</i>	<ul style="list-style-type: none"> <li>• sum of all(TX) packets dropped</li> </ul>
<i>total_rx_pkt</i>	<ul style="list-style-type: none"> <li>• sum of all packets received</li> </ul>

<i>total_tx_bytes</i>	<ul style="list-style-type: none"> <li>• sum of all bytes sent</li> </ul>
<i>total_tx_bytes</i> <sub>↔</sub> <i>_drp</i>	<ul style="list-style-type: none"> <li>• sum of all(TX) bytes dropped</li> </ul>
<i>total_rx_bytes</i>	<ul style="list-style-type: none"> <li>• number of received bytes for the QoS flow ID</li> </ul>
<i>numQosFlow</i>	<ul style="list-style-type: none"> <li>• pointer to number of QoS flow Stat</li> </ul>
<i>qosFlow</i> [ <i>LIBP</i> <sub>↔</sub> <i>ACK_MAX_Q</i> <sub>↔</sub> <i>OS_FLOW_P</i> <sub>↔</sub> <i>ER_APN_STA</i> <sub>↔</sub> <i>TS</i> ]	<ul style="list-style-type: none"> <li>• Data statistic per QoS flow</li> <li>• See <a href="#">unpack_QosFlowStat_t</a> for more information</li> <li>• See <a href="#">LIBPACK_MAX_QOS_FLOW_PER_APN_STATS</a> for more information</li> </ul>

## 8.427.2 Field Documentation

8.427.2.1 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId`

8.427.2.2 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow`

8.427.2.3 `unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]`

8.427.2.4 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes`

8.427.2.5 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt`

8.427.2.6 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes`

8.427.2.7 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp`

8.427.2.8 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt`

8.427.2.9 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp`

## 8.428 `unpack_qos_SLQSSetQosEventCallback_ind_t` Struct Reference

### Data Fields

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

### 8.428.1 Detailed Description

Structure with QoS event details

#### Parameters

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

## 8.428.2 Field Documentation

8.428.2.1 `uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows`

8.428.2.2 `unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]`

## 8.429 unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t Struct Reference

### Data Fields

- `uint8_t status`

### 8.429.1 Detailed Description

Structure with network's QoS status

#### Parameters

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

#### Note

- Technology Supported: CDMA

## 8.429.2 Field Documentation

8.429.2.1 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

## 8.430 unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t Struct Reference

### Data Fields

- `uint16_t event`

### 8.430.1 Detailed Description

Structure with QoS primary flow events

#### Parameters

<i>event</i>	Event which causes this indication: <ul style="list-style-type: none"> <li>• 0x0001 – Primary flow QoS modify operation success</li> <li>• 0x0002 – Primary flow QoS modify operation failure</li> </ul>
--------------	--

### 8.430.2 Field Documentation

8.430.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

## 8.431 `unpack_qos_SLQSSetQosStatusCallback_ind_t` Struct Reference

#### Data Fields

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

### 8.431.1 Detailed Description

Structure with QoS status indication details

#### Parameters

<i>id</i>	<ul style="list-style-type: none"> <li>• Index identifying the QoS flow whose status is being reported</li> </ul>
<i>status</i>	Current QoS flow status: <ul style="list-style-type: none"> <li>• 0x01 – QMI_QOS_STATUS_ACTIVATED</li> <li>• 0x02 – QMI_QOS_STATUS_SUSPENDED</li> <li>• 0x03 – QMI_QOS_STATUS_GONE</li> </ul>
<i>event</i>	<ul style="list-style-type: none"> <li>• 0x01 – QMI_QOS_ACTIVATED_EV</li> <li>• 0x02 – QMI_QOS_SUSPENDED_EV</li> <li>• 0x03 – QMI_QOS_GONE_EV</li> <li>• 0x04 – QMI_QOS_MODIFY_ACCEPTED_EV</li> <li>• 0x05 – QMI_QOS_MODIFY_REJECTED_EV</li> <li>• 0x06 – QMI_QOS_INFO_CODE_UPDATED_EV</li> </ul>

<i>reason</i>	<ul style="list-style-type: none"> <li>• 0x01 - QMI_QOS_INVALID_PARAMS</li> <li>• 0x02 - QMI_QOS_INTERNAL_CALL_ENDED</li> <li>• 0x03 - QMI_QOS_INTERNAL_ERROR</li> <li>• 0x04 - QMI_QOS_INSUFFICIENT_LOCAL_Resources</li> <li>• 0x05 - QMI_QOS_TIMED_OUT_OPERATION</li> <li>• 0x06 - QMI_QOS_INTERNAL_UNKNOWN_CAUSE_CODE</li> <li>• 0x07 - QMI_QOS_INTERNAL_MODIFY_IN_PROGRESS</li> <li>• 0x08 - QMI_QOS_NOT_SUPPORTED</li> <li>• 0x09 - QMI_QOS_NOT_AVAILABLE</li> <li>• 0x0A - QMI_QOS_NOT_GUARANTEED</li> <li>• 0x0B - QMI_QOS_INSUFFICIENT_NETWORK_RESOURCES</li> <li>• 0x0C - QMI_QOS_AWARE_SYSTEM</li> <li>• 0x0D - QMI_QOS_UNAWARE_SYSTEM</li> <li>• 0x0E - QOS_REJECTED_OPERATION</li> <li>• 0x0F - QMI_QOS_WILL_GRANT_WHEN_QOS_RESUMED</li> <li>• 0x10 - QMI_QOS_NETWORK_CALL_ENDED</li> <li>• 0x11 - QMI_QOS_NETWORK_SERVICE_NOT_AVAILABLE</li> <li>• 0x12 - QMI_QOS_NETWORK_L2_LINK_RELEASED</li> <li>• 0x13 - QMI_QOS_NETWORK_L2_LINK_REESTAB_REJ</li> <li>• 0x14 - QMI_QOS_NETWORK_L2_LINK_REESTAB_IND</li> <li>• 0x15 - QMI_QOS_NETWORK_UNKNOWN_CAUSE_CODE</li> <li>• 0x16 - QMI_NETWORK_BUSY</li> </ul>
---------------	--

### 8.431.2 Field Documentation

8.431.2.1 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event`

8.431.2.2 `uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id`

8.431.2.3 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason`

8.431.2.4 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status`

## 8.432 unpack\_qos\_swiQosFilter\_t Struct Reference

### Data Fields

- `uint8_t index`
- `uint8_t version`
- `uint8_t is_IPv4SrcAddr_Available`
- `unpack_qos_IPv4Addr_t IPv4SrcAddr`
- `uint8_t is_IPv4DstAddr_Available`
- `unpack_qos_IPv4Addr_t IPv4DstAddr`
- `uint8_t is_NxtHdrProto_Available`
- `uint8_t NxtHdrProto`

- [uint8\\_t is\\_IPv4Tos\\_Available](#)
- [unpack\\_qos\\_Tos\\_t IPv4Tos](#)
- [uint8\\_t is\\_IPv6SrcAddr\\_Available](#)
- [unpack\\_qos\\_IPv6Addr\\_t IPv6SrcAddr](#)
- [uint8\\_t is\\_IPv6DstAddr\\_Available](#)
- [unpack\\_qos\\_IPv6Addr\\_t IPv6DstAddr](#)
- [uint8\\_t is\\_IPv6TrafCls\\_Available](#)
- [unpack\\_qos\\_IPv6TrafCls\\_t IPv6TrafCls](#)
- [uint8\\_t is\\_IPv6Label\\_Available](#)
- [uint32\\_t IPv6Label](#)
- [uint8\\_t is\\_TCPSrcPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t TCPSrcPort](#)
- [uint8\\_t is\\_TCPDstPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t TCPDstPort](#)
- [uint8\\_t is\\_UDPSrcPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t UDPSrcPort](#)
- [uint8\\_t is\\_UDPDstPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t UDPDstPort](#)
- [uint8\\_t is\\_EspSpi\\_Available](#)
- [uint32\\_t EspSpi](#)
- [uint8\\_t is\\_Precedence\\_Available](#)
- [uint16\\_t Precedence](#)
- [uint8\\_t is\\_Id\\_Available](#)
- [uint16\\_t Id](#)
- [uint8\\_t is\\_TransrcPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t TransrcPort](#)
- [uint8\\_t is\\_TransdstPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t TransdstPort](#)

### 8.432.1 Detailed Description

This structure contains the QoS Filter Request.

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

#### Parameters

<i>index</i>	Mandatory parameter IP filter index Integer that uniquely identifies each filter instance This TLV must be present in the request
<i>version</i>	Mandatory parameter IP filter version Identifies whether the filter is associated with IPv4 or IPv6; value specified also implies that only TLVs defined for that IP version, i.e., TLVs with IPv4 or IPv6 in the name, can be specified <ul style="list-style-type: none"> <li>• 0x04 – IPv4</li> <li>• 0x06 – Ipv6</li> </ul>
<i>IPv4SrcAddr</i>	IPv4 filter soruce address See <a href="#">unpack_qos_IPv4Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>IPv4DstAddr</i>	IPv4 filter destination address See <a href="#">unpack_qos_IPv4Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>

<i>NxtHdrProto</i>	IP filter next header protocol This TLV must be present if any non-IP filter TLV(s) are provided If this field is specified, only IP packets belonging to specified higher layer protocol are considered when filtering The following protocols may be specified: <ul style="list-style-type: none"> <li>• 0x01 = ICMP</li> <li>• 0x06 = TCP</li> <li>• 0x11 = UDP</li> <li>• 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP &amp; UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified.</li> </ul>
<i>IPv4Tos</i>	IPv4 filter type of service See <a href="#">unpack_qos_Tos_t</a> for more information
<i>IPv6SrcAddr</i>	IPv6 filter source address See <a href="#">unpack_qos_IPv6Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>IPv6DstAddr</i>	IPv6 filter destination address See <a href="#">unpack_qos_IPv6Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>IPv6TrafCls</i>	IPv6 filter traffic class See <a href="#">unpack_qos_IPv6TrafCls_t</a> for more information
<i>IPv6Label</i>	IPv6 flow label Packet matches the IPv6 flow label filter if: ( *pIPv6Label == flow label in the IPv6 header) <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>TCPsrcPort</i>	TCP filter source port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>TCPdstPort</i>	TCP filter destination port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>UDPSrcPort</i>	UDP filter source port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>UDPdstPort</i>	UDP filter destination port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>EspSpi</i>	ESP filter security policy index Security policy index to uniquely identify each IP flow for filtering encrypted packets for encapsulating security payload <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>Precedence</i>	Filter Precedence Specifies the order in which filters are applied; lower numerical value has higher precedence Note: This TLV only applies to network-initiated QoS; QoS requests containing this TLV from control points will be ignored
<i>Id</i>	Filter ID Unique identifier for each filter; filter ID is assigned by the modem Note: This TLV only applies to network-initiated QoS; QoS requests containing this TLV from control points will be ignored
<i>TranSrcPort</i>	Transport protocol filter source port See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>UDPdstPort</i>	Transport protocol filter destination port See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>

## 8.432.2 Field Documentation

### 8.432.2.1 uint32\_t unpack\_qos\_swiQosFilter\_t::EspSpi

- 8.432.2.2    `uint16_t unpack_qos_swQosFilter_t::ld`
- 8.432.2.3    `uint8_t unpack_qos_swQosFilter_t::index`
- 8.432.2.4    `unpack_qos_IPv4Addr_t unpack_qos_swQosFilter_t::IPv4DstAddr`
- 8.432.2.5    `unpack_qos_IPv4Addr_t unpack_qos_swQosFilter_t::IPv4SrcAddr`
- 8.432.2.6    `unpack_qos_Tos_t unpack_qos_swQosFilter_t::IPv4Tos`
- 8.432.2.7    `unpack_qos_IPv6Addr_t unpack_qos_swQosFilter_t::IPv6DstAddr`
- 8.432.2.8    `uint32_t unpack_qos_swQosFilter_t::IPv6Label`
- 8.432.2.9    `unpack_qos_IPv6Addr_t unpack_qos_swQosFilter_t::IPv6SrcAddr`
- 8.432.2.10    `unpack_qos_IPv6TrafCls_t unpack_qos_swQosFilter_t::IPv6TrafCls`
- 8.432.2.11    `uint8_t unpack_qos_swQosFilter_t::is_EspSpi_Available`
- 8.432.2.12    `uint8_t unpack_qos_swQosFilter_t::is_Id_Available`
- 8.432.2.13    `uint8_t unpack_qos_swQosFilter_t::is_IPv4DstAddr_Available`
- 8.432.2.14    `uint8_t unpack_qos_swQosFilter_t::is_IPv4SrcAddr_Available`
- 8.432.2.15    `uint8_t unpack_qos_swQosFilter_t::is_IPv4Tos_Available`
- 8.432.2.16    `uint8_t unpack_qos_swQosFilter_t::is_IPv6DstAddr_Available`
- 8.432.2.17    `uint8_t unpack_qos_swQosFilter_t::is_IPv6Label_Available`
- 8.432.2.18    `uint8_t unpack_qos_swQosFilter_t::is_IPv6SrcAddr_Available`
- 8.432.2.19    `uint8_t unpack_qos_swQosFilter_t::is_IPv6TrafCls_Available`
- 8.432.2.20    `uint8_t unpack_qos_swQosFilter_t::is_NxtHdrProto_Available`
- 8.432.2.21    `uint8_t unpack_qos_swQosFilter_t::is_Precedence_Available`
- 8.432.2.22    `uint8_t unpack_qos_swQosFilter_t::is_TCPDstPort_Available`
- 8.432.2.23    `uint8_t unpack_qos_swQosFilter_t::is_TCPSrcPort_Available`
- 8.432.2.24    `uint8_t unpack_qos_swQosFilter_t::is_TranDstPort_Available`



- 8.432.2.25 `uint8_t unpack_qos_swiQosFilter_t::is_TransrcPort_Available`
- 8.432.2.26 `uint8_t unpack_qos_swiQosFilter_t::is_UDPdstPort_Available`
- 8.432.2.27 `uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available`
- 8.432.2.28 `uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto`
- 8.432.2.29 `uint16_t unpack_qos_swiQosFilter_t::Precedence`
- 8.432.2.30 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPdstPort`
- 8.432.2.31 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPsrcPort`
- 8.432.2.32 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort`
- 8.432.2.33 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort`
- 8.432.2.34 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPdstPort`
- 8.432.2.35 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort`
- 8.432.2.36 `uint8_t unpack_qos_swiQosFilter_t::version`

## 8.433 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`

### 8.433.1 Detailed Description

This structure contains the QoS Flow Request.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

#### Parameters

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

<i>MinPolicedPktSz</i>	<ul style="list-style-type: none"> <li>• IP flow minimum policed packet size</li> <li>• Integer that defines the minimum packet size (in bytes) that will be policed for QoS guarantees; any IP packets that are smaller than the minimum specified policed size may not receive requested QoS</li> </ul>
<i>MaxAllowedPktSz</i>	<ul style="list-style-type: none"> <li>• IP flow maximum allowed packet size</li> <li>• Integer that defines the maximum packet size (in bytes) allowed in the IP flow; any IP packets greater in size than the maximum allowed packet size are not queued for transmission</li> </ul>
<i>val_3GPPResidualBER</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP residual bit error rate</li> <li>• residual_bit_error_rate</li> <li>• 0 = <math>5 \times 10^{-2}</math> residual BER</li> <li>• 1 = <math>1 \times 10^{-2}</math> residual BER</li> <li>• 2 = <math>5 \times 10^{-3}</math> residual BER</li> <li>• 3 = <math>4 \times 10^{-3}</math> residual BER</li> <li>• 4 = <math>1 \times 10^{-3}</math> residual BER</li> <li>• 5 = <math>1 \times 10^{-4}</math> residual BER</li> <li>• 6 = <math>1 \times 10^{-5}</math> residual BER</li> <li>• 7 = <math>1 \times 10^{-6}</math> residual BER</li> <li>• 8 = <math>6 \times 10^{-8}</math> residual BER</li> <li>• Integer that indicates the undetected BER for each IP flow in the delivered packets; Applies only to 3GPP networks</li> </ul>
<i>val_3GPPTrafficHandlingPriority</i>	<ul style="list-style-type: none"> <li>• 3GPP traffic handling priority</li> <li>• 0 – Relative traffic handling priority 1</li> <li>• 1 – Relative traffic handling priority 2</li> <li>• 2 – Relative traffic handling priority 3</li> <li>• Defines the relative priority of the flow; applies only to 3GPP networks</li> </ul>
<i>val_3GPPIMCn</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP IM CN flag</li> <li>• IM CN subsystem signaling flag:</li> <li>• 0x00 – FALSE</li> <li>• 0x01 – TRUE</li> <li>• This parameter applies only to 3GPP networks</li> </ul>
<i>val_3GPPSigInd</i>	<ul style="list-style-type: none"> <li>• IP flow 3GPP signaling indication</li> <li>• 0x00 – FALSE</li> <li>• 0x01 – TRUE</li> <li>• This parameter applies only to 3GPP networks</li> </ul>

<i>LteQci</i>	<ul style="list-style-type: none"> <li>• LTE QoS Class Identifier</li> <li>• QoS Class Identifier(QCI) is a required parameter to request QoS in LTE</li> <li>• QCI values: <ul style="list-style-type: none"> <li>– QCI value 0 requests the network to assign the appropriate QCI value</li> <li>– QCI values 1-4 are associated with guaranteed bitrates</li> <li>– QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored</li> </ul> </li> </ul>
---------------	---

### 8.433.2 Field Documentation

8.433.2.1 `unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate`

8.433.2.2 `uint8_t unpack_qos_swiQosFlow_t::index`

8.433.2.3 `uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.433.2.4 `uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.433.2.5 `uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available`

8.433.2.6 `uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.433.2.7 `uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.433.2.8 `uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.433.2.9 `uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.433.2.10 `uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.433.2.11 `uint8_t unpack_qos_swiQosFlow_t::is-TokenBucket_Available`

8.433.2.12 `uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.433.2.13 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.433.2.14 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

8.433.2.15 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available`

8.433.2.16 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available`

8.433.2.17 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available`

- 8.433.2.18 uint32\_t unpack\_qos\_swiQosFlow\_t::Jitter
- 8.433.2.19 uint32\_t unpack\_qos\_swiQosFlow\_t::Latency
- 8.433.2.20 uint8\_t unpack\_qos\_swiQosFlow\_t::LteQci
- 8.433.2.21 uint32\_t unpack\_qos\_swiQosFlow\_t::MaxAllowedPktSz
- 8.433.2.22 uint32\_t unpack\_qos\_swiQosFlow\_t::MinPolicedPktSz
- 8.433.2.23 unpack\_qos\_pktErrRate\_t unpack\_qos\_swiQosFlow\_t::PktErrRate
- 8.433.2.24 uint16\_t unpack\_qos\_swiQosFlow\_t::ProfileId3GPP2
- 8.433.2.25 unpack\_qos\_tokenBucket\_t unpack\_qos\_swiQosFlow\_t::TokenBucket
- 8.433.2.26 uint8\_t unpack\_qos\_swiQosFlow\_t::TrafficClass
- 8.433.2.27 uint8\_t unpack\_qos\_swiQosFlow\_t::val\_3GPP2Pri
- 8.433.2.28 uint8\_t unpack\_qos\_swiQosFlow\_t::val\_3GPPImCn
- 8.433.2.29 uint16\_t unpack\_qos\_swiQosFlow\_t::val\_3GPPResResidualBER
- 8.433.2.30 uint8\_t unpack\_qos\_swiQosFlow\_t::val\_3GPPSigInd
- 8.433.2.31 uint8\_t unpack\_qos\_swiQosFlow\_t::val\_3GPPTraHdlPri

## 8.434 unpack\_qos\_tokenBucket\_t Struct Reference

### Data Fields

- uint32\_t [peakRate](#)
- uint32\_t [tokenRate](#)
- uint32\_t [bucketSz](#)

### 8.434.1 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

### 8.434.2 Field Documentation

8.434.2.1 `uint32_t unpack_qos_tokenBucket_t::bucketSz`

8.434.2.2 `uint32_t unpack_qos_tokenBucket_t::peakRate`

8.434.2.3 `uint32_t unpack_qos_tokenBucket_t::tokenRate`

## 8.435 `unpack_qos_Tos_t` Struct Reference

### Data Fields

- `uint8_t val`
- `uint8_t mask`

### 8.435.1 Detailed Description

This structure contains the IPv4 filter type of service

#### Parameters

<i>val</i>	Type of service value
<i>mask</i>	Packet matches the TOS filter if: $(IPv4\_filter\_tos\_val \text{ and } IPv4\_filter\_tos\_mask) == (\text{TOS value in the IP packet} \text{ \& } IPv4\_filter\_tos\_mask)$ Example: <ul style="list-style-type: none"> <li>• <code>IPv4_filter_tos_val = 00101000</code></li> <li>• <code>IPv4_filter_tos_mask = 11111100</code> 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.</li> </ul>

### 8.435.2 Field Documentation

8.435.2.1 `uint8_t unpack_qos_Tos_t::mask`

8.435.2.2 `uint8_t unpack_qos_Tos_t::val`

## 8.436 `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`

### 8.436.1 Detailed Description

This structure contains the Data statistic per QoS flow

#### Parameters

<i>bearerId</i>	<ul style="list-style-type: none"> <li>• Bearer ID</li> </ul>
<i>tx_pkt</i>	<ul style="list-style-type: none"> <li>• number of sent packets for the QoS flow ID</li> </ul>
<i>tx_pkt_drp</i>	<ul style="list-style-type: none"> <li>• number of dropped(TX) packets for the QoS flow ID</li> </ul>
<i>tx_bytes</i>	<ul style="list-style-type: none"> <li>• number of sent bytes for the QoS flow ID</li> </ul>
<i>tx_bytes_drp</i>	<ul style="list-style-type: none"> <li>• number of dropped(TX) bytes for the QoS flow ID</li> </ul>

### 8.436.2 Field Documentation

8.436.2.1 `uint32_t unpack_QosFlowStat_t::bearerId`

8.436.2.2 `uint64_t unpack_QosFlowStat_t::tx_bytes`

8.436.2.3 `uint64_t unpack_QosFlowStat_t::tx_bytes_drp`

8.436.2.4 `uint32_t unpack_QosFlowStat_t::tx_pkt`

8.436.2.5 `uint32_t unpack_QosFlowStat_t::tx_pkt_drp`

## 8.437 unpack\_RMTransferStatistics\_ind\_t Struct Reference

### Data Fields

- [wds\\_DataUlongTlv TxOkConutTlv](#)
- [wds\\_DataUlongTlv RxOkConutTlv](#)
- [wds\\_DataUlongLongTlv TxOkByteCountTlv](#)
- [wds\\_DataUlongLongTlv RxOkByteCountTlv](#)
- [wds\\_DataUlongTlv TxDropConutTlv](#)
- [wds\\_DataUlongTlv RxDropConutTlv](#)

### 8.437.1 Detailed Description

WDS Pkt RM Transfer Statistics data structure for individual session

#### Parameters

<i>TxOkConutTlv</i>	<ul style="list-style-type: none"> <li>Tx Ok Packet Tlv Value.</li> </ul>
<i>RxOkConutTlv</i>	<ul style="list-style-type: none"> <li>Rx Ok Packet Tlv Value.</li> </ul>
<i>TxOkByte↔CountTlv</i>	<ul style="list-style-type: none"> <li>Tx Ok Byte Count Packet Tlv Value.</li> </ul>
<i>RxOkByte↔CountTlv</i>	<ul style="list-style-type: none"> <li>Rx Ok Byte Count Packet Tlv Value.</li> </ul>
<i>TxDropConutTlv</i>	<ul style="list-style-type: none"> <li>Tx Drop Count Packet Tlv Value.</li> </ul>
<i>RxDropConutTlv</i>	<ul style="list-style-type: none"> <li>Rx Drop Count Packet Tlv Value.</li> </ul>

## 8.437.2 Field Documentation

8.437.2.1 `wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv`

8.437.2.2 `wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv`

8.437.2.3 `wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv`

8.437.2.4 `wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv`

8.437.2.5 `wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv`

8.437.2.6 `wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv`

## 8.438 `unpack_sms_SendSMS_t` Struct Reference

### Data Fields

- uint16\_t [messageID](#)
- uint32\_t [messageFailureCode](#)

### 8.438.1 Detailed Description

#### Parameters

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



## 8.438.2 Field Documentation

8.438.2.1 uint32\_t unpack\_sms\_SendSMS\_t::messageFailureCode

8.438.2.2 uint16\_t unpack\_sms\_SendSMS\_t::messageID

## 8.439 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

### 8.439.1 Detailed Description

#### Parameters

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

## 8.439.2 Field Documentation

8.439.2.1 struct `eTWSPLMNInfoTlv` `unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv`

8.439.2.2 struct `sMSEtwMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv`

8.439.2.3 struct `sMSOnIMSTlv` `unpack_sms_SetNewSMSCallback_ind_t::IMSTlv`

8.439.2.4 struct `messageModeTlv` `unpack_sms_SetNewSMSCallback_ind_t::MMTlv`

8.439.2.5 struct `newMTMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv`

8.439.2.6 struct `sMSCAddressTlv` `unpack_sms_SetNewSMSCallback_ind_t::SMSTlv`

8.439.2.7 struct `transferRouteMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv`

## 8.440 `unpack_sms_SetNewSMSCallback_t` Struct Reference

## 8.441 `unpack_sms_SLQSDeleteSMS_t` Struct Reference

## 8.442 `unpack_sms_SLQSGetSMS_t` Struct Reference

### Data Fields

- uint32\_t [messageTag](#)
- uint32\_t [messageFormat](#)
- uint32\_t [messageSize](#)
- uint8\_t [message](#) [2048]

### 8.442.1 Detailed Description

#### Parameters

<i>messageTag</i>	<ul style="list-style-type: none"> <li>• Message tag <ul style="list-style-type: none"> <li>– 0 - Read</li> <li>– 1 - Not read</li> <li>– 2 - Mobile originated and sent</li> <li>– 3 - Mobile originated but not yet sent</li> </ul> </li> </ul>
<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	
	<ul style="list-style-type: none"> <li>• Upon input the maximum number of bytes that can be written to the message buffer</li> </ul>

- Upon successful output the actual number of bytes written to the message array.

## Parameters

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

## 8.442.2 Field Documentation

8.442.2.1 uint8\_t unpack\_sms\_SLQSGetSMS\_t::message[2048]

8.442.2.2 uint32\_t unpack\_sms\_SLQSGetSMS\_t::messageFormat

8.442.2.3 uint32\_t unpack\_sms\_SLQSGetSMS\_t::messageSize

8.442.2.4 uint32\_t unpack\_sms\_SLQSGetSMS\_t::messageTag

## 8.443 unpack\_sms\_SLQSGetSMSList\_t Struct Reference

## Data Fields

- uint32\_t [messageListSize](#)
- [qmiSmsMessageList](#) [messageList](#) [255]

## 8.443.1 Detailed Description

## Parameters

<i>messageListSize</i>	<ul style="list-style-type: none"> <li>• Upon input the maximum number of elements that the message list array can contain.</li> <li>• Upon successful output the actual number of elements in the message list array.</li> </ul>
<i>messageList</i>	<ul style="list-style-type: none"> <li>• Message List</li> <li>• See <a href="#">qmiSmsMessageList</a> for more information</li> </ul>

## 8.443.2 Field Documentation

8.443.2.1 [qmiSmsMessageList](#) unpack\_sms\_SLQSGetSMSList\_t::messageList[255]

8.443.2.2 uint32\_t unpack\_sms\_SLQSGetSMSList\_t::messageListSize

## 8.444 unpack\_sms\_SLQSModifySMSStatus\_t Struct Reference

## 8.445 unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t Struct Reference

### Data Fields

- [uint8\\_t storageType](#)
- [uint8\\_t messageMode](#)

### 8.445.1 Detailed Description

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>messageMode</i>	<ul style="list-style-type: none"> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>

### 8.445.2 Field Documentation

8.445.2.1 [uint8\\_t unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind\\_t::messageMode](#)

8.445.2.2 [uint8\\_t unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind\\_t::storageType](#)

## 8.446 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](#)

### 8.446.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

#### Parameters

<i>function</i>	<ul style="list-style-type: none"> <li>Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> <li>0 - disabled</li> <li>1 - At bootup</li> <li>2 - When NMEA port is opened</li> </ul> </li> </ul>
<i>function_↔ reported</i>	<ul style="list-style-type: none"> <li>0 - not reported by modem</li> <li>1 - reported by modem</li> </ul>
<i>fix_type</i>	<ul style="list-style-type: none"> <li>Type of GNSS fix: <ul style="list-style-type: none"> <li>1 - Default Engine mode</li> <li>2 - MS-Based</li> <li>3 - MS-Assisted</li> <li>4 - Standalone</li> </ul> </li> </ul>
<i>fix_type_↔ reported</i>	<ul style="list-style-type: none"> <li>0 - not reported by modem</li> <li>1 - reported by modem</li> </ul>
<i>max_time</i>	<ul style="list-style-type: none"> <li>Maximum time allowed for the receiver to get a fix in seconds</li> <li>Valid range: 1-255</li> </ul>
<i>max_time_↔ reported</i>	<ul style="list-style-type: none"> <li>0 - not reported by modem</li> <li>1 - reported by modem</li> </ul>
<i>max_dist</i>	<ul style="list-style-type: none"> <li>Maximum uncertainty of a fix measured by distance in meters</li> <li>Valid range: 1 - 4294967280</li> </ul>
<i>max_dist_↔ reported</i>	<ul style="list-style-type: none"> <li>0 - not reported by modem</li> <li>1 - reported by modem</li> </ul>
<i>fix_rate</i>	<ul style="list-style-type: none"> <li>Time between fixes in seconds</li> <li>Valid range: 1–65535</li> </ul>
<i>fix_rate_↔ reported</i>	<ul style="list-style-type: none"> <li>0 - not reported by modem</li> <li>1 - reported by modem</li> </ul>

## 8.446.2 Field Documentation

8.446.2.1 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate`

8.446.2.2 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported`

8.446.2.3 uint8\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::fix\_type

8.446.2.4 int unpack\_swiloc\_SwiLocGetAutoStart\_t::fix\_type\_reported

8.446.2.5 uint8\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::function

8.446.2.6 int unpack\_swiloc\_SwiLocGetAutoStart\_t::function\_reported

8.446.2.7 uint32\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_dist

8.446.2.8 int unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_dist\_reported

8.446.2.9 uint8\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_time

8.446.2.10 int unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_time\_reported

## 8.447 unpack\_swroma\_SLQSOMADMAAlertCallback\_ind\_t Struct Reference

### Data Fields

- uint32\_t [eventType](#)
- [unpack\\_omaDmFotaTlv\\_t](#) SessionInfoFota
- [unpack\\_omaDmConfigTlv\\_t](#) SessionInfoConfig
- [unpack\\_omaDmNotificationsTlv\\_t](#) SessionInfoNotification

### 8.447.1 Detailed Description

Structure that contains OMA indication information based on eventType Structures for which the event is not valid will have values set to 0

#### Parameters

<i>eventType</i>	<ul style="list-style-type: none"> <li>• 0x00 - SWIOMA-DM FOTA</li> <li>• 0x01 - SWIOMA-DM Config</li> <li>• 0x02 - SWIOMA-DM Notification</li> </ul>
<i>SessionInfo</i> ↔ <i>Fota</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_omaDmFotaTlv_t</a> for more information</li> </ul>
<i>SessionInfo</i> ↔ <i>Config</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_omaDmConfigTlv_t</a> for more information</li> </ul>
<i>SessionInfo</i> ↔ <i>Notification</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_omaDmNotificationsTlv_t</a> for more information</li> </ul>

## 8.447.2 Field Documentation

8.447.2.1 uint32\_t unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t::eventType

8.447.2.2 unpack\_omaDmConfigTlv\_t unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t::SessionInfoConfig

8.447.2.3 unpack\_omaDmFotaTlv\_t unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t::SessionInfoFota

8.447.2.4 unpack\_omaDmNotificationsTlv\_t unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t::SessionInfoNotification

## 8.448 unpack\_swima\_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](#)

### 8.448.1 Detailed Description

Structure that contains the session type for OMA get session info unpack command Also used as input parameter to specify the size of variable parameters. (ref. notes)

#### Parameters

<i>Status</i>	<ul style="list-style-type: none"><li>• 1 Byte parameter indicating status<ul style="list-style-type: none"><li>– 0x01 - No Firmware available</li><li>– 0x02 - Query Firmware Download</li><li>– 0x03 - Firmware Downloading</li><li>– 0x04 - Firmware Downloaded</li><li>– 0x05 - Query Firmware Update</li><li>– 0x06 - Firmware Updating</li><li>– 0x07 - Firmware Updated</li></ul></li></ul>
---------------	--

<i>Update↔ CompleteStatus</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Update Complete Status <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a> Update Complete Status</li> </ul> </li> </ul>
<i>Severity</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating severity <ul style="list-style-type: none"> <li>– 0x01 - Mandatory</li> <li>– 0x02 - Optional</li> </ul> </li> </ul>
<i>SourceLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Vendor Name String in Bytes.</li> </ul>
<i>Source</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Vendor Name in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>PkgNameLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.</li> </ul>
<i>PkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>PkgDescLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>PkgDescription</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>DateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>Date</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>TimeLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Time String in Bytes.</li> </ul>
<i>Time</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Time String in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>



<i>SessionType</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter reflects the last session started for Sprint <ul style="list-style-type: none"> <li>– 0x00 - No session since boot</li> <li>– 0x01 - Sprint CI-DC Session</li> <li>– 0x02 - Sprint CI-PRL Session</li> <li>– 0x03 - Sprint CI-FUMO Session</li> <li>– 0x04 - Sprint HFA-DC Session</li> <li>– 0x05 - Sprint HFA-PRL Session</li> <li>– 0x06 - Sprint HFA-FUMO Session</li> <li>– 0x07 - Sprint NI Session</li> </ul> </li> </ul>
<i>SessionState</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating session state <ul style="list-style-type: none"> <li>– 0x01 - idle</li> <li>– 0x02 - active</li> <li>– 0x03 - pending</li> </ul> </li> </ul>
<i>RetryCount</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating retries left count <ul style="list-style-type: none"> <li>– valid values 0 to 6</li> </ul> </li> </ul>

## 8.448.2 Field Documentation

8.448.2.1 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::Date[255]

8.448.2.2 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::DateLength

8.448.2.3 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::PkgDescLength

8.448.2.4 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::PkgDescription[255]

8.448.2.5 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::PkgName[255]

8.448.2.6 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::PkgNameLength

8.448.2.7 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::RetryCount

8.448.2.8 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::SessionState

8.448.2.9 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::SessionType

8.448.2.10 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::Severity

8.448.2.11 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::Source[255]

8.448.2.12 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::SourceLength

8.448.2.13 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::Status

8.448.2.14 uint8\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::Time[255]

8.448.2.15 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::TimeLength

8.448.2.16 uint16\_t unpack\_swima\_SLQSOMADMGetSessionInfo\_t::UpdateCompleteStatus

## 8.449 unpack\_swima\_SLQSOMADMGetSettings\_t Struct Reference

### Data Fields

- uint32\_t [OMADMEabled](#)
- uint8\_t [FOTAdownload](#)
- uint8\_t [FOTAUpdate](#)
- uint8\_t [Autosdm](#)
- uint8\_t [FwAutoCheck](#)

### 8.449.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

#### Parameters

<p><i>OMADM</i>↔ <i>Enabled</i>[OUT]</p>	<ul style="list-style-type: none"> <li>• Optional 4 byte parameter indicating OMADM service enabled <ul style="list-style-type: none"> <li>– 0x00000001 - Client-initiated device configuration</li> <li>– 0x00000002 - Network-initiated device configuration</li> <li>– 0x00000010 - Client-initiated FUMO</li> <li>– 0x00000020 - Network-initiated FUMO</li> </ul> </li> <li>• function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem.</li> </ul>
<p><i>FOTA</i>↔ <i>Adownload</i>[O↔ <i>UT</i>]</p>	<ul style="list-style-type: none"> <li>• Optional 1 Byte parameter indicating support for FOTA Automatic download <ul style="list-style-type: none"> <li>– 0x00 - Host permission required before downloading</li> <li>– 0x01 - Automatically start downloading, no host permission required</li> <li>– 0x02 - Automatically start downloading, while not roaming</li> <li>– 0x03 - Automatically reject download</li> <li>– 0x04 - Automatically reject download with “Enterprise Reject Policy”</li> </ul> </li> <li>• function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>

<i>FOTAUpdate</i> <sub>[↔ OUT]</sub>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating FOTA Automatic update <ul style="list-style-type: none"> <li>0x00 - User permission required before updating firmware</li> <li>0x01 - No user permission required before updating firmware</li> <li>0x02 - User permission required, auto update on power up</li> </ul> </li> <li>function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
<i>Autosdm</i> <sub>[OUT]</sub>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic UI Alert Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled Accept</li> <li>0x02 - Enabled Reject</li> </ul> </li> <li>function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>
<i>FwAutoCheck</i> <sub>[↔ OUT]</sub>	<ul style="list-style-type: none"> <li>Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> <li>function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.</li> </ul>

## 8.449.2 Field Documentation

8.449.2.1 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::Autosdm

8.449.2.2 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::FOTAdownload

8.449.2.3 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::FOTAUpdate

8.449.2.4 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::FwAutoCheck

8.449.2.5 uint32\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::OMADMEabled

## 8.450 unpack\_swioama\_SLQSOMADMStartSession\_t Struct Reference

### Data Fields

- uint32\_t [FwAvailability](#)

### 8.450.1 Detailed Description

Structure that contains the responses for OMA start session command

### Parameters

<i>pFw</i> ↔ <i>Availability[OUT]</i>	<ul style="list-style-type: none"> <li>• OMA-DM CHECK FW Available <ul style="list-style-type: none"> <li>– 0x00000001 - FW Available. For CIDC and CIPRL, this value will be returned by the modem. CIDC and CIPRL are asynchronous OMADM sessions.</li> <li>– 0x00000002 - FW Not Available</li> <li>– 0x00000003 - FW Check Timed Out</li> </ul> </li> </ul>
--	---

## 8.450.2 Field Documentation

8.450.2.1 uint32\_t unpack\_swima\_SLQSOMADMStartSession\_t::FwAvailability

## 8.451 unpack\_uim\_ChangePin\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- uint32\_t \* [pIndicationToken](#)
- uint16\_t [Tlvresult](#)

### 8.451.1 Detailed Description

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

#### Parameters

<i>pRemaining</i> ↔ <i>Retries(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1</i> ↔ <i>N1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</li> </ul>
<i>pIndication</i> ↔ <i>Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result is provided in a subsequent indication.</li> <li>• 0xFFFFFFFF, if unavailable</li> </ul>

#### Note

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

## 8.451.2 Field Documentation

8.451.2.1 [uim\\_encryptedPIN1](#)\* [unpack\\_uim\\_ChangePin\\_t::pEncryptedPIN1](#)

8.451.2.2 uint32\_t\* unpack\_uim\_ChangePin\_t::pIndicationToken

8.451.2.3 uim\_remainingRetries\* unpack\_uim\_ChangePin\_t::pRemainingRetries

8.451.2.4 uint16\_t unpack\_uim\_ChangePin\_t::Tlvresult

## 8.452 unpack\_uim\_GetCardStatus\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* [pCardStatus](#)
- [uim\\_hotSwapStatus](#) \* [pHotSwapStatus](#)
- uint16\_t [Tlvresult](#)

### 8.452.1 Detailed Description

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

#### Parameters

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

#### Note

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

### 8.452.2 Field Documentation

8.452.2.1 uim\_cardStatus\* unpack\_uim\_GetCardStatus\_t::pCardStatus

8.452.2.2 uim\_hotSwapStatus\* unpack\_uim\_GetCardStatus\_t::pHotSwapStatus

8.452.2.3 uint16\_t unpack\_uim\_GetCardStatus\_t::Tlvresult

## 8.453 unpack\_uim\_ReadTransparent\_t Struct Reference

### Data Fields

- [uim\\_cardResult](#) \* [pCardResult](#)
- [uim\\_readResult](#) \* [pReadResult](#)
- uint32\_t \* [pIndicationToken](#)
- uint8\_t \* [pEncryptedData](#)
- uint16\_t [Tlvresult](#)

### 8.453.1 Detailed Description

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

#### Parameters

<i>pCardResult</i>	<ul style="list-style-type: none"> <li>• See cardResult for more information.</li> </ul>
<i>pReadResult</i>	<ul style="list-style-type: none"> <li>• See readResult for more information.</li> </ul>
<i>pIndication</i> ↔ <i>Token(optional)</i>	<ul style="list-style-type: none"> <li>• Response in Indication.</li> <li>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.</li> </ul>
<i>pEncrypted</i> ↔ <i>Data(optional)</i>	<ul style="list-style-type: none"> <li>• Encrypted Data.</li> <li>• Indicates whether the data from the card passed in read_result is encrypted.</li> </ul>

#### Note

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

### 8.453.2 Field Documentation

8.453.2.1 `uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult`

8.453.2.2 `uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData`

8.453.2.3 `uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken`

8.453.2.4 `uim_readResult* unpack_uim_ReadTransparent_t::pReadResult`

8.453.2.5 `uint16_t unpack_uim_ReadTransparent_t::Tlvresult`

## 8.454 `unpack_uim_SetPinProtection_t` Struct Reference

#### Data Fields

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

### 8.454.1 Detailed Description

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

#### Parameters

<i>pRemainingRetries</i> (optional)	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1</i> (optional)	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</li> </ul>
<i>pIndicationToken</i> (optional)	<ul style="list-style-type: none"> <li>Response in Indication.</li> <li>When this TLV is present, it indicates that the result is provided in a subsequent indication.</li> <li>0xFFFFFFFF, if unavailable</li> </ul>

**Note**

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

**8.454.2 Field Documentation**

8.454.2.1 `uim_encryptedPIN1*` `unpack_uim_SetPinProtection_t::pEncryptedPIN1`

8.454.2.2 `uint32_t*` `unpack_uim_SetPinProtection_t::pIndicationToken`

8.454.2.3 `uim_remainingRetries*` `unpack_uim_SetPinProtection_t::pRemainingRetries`

8.454.2.4 `uint16_t` `unpack_uim_SetPinProtection_t::Tlvresult`

**8.455 unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t Struct Reference****Data Fields**

- [slots\\_t slotsstatusChange](#)
- `uint8_t` [bNumberOfPhySlots](#)

**8.455.1 Detailed Description**

Structure consist of card status params

**Parameters**

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

## 8.455.2 Field Documentation

8.455.2.1 `uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots`

8.455.2.2 `slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange`

## 8.456 `unpack_uim_SLQSUIMEventRegister_t` Struct Reference

### Data Fields

- `uint32_t eventMask`

### 8.456.1 Detailed Description

#### Parameters

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

## 8.456.2 Field Documentation

8.456.2.1 `uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask`

## 8.457 `unpack_uim_SLQSUIMGetSlotsStatus_t` Struct Reference

### Data Fields

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

### 8.457.1 Detailed Description

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

#### Parameters

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



## 8.457.2 Field Documentation

8.457.2.1 `uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot`

8.457.2.2 `slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus`

## 8.458 unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* [pCardStatus](#)

### 8.458.1 Detailed Description

This structure contains information about Status change callback.

#### Parameters

<i>pCardStatus</i>	Card Status <ul style="list-style-type: none"><li>• See <a href="#">uim_cardStatus</a> for more information.</li></ul>
--------------------	--

## 8.458.2 Field Documentation

8.458.2.1 `uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus`

## 8.459 unpack\_uim\_UnblockPin\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- `uint32_t` \* [pIndicationToken](#)
- `uint16_t` [Tlvresult](#)

### 8.459.1 Detailed Description

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

## Parameters

<i>pRemainingRetries</i> (optional)	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1</i> (optional)	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</li> </ul>
<i>pIndicationToken</i> (optional)	<ul style="list-style-type: none"> <li>Response in Indication.</li> <li>When this TLV is present, it indicates that the result is provided in a subsequent indication.</li> <li>0xFFFFFFFF, if unavailable</li> </ul>

## 8.459.2 Field Documentation

8.459.2.1 [uim\\_encryptedPIN1](#)\* [unpack\\_uim\\_UnblockPin\\_t::pEncryptedPIN1](#)8.459.2.2 [uint32\\_t](#)\* [unpack\\_uim\\_UnblockPin\\_t::pIndicationToken](#)8.459.2.3 [uim\\_remainingRetries](#)\* [unpack\\_uim\\_UnblockPin\\_t::pRemainingRetries](#)8.459.2.4 [uint16\\_t](#) [unpack\\_uim\\_UnblockPin\\_t::Tlvresult](#)8.460 [unpack\\_uim\\_VerifyPin\\_t](#) Struct Reference

## Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [uint16\\_t](#) [Tlvresult](#)

## 8.460.1 Detailed Description

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

## Parameters

<i>pRemainingRetries</i> (optional)	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1</i> (optional)	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</li> </ul>
<i>pIndicationToken</i> (optional)	<ul style="list-style-type: none"> <li>Response in Indication.</li> <li>When this TLV is present, it indicates that the result is provided in a subsequent indication.</li> <li>0xFFFFFFFF, if unavailable</li> </ul>

## Note

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

## 8.460.2 Field Documentation

8.460.2.1 uim\_encryptedPIN1\* unpack\_uim\_VerifyPin\_t::pEncryptedPIN1

8.460.2.2 uint32\_t\* unpack\_uim\_VerifyPin\_t::pIndicationToken

8.460.2.3 uim\_remainingRetries\* unpack\_uim\_VerifyPin\_t::pRemainingRetries

8.460.2.4 uint16\_t unpack\_uim\_VerifyPin\_t::Tlvresult

## 8.461 unpack\_wds\_DHCPv4ClientLease\_ind\_t Struct Reference

### Data Fields

- [wds\\_DHCPProfileIdTlv ProfileIdTlv](#)
- [wds\\_DHCPLeaseStateTlv DHCPv4LeaseStateTlv](#)
- [wds\\_IPv4AdTlv IPv4AddrTlv](#)
- [wds\\_DHCPLeaseOptTlv DHCPv4LeaseOptTlv](#)

### 8.461.1 Field Documentation

8.461.1.1 wds\_DHCPLeaseOptTlv unpack\_wds\_DHCPv4ClientLease\_ind\_t::DHCPv4LeaseOptTlv

8.461.1.2 wds\_DHCPLeaseStateTlv unpack\_wds\_DHCPv4ClientLease\_ind\_t::DHCPv4LeaseStateTlv

8.461.1.3 wds\_IPv4AdTlv unpack\_wds\_DHCPv4ClientLease\_ind\_t::IPv4AddrTlv

8.461.1.4 wds\_DHCPProfileIdTlv unpack\_wds\_DHCPv4ClientLease\_ind\_t::ProfileIdTlv

## 8.462 unpack\_wds\_GetAutoconnect\_t Struct Reference

### Data Fields

- uint32\_t \* [psetting](#)

### 8.462.1 Detailed Description

auto connect data session setting parameter.

### Parameters

<i>pSetting</i>	<ul style="list-style-type: none"> <li>• NDIS auto connect setting <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> </ul>
-----------------	--

### 8.462.2 Field Documentation

8.462.2.1 `uint32_t*` `unpack_wds_GetAutoconnect_t::psetting`

## 8.463 `unpack_wds_GetByteTotals_t` Struct Reference

### Data Fields

- `uint64_t` \* [pTXTotalBytes](#)
- `uint64_t` \* [pRXTotalBytes](#)

### 8.463.1 Detailed Description

#### Parameters

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

### 8.463.2 Field Documentation

8.463.2.1 `uint64_t*` `unpack_wds_GetByteTotals_t::pRXTotalBytes`

8.463.2.2 `uint64_t*` `unpack_wds_GetByteTotals_t::pTXTotalBytes`

## 8.464 `unpack_wds_GetConnectionRate_t` Struct Reference

### Data Fields

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

### 8.464.1 Detailed Description

#### Parameters

<i>current</i> <i>ChannelTXRate</i>	Instantaneous channel Tx rate
<i>current</i> <i>ChannelRXRate</i>	Instantaneous channel Rx rate
<i>maxChannelT</i> <i>XRate</i>	Maximum Tx rate
<i>maxChannelR</i> <i>XRate</i>	Maximum Rx rate

## 8.464.2 Field Documentation

8.464.2.1 uint32\_t unpack\_wds\_GetConnectionRate\_t::currentChannelRXRate

8.464.2.2 uint32\_t unpack\_wds\_GetConnectionRate\_t::currentChannelTXRate

8.464.2.3 uint32\_t unpack\_wds\_GetConnectionRate\_t::maxChannelRXRate

8.464.2.4 uint32\_t unpack\_wds\_GetConnectionRate\_t::maxChannelTXRate

## 8.465 unpack\_wds\_GetDataBearerTechnology\_t Struct Reference

### Data Fields

- uint32\_t \* [pDataBearer](#)

### 8.465.1 Detailed Description

#### Parameters

<i>pDataBearer</i> <i>OUT</i>	<ul style="list-style-type: none"> <li>• Data bearer technology <ul style="list-style-type: none"> <li>– 0x01 - CDMA2000 1x</li> <li>– 0x02 - CDMA 1xEV-DO Rev 0</li> <li>– 0x03 - GSM</li> <li>– 0x04 - UMTS</li> <li>– 0x05 - CDMA2000 HRPD (1xEV-DO Rev A)</li> <li>– 0x06 - EDGE</li> <li>– 0x07 - HSDPA AND WCDMA</li> <li>– 0x08 - WCDMA AND HSUPA</li> <li>– 0x09 - HSDPA AND HSUPA</li> <li>– 0x0A - LTE</li> <li>– 0x0B - CDMA2000 EHRPD</li> <li>– 0x0C - HSDPA+ and WCDMA</li> <li>– 0x0D - HSDPA+ and HSUPA</li> <li>– 0x0E - DC_HSDPA+ and WCDMA</li> <li>– 0x0F - DC_HSDPA+ and HSUPA</li> <li>– 0x10 - HSDPA+ and 64QAM</li> </ul> </li> </ul>
Generated by Doxygen	<ul style="list-style-type: none"> <li>– 0x11 - HSDPA+, 64QAM and HSUPA</li> <li>– 0x12 - TDSCDMA</li> <li>– 0x13 - TDSCDMA and HSDPA</li> <li>– 0xFF - Unknown</li> </ul>

## 8.465.2 Field Documentation

8.465.2.1 uint32\_t\* unpack\_wds\_GetDataBearerTechnology\_t::pDataBearer

## 8.466 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]

### 8.466.1 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

## 8.466.2 Field Documentation

8.466.2.1 int8\_t unpack\_wds\_GetDefaultProfile\_t::apnname[255]

8.466.2.2 uint8\_t unpack\_wds\_GetDefaultProfile\_t::apnsize

8.466.2.3 uint32\_t unpack\_wds\_GetDefaultProfile\_t::auth

- 8.466.2.4    uint32\_t unpack\_wds\_GetDefaultProfile\_t::ipaddr
- 8.466.2.5    uint16\_t unpack\_wds\_GetDefaultProfile\_t::ipaddrv6
- 8.466.2.6    int8\_t unpack\_wds\_GetDefaultProfile\_t::name[255]
- 8.466.2.7    uint8\_t unpack\_wds\_GetDefaultProfile\_t::namesize
- 8.466.2.8    uint32\_t unpack\_wds\_GetDefaultProfile\_t::pdptype
- 8.466.2.9    uint32\_t unpack\_wds\_GetDefaultProfile\_t::pridns
- 8.466.2.10    uint16\_t unpack\_wds\_GetDefaultProfile\_t::pridnsv6
- 8.466.2.11    uint32\_t unpack\_wds\_GetDefaultProfile\_t::secdns
- 8.466.2.12    uint16\_t unpack\_wds\_GetDefaultProfile\_t::secdnsv6
- 8.466.2.13    int8\_t unpack\_wds\_GetDefaultProfile\_t::username[255]
- 8.466.2.14    uint8\_t unpack\_wds\_GetDefaultProfile\_t::usersize

## 8.467    unpack\_wds\_GetDefaultProfileNum\_t Struct Reference

### Data Fields

- uint8\_t [index](#)

### 8.467.1    Detailed Description

#### Parameters

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

### 8.467.2    Field Documentation

- 8.467.2.1    uint8\_t unpack\_wds\_GetDefaultProfileNum\_t::index

## 8.468    unpack\_wds\_GetDormancyState\_t Struct Reference

### Data Fields

- uint32\_t [dormancyState](#)

### 8.468.1 Detailed Description

#### Parameters

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

### 8.468.2 Field Documentation

8.468.2.1 uint32\_t unpack\_wds\_GetDormancyState\_t::dormancyState

## 8.469 unpack\_wds\_GetLastMobileIPError\_t Struct Reference

### Data Fields

- uint32\_t [error](#)

### 8.469.1 Detailed Description

#### Parameters

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

### 8.469.2 Field Documentation

8.469.2.1 uint32\_t unpack\_wds\_GetLastMobileIPError\_t::error

## 8.470 unpack\_wds\_GetMobileIP\_t Struct Reference

### Data Fields

- uint32\_t [mipMode](#)

### 8.470.1 Detailed Description

#### Parameters

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

### 8.470.2 Field Documentation

8.470.2.1 uint32\_t unpack\_wds\_GetMobileIP\_t::mipMode

## 8.471 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](#)

### 8.471.1 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

### 8.471.2 Field Documentation

8.471.2.1 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::AAASPI

8.471.2.2 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::AAASState

8.471.2.3 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::address

8.471.2.4 uint8\_t unpack\_wds\_GetMobileIPProfile\_t::enabled

8.471.2.5 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::HASPI

8.471.2.6 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::HASState

8.471.2.7 int8\_t unpack\_wds\_GetMobileIPProfile\_t::NAI[255]

8.471.2.8 uint8\_t unpack\_wds\_GetMobileIPProfile\_t::naiSize

8.471.2.9 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::primaryHA

8.471.2.10 `uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling`

8.471.2.11 `uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA`

## 8.472 `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`

### 8.472.1 Detailed Description

#### Parameters

<i>pTXPacket↔ Successes</i>	<ul style="list-style-type: none"> <li>• No of transmitted Packets without error.</li> </ul>
<i>pRXPacket↔ Successes</i>	<ul style="list-style-type: none"> <li>• No of received Packets without error.</li> </ul>
<i>pTXPacketErrors</i>	<ul style="list-style-type: none"> <li>• Number of outgoing packets with framing errors.</li> </ul>
<i>pRXPacket↔ Errors</i>	<ul style="list-style-type: none"> <li>• Number of incoming packets with framing errors.</li> </ul>
<i>pTXPacket↔ Overflows</i>	<ul style="list-style-type: none"> <li>• Number of packets dropped because Tx buffer overflowed (out of memory).</li> </ul>
<i>pRXPacket↔ Overflows</i>	<ul style="list-style-type: none"> <li>• Number of packets dropped because Rx buffer overflowed (out of memory).</li> </ul>
<i>pTXOkBytes↔ Count</i>	<ul style="list-style-type: none"> <li>• No of bytes transmitted without error.</li> </ul>
<i>pRXOkBytes↔ Count</i>	<ul style="list-style-type: none"> <li>• No of bytes received without error.</li> </ul>

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

## 8.472.2 Field Documentation

8.472.2.1 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pRXDroppedCount

8.472.2.2 uint64\_t\* unpack\_wds\_GetPacketStatistics\_t::pRXOkBytesCount

8.472.2.3 uint64\_t\* unpack\_wds\_GetPacketStatistics\_t::pRXOKBytesLastCall

8.472.2.4 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pRXPacketErrors

8.472.2.5 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pRXPacketOverflows

8.472.2.6 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pRXPacketSuccesses

8.472.2.7 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pTXDroppedCount

8.472.2.8 uint64\_t\* unpack\_wds\_GetPacketStatistics\_t::pTXOkBytesCount

8.472.2.9 uint64\_t\* unpack\_wds\_GetPacketStatistics\_t::pTXOKBytesLastCall

8.472.2.10 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pTXPacketErrors

8.472.2.11 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pTXPacketOverflows

8.472.2.12 uint32\_t\* unpack\_wds\_GetPacketStatistics\_t::pTXPacketSuccesses

## 8.473 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](#)

### 8.473.1 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>tXOkBytes↔ LastCall</i>	Last call Tx Bytes OK
<i>rXOkBytes↔ LastCall</i>	Last call Rx Bytes OK
<i>tXDroppedCount</i>	Tx Packets Dropped
<i>rXDroppedCount</i>	Rx Packets Dropped

### 8.473.2 Field Documentation

8.473.2.1 [uint32\\_t unpack\\_wds\\_GetPacketStatus\\_t::rXDroppedCount](#)

8.473.2.2 [uint64\\_t unpack\\_wds\\_GetPacketStatus\\_t::rXOkBytesCount](#)

8.473.2.3 [uint64\\_t unpack\\_wds\\_GetPacketStatus\\_t::rXOkBytesLastCall](#)

8.473.2.4 [uint32\\_t unpack\\_wds\\_GetPacketStatus\\_t::rXPacketErrors](#)

8.473.2.5 [uint32\\_t unpack\\_wds\\_GetPacketStatus\\_t::rXPacketOverflows](#)

8.473.2.6 [uint32\\_t unpack\\_wds\\_GetPacketStatus\\_t::rXPacketSuccesses](#)

8.473.2.7 [uint32\\_t unpack\\_wds\\_GetPacketStatus\\_t::tXDroppedCount](#)

8.473.2.8 uint64\_t unpack\_wds\_GetPacketStatus\_t::tXOkBytesCount

8.473.2.9 uint64\_t unpack\_wds\_GetPacketStatus\_t::tXOkBytesLastCall

8.473.2.10 uint32\_t unpack\_wds\_GetPacketStatus\_t::tXPacketErrors

8.473.2.11 uint32\_t unpack\_wds\_GetPacketStatus\_t::tXPacketOverflows

8.473.2.12 uint32\_t unpack\_wds\_GetPacketStatus\_t::tXPacketSuccesses

## 8.474 unpack\_wds\_GetSessionDuration\_t Struct Reference

### Data Fields

- uint64\_t [callDuration](#)

### 8.474.1 Detailed Description

#### Parameters

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

### 8.474.2 Field Documentation

8.474.2.1 uint64\_t unpack\_wds\_GetSessionDuration\_t::callDuration

## 8.475 unpack\_wds\_GetSessionState\_t Struct Reference

### Data Fields

- uint32\_t [connectionStatus](#)

### 8.475.1 Detailed Description

#### Parameters

<a href="#">connection↔ Status</a>	state of the current packet data session
--	--

### 8.475.2 Field Documentation

8.475.2.1 uint32\_t unpack\_wds\_GetSessionState\_t::connectionStatus

## 8.476 unpack\_wds\_RMSetTransferStatistics\_t Struct Reference

## 8.477 unpack\_wds\_SetMobileIPProfile\_t Struct Reference

## 8.478 unpack\_wds\_SLQSCreateProfile\_t Struct Reference

### Data Fields

- [PackCreateProfileOut](#) \* [pCreateProfileOut](#)
- [uint8\\_t](#) \* [pProfileID](#)
- [uint16\\_t](#) [Tlvresult](#)

### 8.478.1 Detailed Description

#### Parameters

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

### 8.478.2 Field Documentation

8.478.2.1 [PackCreateProfileOut](#)\* [unpack\\_wds\\_SLQSCreateProfile\\_t::pCreateProfileOut](#)

8.478.2.2 [uint8\\_t](#)\* [unpack\\_wds\\_SLQSCreateProfile\\_t::pProfileID](#)

8.478.2.3 [uint16\\_t](#) [unpack\\_wds\\_SLQSCreateProfile\\_t::Tlvresult](#)

## 8.479 unpack\_wds\_SLQSDeleteProfile\_t Struct Reference

### Data Fields

- [uint16\\_t](#) [extendedErrorCode](#)

### 8.479.1 Detailed Description

#### Parameters

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

### 8.479.2 Field Documentation

8.479.2.1 [uint16\\_t](#) [unpack\\_wds\\_SLQSDeleteProfile\\_t::extendedErrorCode](#)

## 8.480 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](#)

### 8.480.1 Detailed Description

#### Parameters

	<i>pLTEAttachProfile</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• LTE Attach Profile <ul style="list-style-type: none"> <li>– points to a single WORD Value indicating the attached LTE Profile</li> <li>– Optional parameter with possible values 1-16 (EM/MC73xx or earlier)</li> </ul> </li> <li>• This setting is deprecated on MC/EM74xx</li> </ul>
	<i>profileList</i>	<p>Profile List</p> <ul style="list-style-type: none"> <li>• an array of 4 profile configurations</li> <li>• Each element points to a single WORD value indicating profile</li> <li>• Optional parameter with possible values <ul style="list-style-type: none"> <li>– 1 - 16 (MC/EM73xx and before)</li> <li>– 1 - 24 (MC/EM74xx and onwards)</li> </ul> </li> <li>• function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present</li> <li>• Note: the 5th entry is currently ignored, please set it to zero</li> </ul>
out	<i>defaultPDNEnabled</i>	<ul style="list-style-type: none"> <li>• 0 - disabled</li> <li>• 1 - enabled</li> </ul>
out	<i>_3gppRelease</i>	<p>3GPP release</p> <ul style="list-style-type: none"> <li>• 0 - Release_99</li> <li>• 1 - Release_5</li> <li>• 2 - Release_6</li> <li>• 3 - Release_7</li> <li>• 4 - Release_8</li> <li>• 5 - Release_9 (In 9x30 and toward)</li> <li>• 6 - Release_10 (In 9x30 and toward)</li> <li>• 7 - Release_11 (In 9x30 and toward)</li> </ul>
out	<i>LTEAttachProfileList</i>	<ul style="list-style-type: none"> <li>• pointer to WORD array indicating LTE Attach Profile List <ul style="list-style-type: none"> <li>– Optional parameter</li> <li>– possible values: 1-24</li> <li>– This setting is only supported for MC/EM74xx onwards</li> <li>– Please provide attach profiles in order of decreasing priority in this list.</li> </ul> </li> </ul>

in, out	<i>LTEAttachProfileListLen</i>	<ul style="list-style-type: none"> <li>Number of element in pLTEAttachProfileList <ul style="list-style-type: none"> <li>valid range: 1-24</li> <li>This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>
---------	--------------------------------	---

## 8.480.2 Field Documentation

8.480.2.1 uint8\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::3gppRelease

8.480.2.2 uint8\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::defaultPDNEnabled

8.480.2.3 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::LTEAttachProfile

8.480.2.4 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::LTEAttachProfileList[24]

8.480.2.5 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::LTEAttachProfileListLen

8.480.2.6 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::profileList[5]

## 8.481 unpack\_wds\_SLQSGetCurrDataSystemStat\_t Struct Reference

### Data Fields

- uint8\_t [prefNetwork](#)
- uint8\_t [networkInfoLen](#)
- [currNetworkInfo](#) [currNetworkInfo](#) [255]

### 8.481.1 Detailed Description

#### Parameters

<i>prefNetwork</i>	preferred network
<i>networkInfoLen</i>	number of set of <a href="#">currNetworkInfo</a> elements
<i>currNetworkInfo</i>	current network information.

### 8.481.2 Field Documentation

8.481.2.1 currNetworkInfo unpack\_wds\_SLQSGetCurrDataSystemStat\_t::currNetworkInfo[255]

8.481.2.2 uint8\_t unpack\_wds\_SLQSGetCurrDataSystemStat\_t::networkInfoLen

8.481.2.3 uint8\_t unpack\_wds\_SLQSGetCurrDataSystemStat\_t::prefNetwork



## 8.482 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](#)

### 8.482.1 Detailed Description

#### Parameters

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

### 8.482.2 Field Documentation

8.482.2.1 uint32\_t unpack\_wds\_SLQSGetCurrentChannelRate\_t::current\_channel\_rx\_rate

8.482.2.2 uint32\_t unpack\_wds\_SLQSGetCurrentChannelRate\_t::current\_channel\_tx\_rate

8.482.2.3 uint32\_t unpack\_wds\_SLQSGetCurrentChannelRate\_t::max\_channel\_rx\_rate

8.482.2.4 uint32\_t unpack\_wds\_SLQSGetCurrentChannelRate\_t::max\_channel\_tx\_rate

## 8.483 unpack\_wds\_SLQSGetDataBearerTechnology\_t Struct Reference

### Data Fields

- [uint8\\_t dataBearerMask](#)
- [qmiWSDDataBearerTechnology curDataBearerTechnology](#)
- [qmiWSDDataBearerTechnology lastCallDataBearerTechnology](#)

### 8.483.1 Detailed Description

#### Parameters

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

### 8.483.2 Field Documentation

8.483.2.1 [qmiWSDDataBearerTechnology unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t::curDataBearerTechnology](#)

8.483.2.2 [uint8\\_t unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t::dataBearerMask](#)

8.483.2.3 [qmiWSDDataBearerTechnology unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t::lastCallDataBearerTechnology](#)

## 8.484 unpack\_wds\_SLQSGetDUNCallInfo\_t Struct Reference

### Data Fields

- [connectionStatus connectionStatus](#)
- [uint16\\_t callEndReason](#)
- [uint64\\_t txOKBytesCount](#)
- [uint64\\_t rxOKBytesCount](#)
- [uint8\\_t dormancyStatus](#)
- [uint8\\_t dataBearerTech](#)
- [dunchannelRate channelRate](#)
- [uint64\\_t lastCallTXOKBytesCnt](#)
- [uint64\\_t lastCallRXOKBytesCnt](#)
- [uint64\\_t mdmCallDurationActive](#)
- [uint8\\_t lastCallDataBearerTech](#)

### 8.484.1 Detailed Description

#### Parameters

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

## 8.484.2 Field Documentation

- 8.484.2.1 `uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason`
- 8.484.2.2 `dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate`
- 8.484.2.3 `connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus`
- 8.484.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`
- 8.484.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`
- 8.484.2.6 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech`
- 8.484.2.7 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt`
- 8.484.2.8 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt`
- 8.484.2.9 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive`
- 8.484.2.10 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount`
- 8.484.2.11 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount`

## 8.485 unpack\_wds\_SLQSGetProfileSettings\_t Struct Reference

### Data Fields

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

## 8.485.1 Field Documentation

8.485.1.1 `UnPackGetProfileSettingOut*` `unpack_wds_SLQSGetProfileSettings_t::pProfileSettings`

8.485.1.2 `uint8_t` `unpack_wds_SLQSGetProfileSettings_t::ProfileType`

8.485.1.3 `uint16_t` `unpack_wds_SLQSGetProfileSettings_t::Tlvresult`

## 8.486 `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](#)

### 8.486.1 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>SecondaryDNSV4</i>	
<i>UMTSGrantedQoS</i>	UMTS Granted Qos

<i>GPRSGrantedQoS</i>	GPRS Granted QoS
<i>Username</i>	
<i>Authentication</i>	
<i>ProfileID</i>	
<i>GWAddressV4</i>	Gateway IPv4
<i>SubnetMaskV4</i>	Subnet mask IPV4
<i>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>	

## 8.486.2 Field Documentation

8.486.2.1 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]`

8.486.2.2 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication`

8.486.2.3 `struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList`

8.486.2.4 `struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS`

8.486.2.5 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4`

8.486.2.6 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag`

8.486.2.7 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference`

8.486.2.8 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4`

8.486.2.9 `struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6AddrInfo`

8.486.2.10 `struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6GWAddrInfo`

8.486.2.11 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu`

8.486.2.12 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO`

8.486.2.13 `struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList`

- 8.486.2.14 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPTType`
- 8.486.2.15 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4`
- 8.486.2.16 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]`
- 8.486.2.17 `struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID`
- 8.486.2.18 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]`
- 8.486.2.19 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4`
- 8.486.2.20 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]`
- 8.486.2.21 `struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList`
- 8.486.2.22 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4`
- 8.486.2.23 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology`
- 8.486.2.24 `LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS`
- 8.486.2.25 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]`

## 8.487 `unpack_wds_SLQSModifyProfile_t` Struct Reference

### Data Fields

- `uint16_t * pExtErrorCode`

### 8.487.1 Detailed Description

#### Parameters

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

### 8.487.2 Field Documentation

- 8.487.2.1 `uint16_t*` `unpack_wds_SLQSModifyProfile_t::pExtErrorCode`

## 8.488 `unpack_wds_SLQSSetIPFamilyPreference_t` Struct Reference

### Data Fields

- `uint16_t Tlvresult`

### 8.488.1 Detailed Description

#### Parameters

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

### 8.488.2 Field Documentation

8.488.2.1 uint16\_t unpack\_wds\_SLQSSetIPFamilyPreference\_t::Tlvresult

## 8.489 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](#)

### 8.489.1 Detailed Description

#### Parameters

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

### 8.489.2 Field Documentation

8.489.2.1 uint8\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::bearerID

8.489.2.2 uint8\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::conn\_status

8.489.2.3 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily`

8.489.2.4 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd`

8.489.2.5 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason`

8.489.2.6 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName`

8.489.2.7 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason`

8.489.2.8 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType`

## 8.490 `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]

### 8.490.1 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>dormancyStat↔ Avail</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



## 8.490.2 Field Documentation

- 8.490.2.1 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail`
- 8.490.2.2 `wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]`
- 8.490.2.3 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail`
- 8.490.2.4 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail`
- 8.490.2.5 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology`
- 8.490.2.6 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail`
- 8.490.2.7 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus`
- 8.490.2.8 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail`
- 8.490.2.9 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus`
- 8.490.2.10 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen`
- 8.490.2.11 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork`
- 8.490.2.12 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask`
- 8.490.2.13 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes`
- 8.490.2.14 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts`
- 8.490.2.15 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask`
- 8.490.2.16 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes`
- 8.490.2.17 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts`
- 8.490.2.18 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail`

## 8.491 unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t Struct Reference

### Data Fields

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

### 8.491.1 Detailed Description

#### Parameters

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

## 8.491.2 Field Documentation

8.491.2.1 wdsDhcpv4HwConfig\* unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t::pHwConfig

8.491.2.2 wdsDhcpv4OptionList\* unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t::pRequestOptionList

## 8.492 unpack\_wds\_SLQSSGetLoopback\_t Struct Reference

### Data Fields

- uint8\_t [ByteLoopbackMode](#)
- uint8\_t [ByteLoopbackMultiplier](#)

### 8.492.1 Detailed Description

#### Parameters

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

## 8.492.2 Field Documentation

8.492.2.1 uint8\_t unpack\_wds\_SLQSSGetLoopback\_t::ByteLoopbackMode

8.492.2.2 uint8\_t unpack\_wds\_SLQSSGetLoopback\_t::ByteLoopbackMultiplier

## 8.493 unpack\_wds\_SLQSSStartDataSession\_t Struct Reference

### Data Fields

- uint32\_t \* [psid](#)
- uint32\_t \* [pFailureReason](#)
- uint32\_t \* [pVerboseFailReasonType](#)
- uint32\_t \* [pVerboseFailureReason](#)

### 8.493.1 Detailed Description

#### Parameters

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

### 8.493.2 Field Documentation

8.493.2.1 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::pFailureReason

8.493.2.2 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::psid

8.493.2.3 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::pVerboseFailReasonType

8.493.2.4 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::pVerboseFailureReason

## 8.494 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]

### 8.494.1 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

## 8.494.2 Field Documentation

8.494.2.1 `int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]`

8.494.2.2 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId`

8.494.2.3 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.494.2.4 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address`

8.494.2.5 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress`

8.494.2.6 `struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address`

8.494.2.7 `struct ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress`

8.494.2.8 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address`

8.494.2.9 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`

8.494.2.10 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`

8.494.2.11 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`

8.494.2.12 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`

8.494.2.13 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`

8.494.2.14 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`

8.494.2.15 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

## 8.495 UnPackGetProfileSettingOut Struct Reference

### Data Fields

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

### 8.495.1 Field Documentation

8.495.1.1 `UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile`

8.495.1.2 `uint16_t* UnPackGetProfileSettingOut::pExtErrCode`

## 8.496 unpackWdsProfileParam Union Reference

### Data Fields

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

### 8.496.1 Field Documentation

8.496.1.1 `LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP`

8.496.1.2 `LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2`

## 8.497 wds\_currNetworkInfo Struct Reference

### Data Fields

- `uint8_t NetworkType`
- `uint32_t RATMask`
- `uint32_t SOMask`

### 8.497.1 Detailed Description

Network information structure

#### Parameters

<i>NetworkType</i>	<ul style="list-style-type: none"><li>• Values:<ul style="list-style-type: none"><li>– 0 - 3GPP</li><li>– 1 - 3GPP2</li></ul></li></ul>
<i>RATMask</i>	<ul style="list-style-type: none"><li>• Radio Access Technology (RAT) mask to indicate the type of technology.</li><li>• Values:<ul style="list-style-type: none"><li>– 0 - Don't Care</li><li>– 0x8000 - NULL Bearer</li></ul></li><li>• CDMA RAT mask values:<ul style="list-style-type: none"><li>– 0x01 - CDMA_1x</li><li>– 0x02 - EVDO_REV0</li><li>– 0x04 - EVDO_REVA</li><li>– 0x08 - EVDO_REVB</li><li>– 0x10 - EHRPD</li><li>– 0x20 - FMC</li></ul></li><li>• UMTS RAT mask values:<ul style="list-style-type: none"><li>– 0x01 - WCDMA</li><li>– 0x02 - GPRS</li><li>– 0x04 - HSDPA</li><li>– 0x08 - HSUPA</li><li>– 0x10 - EDGE</li><li>– 0x20 - LTE</li><li>– 0x40 - HSDPA+</li><li>– 0x80 - DC_HSDPA+</li><li>– 0x100 - 64_QAM</li><li>– 0x200 - TDSCDMA</li></ul></li></ul>

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

### 8.497.2 Field Documentation

8.497.2.1 `uint8_t wds_currNetworkInfo::NetworkType`

8.497.2.2 `uint32_t wds_currNetworkInfo::RATMask`

8.497.2.3 `uint32_t wds_currNetworkInfo::SOMask`

## 8.498 wds\_DataULongLongTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t ulData`

### 8.498.1 Field Documentation

8.498.1.1 `uint8_t wds_DataULongLongTlv::TlvPresent`

8.498.1.2 `uint64_t wds_DataULongLongTlv::ulData`

## 8.499 wds\_DataULongTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t ulData`

### 8.499.1 Field Documentation

8.499.1.1 `uint8_t wds_DataULongTlv::TlvPresent`

8.499.1.2 `uint32_t wds_DataULongTlv::ulData`

## 8.500 wds\_DHCPLeaseOptTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t numOpt`
- `wds_DHCPOpt optList [30]`
- `uint8_t optListData [2048]`

### 8.500.1 Field Documentation

8.500.1.1 `uint8_t wds_DHCPLeaseOptTlv::numOpt`

8.500.1.2 `wds_DHCPOpt wds_DHCPLeaseOptTlv::optList[30]`

8.500.1.3 `uint8_t wds_DHCPLeaseOptTlv::optListData[2048]`

8.500.1.4 `uint8_t wds_DHCPLeaseOptTlv::TlvPresent`

## 8.501 wds\_DHCPLeaseStateTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t leaseState`

### 8.501.1 Field Documentation

8.501.1.1 `uint8_t wds_DHCPLeaseStateTlv::leaseState`

8.501.1.2 `uint8_t wds_DHCPLeaseStateTlv::TlvPresent`

## 8.502 wds\_DHCPOpt Struct Reference

### Data Fields

- `uint8_t optCode`
- `uint8_t optValLen`
- `uint8_t * pOptVal`



### 8.502.1 Field Documentation

8.502.1.1 `uint8_t wds_DHCPOpt::optCode`

8.502.1.2 `uint8_t wds_DHCPOpt::optValLen`

8.502.1.3 `uint8_t* wds_DHCPOpt::pOptVal`

## 8.503 wds\_DHCPProfileIdTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t profileType`
- `uint8_t profileId`

### 8.503.1 Field Documentation

8.503.1.1 `uint8_t wds_DHCPProfileIdTlv::profileId`

8.503.1.2 `uint8_t wds_DHCPProfileIdTlv::profileType`

8.503.1.3 `uint8_t wds_DHCPProfileIdTlv::TlvPresent`

## 8.504 wds\_DHCPv4HWConfig Struct Reference

### Data Fields

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

### 8.504.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

#### Parameters

<i>hwType</i>	<ul style="list-style-type: none"><li>• HW Type 1 - Ethernet 20 - Serial</li></ul>
<i>chaddrlen</i>	<ul style="list-style-type: none"><li>• chaddrlen</li></ul>
<i>chaddr</i>	<ul style="list-style-type: none"><li>• chaddr. Max size 16 bytes</li></ul>

## 8.504.2 Field Documentation

8.504.2.1 `uint8_t wds_DHCPv4HWConfig::chaddr[16]`

8.504.2.2 `uint8_t wds_DHCPv4HWConfig::chaddrLen`

8.504.2.3 `uint8_t wds_DHCPv4HWConfig::hwType`

## 8.505 wds\_DHCPv4Option Struct Reference

### Data Fields

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

### 8.505.1 Detailed Description

WDS SWI DHCPv4 Option Structure

#### Parameters

<i>optCode</i>	<ul style="list-style-type: none"> <li>• Option code             <ul style="list-style-type: none"> <li>– 0 - 255</li> </ul> </li> </ul>
<i>optValLen</i>	<ul style="list-style-type: none"> <li>• Option value length             <ul style="list-style-type: none"> <li>– 0 - 255</li> </ul> </li> </ul>
<i>optVal</i>	<ul style="list-style-type: none"> <li>• Option value</li> </ul>

## 8.505.2 Field Documentation

8.505.2.1 `uint8_t wds_DHCPv4Option::optCode`

8.505.2.2 `uint8_t wds_DHCPv4Option::optVal[255]`

8.505.2.3 `uint8_t wds_DHCPv4Option::optValLen`

## 8.506 wds\_DHCPv4OptionList Struct Reference

### Data Fields

- `uint8_t numOpt`
- `wds_DHCPv4Option * pOptList`

### 8.506.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

#### Parameters

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

### 8.506.2 Field Documentation

8.506.2.1 `uint8_t wds_DHCPv4OptionList::numOpt`

8.506.2.2 `wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList`

## 8.507 wds\_DHCPv4ProfileId Struct Reference

#### Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

### 8.507.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

#### Parameters

<i>profileType</i>	<ul style="list-style-type: none"><li>0 for 3GPP</li></ul>
<i>profileId</i>	<ul style="list-style-type: none"><li>1 to 24 for 3GPP profile</li></ul>

### 8.507.2 Field Documentation

8.507.2.1 `uint8_t wds_DHCPv4ProfileId::profileId`

8.507.2.2 `uint8_t wds_DHCPv4ProfileId::profileType`

## 8.508 wds\_Domain Struct Reference

### Data Fields

- uint16\_t [domainLen](#)
- uint8\_t [domainName](#) [256]

### 8.508.1 Detailed Description

This structure contains the DomainName Information

#### Parameters

<i>domainLen</i>	<ul style="list-style-type: none"><li>• length of the recieved Domain name</li></ul>
<i>domainName</i>	<ul style="list-style-type: none"><li>• Domain name(Max 256 characters)</li></ul>

### 8.508.2 Field Documentation

8.508.2.1 uint16\_t wds\_Domain::domainLen

8.508.2.2 uint8\_t wds\_Domain::domainName[256]

## 8.509 wds\_DomainNameList Struct Reference

### Data Fields

- uint8\_t [numInstances](#)
- struct [wds\\_Domain domain](#) [10]

### 8.509.1 Detailed Description

This structure contains the DomainNameList Information

#### Parameters

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

## 8.509.2 Field Documentation

8.509.2.1 struct wds\_Domain wds\_DomainNameList::domain[10]

8.509.2.2 uint8\_t wds\_DomainNameList::numInstances

## 8.510 wds\_GPRSQoS Struct Reference

### Data Fields

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

### 8.510.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

#### Parameters

<i>precedence</i> ↔ <i>Class</i>	<ul style="list-style-type: none"> <li>• Precedence class</li> </ul>
<i>delayClass</i>	<ul style="list-style-type: none"> <li>• Delay class</li> </ul>
<i>reliabilityClass</i>	<ul style="list-style-type: none"> <li>• Reliability class</li> </ul>
<i>peak</i> ↔ <i>Throughput</i> ↔ <i>Class</i>	<ul style="list-style-type: none"> <li>• Peak throughput class</li> </ul>
<i>mean</i> ↔ <i>Throughput</i> ↔ <i>Class</i>	<ul style="list-style-type: none"> <li>• Mean throughput class</li> </ul>

## 8.510.2 Field Documentation

8.510.2.1 uint32\_t wds\_GPRSQoS::delayClass

8.510.2.2 uint32\_t wds\_GPRSQoS::meanThroughputClass

8.510.2.3 uint32\_t wds\_GPRSQoS::peakThroughputClass

8.510.2.4 uint32\_t wds\_GPRSQoS::precedenceClass

8.510.2.5 uint32\_t wds\_GPRSQoS::reliabilityClass

## 8.511 wds\_IPv4AdTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [IPv4Addr](#)

### 8.511.1 Field Documentation

8.511.1.1 uint32\_t wds\_IPv4AdTlv::IPv4Addr

8.511.1.2 uint8\_t wds\_IPv4AdTlv::TlvPresent

## 8.512 wds\_IPV6AddressInfo Struct Reference

### Data Fields

- uint8\_t [IPv6PrefixLen](#)
- uint16\_t [IPAddressV6](#) [8]

### 8.512.1 Detailed Description

This structure contains the IPV6 Address Information

#### Parameters

<i>IPv6PrefixLen</i>	<ul style="list-style-type: none"> <li>• Length of the received IPv6 address in no. of bits; can take value between 0 and 128             <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>IPAddressV6</i>	<ul style="list-style-type: none"> <li>• IPv6 address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format.</li> </ul>

### 8.512.2 Field Documentation

8.512.2.1 uint16\_t wds\_IPV6AddressInfo::IPAddressV6[8]

8.512.2.2 uint8\_t wds\_IPV6AddressInfo::IPv6PrefixLen

## 8.513 wds\_IPV6GWAddressInfo Struct Reference

## Data Fields

- uint8\_t [gwV6PrefixLen](#)
- uint16\_t [gwAddressV6](#) [8]

### 8.513.1 Detailed Description

This structure contains the IPV6 Gateway Address Information

#### Parameters

<i>gwV6PrefixLen</i>	<ul style="list-style-type: none"> <li>• Length of the received IPV6 Gateway address in no. of bits; can take value between 0 and 128</li> </ul>
<i>IPAddressV6</i>	<ul style="list-style-type: none"> <li>• IPV6 Gateway address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format.</li> </ul>

### 8.513.2 Field Documentation

8.513.2.1 uint16\_t wds\_IPV6GWAddressInfo::gwAddressV6[8]

8.513.2.2 uint8\_t wds\_IPV6GWAddressInfo::gwV6PrefixLen

## 8.514 wds\_PCSCFFQDNAddress Struct Reference

## Data Fields

- uint16\_t [fqdnLen](#)
- uint8\_t [fqdnAddr](#) [256]

### 8.514.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

#### Parameters

<i>fqdnLen</i>	<ul style="list-style-type: none"> <li>• length of the received FQDN address</li> </ul>
<i>fqdnAddr</i>	<ul style="list-style-type: none"> <li>• FQDN address(Max 256 characters)</li> </ul>

### 8.514.2 Field Documentation

8.514.2.1 `uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]`

8.514.2.2 `uint16_t wds_PCSCFFQDNAddress::fqdnLen`

## 8.515 wds\_PCSCFFQDNAddressList Struct Reference

### Data Fields

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

### 8.515.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of FQDN addresses received</li> </ul>
<i>pcsfQDN↔ Address</i>	<ul style="list-style-type: none"> <li>• FQDN address information(Max 10 addresses)</li> </ul>

### 8.515.2 Field Documentation

8.515.2.1 `uint8_t wds_PCSCFFQDNAddressList::numInstances`

8.515.2.2 `struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfQDNAddress[10]`

## 8.516 wds\_PCSCFIPv4ServerAddressList Struct Reference

### Data Fields

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

### 8.516.1 Detailed Description

This structure contains the PCSCFIPv4ServerAddressList Information

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• number of address following</li> </ul>
<i>pcsfIPv4Addr</i>	<ul style="list-style-type: none"> <li>• P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)</li> </ul>



## 8.516.2 Field Documentation

8.516.2.1 uint8\_t wds\_PCSCFIPv4ServerAddressList::numInstances

8.516.2.2 uint32\_t wds\_PCSCFIPv4ServerAddressList::pcscfIPv4Addr[64]

## 8.517 wds\_ProfileIdentifier Struct Reference

### Data Fields

- uint8\_t [profileType](#)
- uint8\_t [profileIndex](#)

### 8.517.1 Detailed Description

This structure contains the Profile Identifier Information

#### Parameters

<i>profileType</i>	<ul style="list-style-type: none"><li>• Identifies the type of profile 0x00 = 3GPP</li></ul>
<i>profileIndex</i>	<ul style="list-style-type: none"><li>• Index of profile whose settings were loaded prior to session parameter negotiation for the current call. If this TLV is not present, data call parameters are based on device default settings for each parameter</li></ul>

## 8.517.2 Field Documentation

8.517.2.1 uint8\_t wds\_ProfileIdentifier::profileIndex

8.517.2.2 uint8\_t wds\_ProfileIdentifier::profileType

## 8.518 wds\_profileInfo Union Reference

### Data Fields

- [LibPackprofile\\_3GPP](#) SlqsProfile3GPP
- [LibPackprofile\\_3GPP2](#) SlqsProfile3GPP2

### 8.518.1 Detailed Description

This union consist of profile\_3GPP and profile\_3GPP2 out of which one will be used to create profile.

## 8.518.2 Field Documentation

8.518.2.1 LibPackprofile\_3GPP wds\_profileInfo::SlqsProfile3GPP

8.518.2.2 LibPackprofile\_3GPP2 wds\_profileInfo::SlqsProfile3GPP2

## 8.519 wds\_TrStatInd Struct Reference

### Data Fields

- uint8\_t [statsPeriod](#)
- uint32\_t [statsMask](#)

### 8.519.1 Detailed Description

This structure contains the information about the Transfer Statistics Indicator parameters.

#### Parameters

<i>statsPeriod</i>	<ul style="list-style-type: none"> <li>• Period between transfer statistics reports. <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– Other - Period between reports (seconds)</li> </ul> </li> </ul>
<i>statsMask</i>	<ul style="list-style-type: none"> <li>• Requested statistic bit mask. <ul style="list-style-type: none"> <li>– 0x00000001 - Tx packets OK</li> <li>– 0x00000002 - Rx packets OK</li> <li>– 0x00000004 - Tx packet errors</li> <li>– 0x00000008 - Rx packet errors</li> <li>– 0x00000010 - Tx overflows</li> <li>– 0x00000020 - Rx overflows</li> <li>– 0x00000040 - Tx bytes OK</li> <li>– 0x00000080 - Rx bytes OK</li> </ul> </li> <li>• Each bit set causes the corresponding optional information to be sent in SLQSWdsEvent↵ ReportCallBack.</li> <li>• All unlisted bits are reserved for future use and must be set to zero.</li> </ul>

## 8.519.2 Field Documentation

8.519.2.1 uint32\_t wds\_TrStatInd::statsMask

8.519.2.2 uint8\_t wds\_TrStatInd::statsPeriod

## 8.520 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](#)

## 8.520.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

## Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> <li>• 0x00 - Subscribed</li> <li>• 0x01 - Conversational</li> <li>• 0x02 - Streaming</li> <li>• 0x03 - Interactive</li> <li>• 0x04 - Background</li> </ul>
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Maximum uplink bit rate in bits/sec</li> </ul>
<i>maxDownlink↔ Bitrate</i>	<ul style="list-style-type: none"> <li>• Maximum downlink bit rate in bits/sec</li> </ul>
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed uplink bit rate in bits/sec</li> </ul>
<i>grntDownlink↔ Bitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed downlink bit rate in bits/sec</li> </ul>
<i>qosDelivery↔ Order</i>	<ul style="list-style-type: none"> <li>- Qos delivery order</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - Delivery order on</li> <li>• 0x02 - Delivery order off</li> </ul>
<i>maxSDUSize</i>	<ul style="list-style-type: none"> <li>• Maximum SDU size</li> </ul>

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

## 8.520.2 Field Documentation

### 8.520.2.1 uint8\_t wds\_UMTSMinQoS::deliveryErrSDU

8.520.2.2    `uint32_t wds_UMTSMInQoS::grntDownlinkBitrate`

8.520.2.3    `uint32_t wds_UMTSMInQoS::grntUplinkBitrate`

8.520.2.4    `uint32_t wds_UMTSMInQoS::maxDownlinkBitrate`

8.520.2.5    `uint32_t wds_UMTSMInQoS::maxSDUSize`

8.520.2.6    `uint32_t wds_UMTSMInQoS::maxUplinkBitrate`

8.520.2.7    `uint8_t wds_UMTSMInQoS::qosDeliveryOrder`

8.520.2.8    `uint8_t wds_UMTSMInQoS::resBerRatio`

8.520.2.9    `uint8_t wds_UMTSMInQoS::sduErrorRatio`

8.520.2.10   `uint8_t wds_UMTSMInQoS::trafficClass`

8.520.2.11   `uint32_t wds_UMTSMInQoS::trafficPriority`

8.520.2.12   `uint32_t wds_UMTSMInQoS::transferDelay`

## 8.521    wdsDhcpv4HwConfig Struct Reference

### Data Fields

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

### 8.521.1   Detailed Description

#### Parameters

<i>hwType</i>	DHCP HW Type, examples: <ul style="list-style-type: none"><li>• 0 - Ethernet</li><li>• 20 - Serial</li></ul>
<i>chaddrLen</i>	Length of chaddr field, examples: <ul style="list-style-type: none"><li>• 6 for Ethernet MAC address</li></ul>
<i>chaddr</i>	Client hardware address

### 8.521.2   Field Documentation

8.521.2.1    `uint8_t wdsDhcpv4HwConfig::chaddr[16]`

8.521.2.2 `uint8_t wdsDhcpv4HwConfig::chaddrLen`

8.521.2.3 `uint8_t wdsDhcpv4HwConfig::hwType`

## 8.522 wdsDhcpv4Option Struct Reference

### Data Fields

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

### 8.522.1 Detailed Description

#### Parameters

<i>optCode</i>	Option code <ul style="list-style-type: none"> <li>• 0 - 255</li> </ul>
<i>optValLen</i>	Option value length <ul style="list-style-type: none"> <li>• 0 - 255</li> </ul>
<i>optVal</i>	Option Value

### 8.522.2 Field Documentation

8.522.2.1 `uint8_t wdsDhcpv4Option::optCode`

8.522.2.2 `uint8_t wdsDhcpv4Option::optVal`[255]

8.522.2.3 `uint8_t wdsDhcpv4Option::optValLen`

## 8.523 wdsDhcpv4OptionList Struct Reference

### Data Fields

- `uint8_t numOpt`
- `wdsDhcpv4Option * pOptList`

### 8.523.1 Detailed Description

#### Parameters

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

### 8.523.2 Field Documentation

8.523.2.1 `uint8_t wdsDhcpv4OptionList::numOpt`

8.523.2.2 `wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList`

## 8.524 wdsDhcpv4ProfileId Struct Reference

### Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

### 8.524.1 Detailed Description

#### Parameters

<i>profileType</i>	profile type <ul style="list-style-type: none"><li>• 0 - 3GPP</li></ul>
<i>profileId</i>	profile index <ul style="list-style-type: none"><li>• index identifying the profile 1-24 valid for 3GPP profile type (EM74xx and onwards)</li></ul>

### 8.524.2 Field Documentation

8.524.2.1 `uint8_t wdsDhcpv4ProfileId::profileId`

8.524.2.2 `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)
- #define [SDK\\_VALIDATE\\_INPUT\\_PACK\\_PARAM\\_AND\\_FILL\\_XID](#)(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,  
[eCAT](#) =0x0A,  
[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 DEault_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 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID( 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; \
} \
if (pCtx->xid == 0) \
    return eQCWWAN_ERR_INVALID_XID; \
pBuf[0] = eREQ; \
pBuf[1] = pCtx->xid & 0xff; \
pBuf[2] = pCtx->xid >> 8;
```

9.2.1.7 `#define SDU_HDR_LEN (3)`

9.2.1.8 `#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***  
***eCAT***  
***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](#)
- struct [pack\\_dms\\_SetActivationStatusCallback\\_t](#)
- struct [unpack\\_dms\\_SetActivationStatusCallback\\_t](#)
- struct [pack\\_dms\\_UIMSetPINProtection\\_t](#)
- struct [unpack\\_dms\\_UIMSetPINProtection\\_t](#)
- struct [pack\\_dms\\_UIMUnblockPIN\\_t](#)
- struct [pack\\_dms\\_UIMVerifyPIN\\_t](#)
- struct [pack\\_dms\\_UIMChangePIN\\_t](#)
- struct [pack\\_dms\\_UIMGetControlKeyStatus\\_t](#)
- struct [unpack\\_dms\\_UIMGetControlKeyStatus\\_t](#)
- struct [unpack\\_dms\\_UIMGetPINStatus\\_t](#)
- struct [pack\\_dms\\_UIMSetControlKeyProtection\\_t](#)
- struct [unpack\\_dms\\_UIMSetControlKeyProtection\\_t](#)
- struct [pack\\_dms\\_UIMUnblockControlKey\\_t](#)
- struct [unpack\\_dms\\_UIMUnblockControlKey\\_t](#)
- struct [pack\\_dms\\_ResetToFactoryDefaults\\_t](#)
- struct [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#)
- struct [pack\\_dms\\_ActivateAutomatic\\_t](#)
- struct [eriDataparams](#)
- struct [unpack\\_dms\\_SLQSGetERIFile\\_t](#)
- struct [unpack\\_dms\\_SLQSUIMGetState\\_t](#)
- struct [pack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#)
- struct [crashInformation](#)
- struct [crashInfoParams](#)
- struct [unpack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetHostDevInfo\\_t](#)
- struct [pack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#)

- struct [unpack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetOSInfo\\_t](#)
- struct [pack\\_dms\\_SLQSSwiSetOSInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiSetOSInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetSerialNoExt\\_t](#)

## 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 [SPC\\_SIZE](#) 6
- #define [CK\\_MAX\\_SIZE](#) 8
- #define [ACT\\_CODE\\_MAX\\_SIZE](#) 81
- #define [ERI\\_DATA\\_MAX\\_SIZE](#) 1024
- #define [MEID\\_MAX\\_SIZE](#) 8
- #define [DMS\\_PM\\_ONLINE](#) 0x00 /\* Online \*/
- #define [DMS\\_PM\\_LOW](#) 0x01 /\* Low Power \*/
- #define [DMS\\_PM\\_FACTORY](#) 0x02 /\* Factory Test Mode \*/
- #define [DMS\\_PM\\_OFFLINE](#) 0x03 /\* Offline \*/
- #define [DMS\\_PM\\_RESET](#) 0x04 /\* Reset \*/
- #define [DMS\\_PM\\_SHUT\\_DOWN](#) 0x05 /\* Shut Down \*/
- #define [DMS\\_PM\\_PERSISTENT\\_LOW](#) 0x06 /\* Persistent Low Power \*/
- #define [DMS\\_SET\\_REPORT\\_ENABLE](#) 1
- #define [DMS\\_SET\\_REPORT\\_DISABLE](#) 0
- #define [DMS\\_SWI\\_SET\\_IND\\_ENABLE](#) 1
- #define [DMS\\_SWI\\_SET\\_IND\\_DISABLE](#) 0



## 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)
- int [pack\\_dms\\_SetActivationStatusCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetActivationStatusCallback\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetActivationStatusCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetActivationStatusCallback\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMSetPINProtection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMSetPINProtection\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMSetPINProtection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetPINProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMUnblockPIN](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMUnblockPIN\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMUnblockPIN](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetPINProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMVerifyPIN](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMVerifyPIN\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMVerifyPIN](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetPINProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMChangePIN](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMChangePIN\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMChangePIN](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetPINProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMGetControlKeyStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMGetControlKeyStatus\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMGetControlKeyStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMGetControlKeyStatus\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMGetPINStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_UIMGetPINStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMGetPINStatus\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMSetControlKeyProtection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMSetControlKeyProtection\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMSetControlKeyProtection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetControlKeyProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMUnblockControlKey](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMUnblockControlKey\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMUnblockControlKey](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMUnblockControlKey\\_t](#) \*pOutput)

- int [pack\\_dms\\_ResetToFactoryDefaults](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pReq)
- int [unpack\\_dms\\_ResetToFactoryDefaults](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pOutput)
- int [pack\\_dms\\_ValidateSPC](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pReq)
- int [unpack\\_dms\\_ValidateSPC](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pOutput)
- int [pack\\_dms\\_ActivateAutomatic](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_ActivateAutomatic\\_t](#) \*pReq)
- int [unpack\\_dms\\_ActivateAutomatic](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetERIFile](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetERIFile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetERIFile\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetState](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetState\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetCrashInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#) \*pReq)
- int [unpack\\_dms\\_SLQSSwiGetCrashInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetHostDevInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetHostDevInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetHostDevInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiSetHostDevInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#) \*pReq)
- int [unpack\\_dms\\_SLQSSwiSetHostDevInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetOSInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetOSInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetOSInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiSetOSInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiSetOSInfo\\_t](#) \*pReq)
- int [unpack\\_dms\\_SLQSSwiSetOSInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiSetOSInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetSerialNoExt](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetSerialNoExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetSerialNoExt\\_t](#) \*pOutput)

### 9.3.1 Macro Definition Documentation

9.3.1.1 `#define ACT_CODE_MAX_SIZE 81`

9.3.1.2 `#define CK_MAX_SIZE 8`

9.3.1.3 `#define DMS_IMGDETAILS_LEN 16`

9.3.1.4 `#define DMS_MAX_CUST_ID_LEN 64`

9.3.1.5 `#define DMS_MAX_CUST_VALUE_LEN 8`

- 9.3.1.6 `#define DMS_MAX_FWUPDATE_LOG_STR_SZ 255`
- 9.3.1.7 `#define DMS_MAX_FWUPDATE_REF_STR_SZ 15`
- 9.3.1.8 `#define DMS_PM_FACTORY 0x02 /* Factory Test Mode */`
- 9.3.1.9 `#define DMS_PM_LOW 0x01 /* Low Power */`
- 9.3.1.10 `#define DMS_PM_OFFLINE 0x03 /* Offline */`
- 9.3.1.11 `#define DMS_PM_ONLINE 0x00 /* Online */`
- 9.3.1.12 `#define DMS_PM_PERSISTENT_LOW 0x06 /* Persistent Low Power */`
- 9.3.1.13 `#define DMS_PM_RESET 0x04 /* Reset */`
- 9.3.1.14 `#define DMS_PM_SHUT_DOWN 0x05 /* Shut Down */`
- 9.3.1.15 `#define DMS_SET_REPORT_DISABLE 0`
- 9.3.1.16 `#define DMS_SET_REPORT_ENABLE 1`
- 9.3.1.17 `#define DMS_SLQSFWINFO_APPVERSION_SZ 85`
- 9.3.1.18 `#define DMS_SLQSFWINFO_BOOTVERSION_SZ 85`
- 9.3.1.19 `#define DMS_SLQSFWINFO_CARRIER_SZ 20`
- 9.3.1.20 `#define DMS_SLQSFWINFO_CUR_CARR_NAME 17`
- 9.3.1.21 `#define DMS_SLQSFWINFO_CUR_CARR_REV 13`
- 9.3.1.22 `#define DMS_SLQSFWINFO_MODELID_SZ 20`
- 9.3.1.23 `#define DMS_SLQSFWINFO_PACKAGEID_SZ 85`
- 9.3.1.24 `#define DMS_SLQSFWINFO_PRIVERSION_SZ 16`
- 9.3.1.25 `#define DMS_SLQSFWINFO_SKU_SZ 15`
- 9.3.1.26 `#define DMS_SWI_SET_IND_DISABLE 0`
- 9.3.1.27 `#define DMS_SWI_SET_IND_ENABLE 1`
- 9.3.1.28 `#define DMS_UINT8_MAX_STRING_SZ 255`

9.3.1.29 `#define ERI_DATA_MAX_SIZE 1024`

9.3.1.30 `#define MAX_BUILD_ID_LEN 255`

9.3.1.31 `#define MEID_MAX_SIZE 8`

9.3.1.32 `#define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160`

9.3.1.33 `#define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20`

9.3.1.34 `#define SPC_SIZE 6`

9.3.1.35 `#define UNIQUE_ID_LEN 16`

## 9.3.2 Function Documentation

9.3.2.1 `int pack_dms_ActivateAutomatic ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ActivateAutomatic_t * pReq )`

To set UIM Unblock Control Key pack

### Parameters

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

### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

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

Get Activation State pack

### Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.3 int pack\_dms\_GetBandCapability ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

Get Band Capability pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.4 int pack\_dms\_GetCrashAction ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

Get Crash Action pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

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

Get Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use [pack\\_dms\\_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

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

9.3.2.7 `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

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

get device capability pack

#### Parameters



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

**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\_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

**9.3.2.10 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

9.3.2.11 `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

9.3.2.12 `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

9.3.2.13 `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

**9.3.2.14** `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

**9.3.2.15** `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

9.3.2.16 `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

9.3.2.17 `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

9.3.2.18 `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

**9.3.2.19** 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

**9.3.2.20** 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

9.3.2.21 `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

9.3.2.22 `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

9.3.2.23 `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

**9.3.2.24** 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

**9.3.2.25** 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

9.3.2.26 `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

9.3.2.27 `int pack_dms_ResetToFactoryDefaults ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ResetToFactoryDefaults_t * pReq )`

To set UIM Unblock Control Key pack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.28 `int pack_dms_SetActivationStatusCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetActivationStatusCallback_t * reqArg )`

Set activation status pack

Parameters



in	<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

**9.3.2.29** 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>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

Set Crash Action unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.30 `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

9.3.2.31 `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

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

Set Event Report pack

#### Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.33** int pack\_dms\_SetFirmwarePreference ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Set Firmware Preference pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.34** 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>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

9.3.2.35 `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

9.3.2.36 `int pack_dms_SLQSDmsSwiGetResetInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To get reset info pack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.37 `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

**9.3.2.38** `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

**9.3.2.39** `int pack_dms_SLQSGetERIFile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To set UIM Get ERI file pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.40 `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

9.3.2.41 `int pack_dms_SLQSSwiGetCrashInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiGetCrashInfo_t * pReq )`

To set Get Crash Info pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.42 `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

9.3.2.43 `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

9.3.2.44 `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

9.3.2.45 `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

9.3.2.46 `int pack_dms_SLQSSwiGetHostDevInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To set Host Dev Info pack

**Parameters**

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

**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 pack_dms_SLQSSwiGetOSInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To set Host OS Info pack



## Parameters

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

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.48 int pack\_dms\_SLQSSwiGetSerialNoExt ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

To set serial number extension pack

## Parameters

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

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.49 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

9.3.2.50 `int pack_dms_SLQSSwiSetHostDevInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiSetHostDevInfo_t * pReq )`

To set Host Dev Info pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.51 `int pack_dms_SLQSSwiSetOSInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiSetOSInfo_t * pReq )`

To set Host OS Info pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.52 `int pack_dms_SLQSUIGetState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To set UIM Get 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
out	<i>pReq</i>	requeset parameter

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.53** int pack\_dms\_UIMChangePIN ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_dms\_UIMChangePIN\_t \* *pReq* )

To set UIM change PIN pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.54** int pack\_dms\_UIMGetControlKeyStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_dms\_UIMGetControlKeyStatus\_t \* *pReq* )

To set UIM Get Control Key Status pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.55** `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

**9.3.2.56** `int pack_dms_UIMGetPINStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To set UIM Get PIN Status pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.57 `int pack_dms_UIMSetControlKeyProtection ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMSetControlKeyProtection_t * pReq )`

To set UIM Set Control Key Protection pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.58 `int pack_dms_UIMSetPINProtection ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMSetPINProtection_t * pReq )`

To set UIM PIN protection pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.59 `int pack_dms_UIMUnblockControlKey ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMUnblockControlKey_t * pReq )`

To set UIM Unblock Control Key pack

#### Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.60 `int pack_dms_UIMUnblockPIN ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMUnblockPIN_t * pReq )`

To set UIM unblock PIN pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.61 `int pack_dms_UIMVerifyPIN ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMVerifyPIN_t * pReq )`

To set UIM verify PIN pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.62 `int pack_dms_ValidateSPC ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ResetToFactoryDefaults_t * pReq )`

To set UIM Unblock Control Key pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.63 `int unpack_dms_ActivateAutomatic ( uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t * pOutput )`

To get UIM Unblock Control Key unpack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.64 `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

9.3.2.65 `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

9.3.2.66 `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



9.3.2.67 `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

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

9.3.2.69 `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

9.3.2.70 `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

**9.3.2.71** 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

**9.3.2.72** 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

9.3.2.73 `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

9.3.2.74 `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

9.3.2.75 `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

**9.3.2.76** 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

**9.3.2.77** 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

**9.3.2.78** `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

**9.3.2.79** `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

**9.3.2.80** `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

9.3.2.81 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

9.3.2.82 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

**9.3.2.83** `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

**9.3.2.84** `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

**9.3.2.85** `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

**9.3.2.86** 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

**9.3.2.87** 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

**9.3.2.88** `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

**9.3.2.89** `int unpack_dms_ResetToFactoryDefaults ( uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t * pOutput )`

To get UIM Unblock Control Key unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.90** `int unpack_dms_SetActivationStatusCallback ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetActivationStatusCallback_t * pOutput )`

Set Activation status unpack

Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.91 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

9.3.2.92 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

**9.3.2.93** `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

**9.3.2.94** `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

**9.3.2.95** `int unpack_dms_SetEventReport_ind ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetEventReport_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

**9.3.2.96** 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

**9.3.2.97** 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

9.3.2.98 `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

9.3.2.99 `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

9.3.2.100 `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

**9.3.2.101** int unpack\_dms\_SLQSDmsSwilIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSDmsSwilIndicationRegister\_t \* *pOutput* )

Set the registration state for different indication unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**Note**

support EM/MC74xx onwards

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.102** 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

9.3.2.103 int unpack\_dms\_SLQSGetERIFile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSGetERIFile\_t \* *pOutput* )

To get UIM Get ERI file unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.104 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

9.3.2.105 int unpack\_dms\_SLQSSwiGetCrashInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSSwiGetCrashInfo\_t \* *pOutput* )

To get Crash 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

9.3.2.106 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

9.3.2.107 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

**9.3.2.108** 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

**9.3.2.109** 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

9.3.2.110 `int unpack_dms_SLQSSwiGetHostDevInfo ( uint8_t * pResp, uint16_t respLen,  
unpack_dms_SLQSSwiGetHostDevInfo_t * pOutput )`

To get Host Dev Info unpack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_XXX error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_XXX error values

9.3.2.111 `int unpack_dms_SLQSSwiGetOSInfo ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiGetOSInfo_t * pOutput )`

To get Host OS Info unpack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_XXX error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_XXX error values

9.3.2.112 `int unpack_dms_SLQSSwiGetSerialNoExt ( uint8_t * pResp, uint16_t respLen,  
unpack_dms_SLQSSwiGetSerialNoExt_t * pOutput )`

To get serial number extension unpack

#### Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.113 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

9.3.2.114 int unpack\_dms\_SLQSSwiSetHostDevInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiSetHostDevInfo\_t \* *pOutput* )

To get Host Dev Info unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.115 int unpack\_dms\_SLQSSwiSetOSInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSSwiSetOSInfo\_t \* *pOutput* )

To get Host OS Info unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.116 int unpack\_dms\_SLQSUIGetState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSUIGetState\_t \* *pOutput* )

To get UIM Get 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

9.3.2.117 int unpack\_dms\_UIMChangePIN ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMSetPINProtection\_t \* *pOutput* )

To get UIM change PIN unpack

Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.118 int unpack\_dms\_UIMGetControlKeyStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_UIMGetControlKeyStatus\_t \* *pOutput* )

To get UIM Get ControlKeyStatus unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.119 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

9.3.2.120 `int unpack_dms_UIMGetPINStatus ( uint8_t * pResp, uint16_t respLen, unpack_dms_UIMGetPINStatus_t * pOutput )`

To get UIM Get PIN Status unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.121 `int unpack_dms_UIMSetControlKeyProtection ( uint8_t * pResp, uint16_t respLen, unpack_dms_UIMSetControlKeyProtection_t * pOutput )`

To get UIM Set Control Key Protection unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.122 `int unpack_dms_UIMSetPINProtection ( uint8_t * pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t * pOutput )`

To get UIM Set PIN protection unpack

Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.123** int unpack\_dms\_UIMUnblockControlKey ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMUnblockControlKey\_t \* *pOutput* )

To get UIM Unblock Control Key unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.3.2.124** int unpack\_dms\_UIMUnblockPIN ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMSetPINProtection\_t \* *pOutput* )

To get UIM unblock PIN unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.125 `int unpack_dms_UIMVerifyPIN ( uint8_t * pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t * pOutput )`

To get UIM verify PIN unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.3.2.126 `int unpack_dms_ValidateSPC ( uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t * pOutput )`

To get UIM Unblock Control Key unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## 9.4 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 [FMSImageElement](#)
- struct [FMSImageIDEntries](#)
- struct [FMSImageList](#)
- struct [unpack\\_fms\\_GetStoredImages\\_t](#)
- struct [pack\\_fms\\_SetImagesPreference\\_t](#)
- struct [unpack\\_fms\\_SetImagesPreference\\_t](#)

## Macros

- `#define FMS_GOBI_MBN_IMG_ID_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_GOB_I_MBN_IMG_ID_STR_LEN 16`

9.4.1.5 `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`

9.4.1.6 `#define FMS_IMAGE_ID_IMG_ID_LEN 16`

9.4.1.7 `#define FMS_IMAGE_ID_MAX_ENTRIES 2`

9.4.1.8 `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

9.4.1.9 `#define FMS_MAX_IMAGE_ID_ELEMENT 50`

9.4.1.10 `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`

## 9.4.2 Function Documentation

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

This API distills valid Firmware/PRI combinations from GetStoredImages result

### Parameters

in	<i>pStoredImage↵ List</i>	<ul style="list-style-type: none"> <li>image list returned from GetStoredImages</li> <li>See <a href="#">FMSImageList</a></li> </ul>
in, out	<i>pValid↵ CombinationSize</i>	<ul style="list-style-type: none"> <li>number of combination passed in and returned</li> </ul>
out	<i>pValid↵ Combinations</i>	<ul style="list-style-type: none"> <li>valid combinations returned</li> <li>See <a href="#">CarrierImage_t</a></li> </ul>

### Returns

- eQCWWAN\_ERR\_INVALID\_ARG - Invalid parameters
- eQCWWAN\_ERR\_BUFFER\_SZ - No enough element to store combinatons returned

### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.4.2.2 `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(* libsdplogger) (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` (libsdplogger func)
- void `libsdh_SetReadBlockSize` (unsigned long IBlockSize)

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

## 9.5.3.2 typedef void(\* libsdplogger) (uint8\_t lvl, const char \*buff)

This Custom Log prototype.

## Parameters

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

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

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>SetPrefRspFromModem</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">unpack_fms_SetImagesPreference_t</a> for more information.</li> </ul>
---------------------------------	--

##### Returns

Image Mask

- IMG\_MASK\_MDM | IMG\_MASK\_PRI

#### 9.5.5.3 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"> <li>See <a href="#">libSDP_FirmwareInfo</a> for more information.</li> </ul>
------------------	---

##### Returns

0 on success, -1 error value otherwise

#### 9.5.5.4 unsigned int libSDP\_DownloadFW ( char \* *pImagePath*, char \* *szTTYPath*, int *iFWImageType*, int *image\_mask*, int *iModelFamily* )

This API Download Firmware.

##### Parameters

<i>pImagePath</i> [IN]	<ul style="list-style-type: none"> <li>Firmware Folder Path.</li> </ul>
<i>szTTYPath</i> [IN]	<ul style="list-style-type: none"> <li>QDL Device Path.</li> </ul>



<i>iFWImage</i> <i>Type</i> [IN]	<ul style="list-style-type: none"> <li>Firmware Type.</li> <li>See <a href="#">libSDP_Fw_Type</a></li> </ul>
<i>image_mask</i> [IN]	<ul style="list-style-type: none"> <li>Image Mask. <ul style="list-style-type: none"> <li>– IMG_MASK_MDM   IMG_MASK_PRI</li> </ul> </li> </ul>
<i>iModelFamily</i> [IN]	<ul style="list-style-type: none"> <li>Modem Family.</li> <li>See <a href="#">libSDP_Models</a></li> </ul>

**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 \* *plmagePath*, libSDP\_FirmwareInfo \* *info* )

This API Extrac Firmware Parameters From Path.

**Parameters**

<i>plmagePath</i> [IN]	<ul style="list-style-type: none"> <li>Firmware Folder Path.</li> </ul>
<i>libSDP_</i> <i>FirmwareInfo</i> [ <i>OUT</i> ]	<ul style="list-style-type: none"> <li>See <a href="#">libSDP_FirmwareInfo</a> for more information.</li> </ul>

**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"><li>• See <a href="#">libSDP_FirmwareInfo</a> for more information.</li></ul>
--------------------	---

**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"><li>• Model String.</li></ul>
--------------------------	---

**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"> <li>See <a href="#">libsdplogger</a> for more information.</li> </ul>
------------------	--

## Returns

none

9.5.5.10 void libsdp\_SetReadBlockSize ( unsigned long *IBlockSize* )

This API Set Read BlockSize.

## Parameters

<i>iBlockSize</i> [IN]	<ul style="list-style-type: none"> <li>Firmware Read Block Size.</li> </ul>
------------------------	---

## 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](#)
- struct [unpack\\_loc\\_SetOperationMode\\_Ind\\_t](#)
- struct [unpack\\_loc\\_DeleteAssistData\\_Ind\\_t](#)
- struct [loc\\_satelliteInfo](#)
- struct [unpack\\_loc\\_GnssSvInfo\\_Ind\\_t](#)
- struct [pack\\_loc\\_SLQSLOCInjectUTCtime\\_t](#)
- struct [altSrcInfo\\_t](#)
- struct [pack\\_loc\\_SLQSLOCInjectPosition\\_t](#)
- struct [pack\\_loc\\_SLQSLOCSetCradleMountConfig\\_t](#)
- struct [sensorData\\_t](#)
- struct [tempData\\_t](#)
- struct [pack\\_loc\\_SLQSLOCInjectSensorData\\_t](#)

## Macros

- #define [LOC\\_UINT8\\_MAX\\_STRING\\_SZ](#) 255
- #define [LOCEVENTMASKPOSITIONREPORT](#) 0x00000001
- #define [LOCEVENTMASKGNSSSVINFO](#) 0x00000002
- #define [LOCEVENTMASKNMEA](#) 0x00000004
- #define [LOCEVENTMASKNINOTIFYVERIFYREQ](#) 0x00000008
- #define [LOCEVENTMASKINJECTTIMERREQ](#) 0x00000010
- #define [LOCEVENTMASKINJECTPREDICTEDORBITSREQ](#) 0x00000020
- #define [LOCEVENTMASKINJECTPOSITIONREQ](#) 0x00000040
- #define [LOCEVENTMASKENGINESTATE](#) 0x00000080
- #define [LOCEVENTMASKFIXSESSIONSTATE](#) 0x00000100
- #define [LOCEVENTMASKWIFIREQ](#) 0x00000200
- #define [LOCEVENTMASKSENSORSTREAMINGREADYSTATUS](#) 0x00000400
- #define [LOCEVENTMASKTIMESYNCREQ](#) 0x00000800
- #define [LOCEVENTMASKSETSPSTREAMINGREPORT](#) 0x00001000
- #define [LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ](#) 0x00002000
- #define [LOCEVENTMASKNIGEOFENCENOTIFICATION](#) 0x00004000
- #define [LOCEVENTMASKGEOFENCEGENALERT](#) 0x00008000
- #define [LOCEVENTMASKGEOFENCEBREACHNOTIFICATION](#) 0x00010000
- #define [LOCEVENTMASKPEDOMETERCONTROL](#) 0x00020000
- #define [LOCEVENTMASKMOTIONDATACONTROL](#) 0x00040000
- #define [LOCEVENTMASKBATCHFULLNOTIFICATION](#) 0x00080000
- #define [LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT](#) 0x00100000
- #define [LOCEVENTMASKINJECTWIFIAPDATAREQ](#) 0x00200000
- #define [LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION](#) 0x00400000
- #define [LOCEVENTMASKVEHICLEDATAREADYSTATUS](#) 0x00800000
- #define [LOCEVENTMASKGNSSMEASUREMENTREPORT](#) 0x01000000
- #define [LOCEVENTMASKINVALIDVALUE](#) 0xFFFFFFFF
- #define [MAX\\_SENSOR\\_DATA\\_LEN](#) 64
- #define [MAX\\_TEMP\\_DATA\\_LEN](#) 64

## 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\\_DeleteAssistData\\_t](#) \*reqArg)
- int [unpack\\_loc\\_DeleteAssistData](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_DeleteAssistData\\_t](#) \*pOutput)
- int [unpack\\_loc\\_PositionRpt\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_PositionRpt\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_EngineState\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_EngineState\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_SetExtPowerConfig\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetExtPowerConfig\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCGetBestAvailPos](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCGetBestAvailPos\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCGetBestAvailPos](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCGetBestAvailPos\\_t](#) \*pOutput)
- int [unpack\\_loc\\_BestAvailPos\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_BestAvailPos\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_SetOperationMode\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetOperationMode\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_DeleteAssistData\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_DeleteAssistData\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_GnssSvInfo\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_GnssSvInfo\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCInjectUTCTime](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCInjectUTCTime\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCInjectUTCTime](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_loc\\_SLQSLOCInjectPosition](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCInjectPosition\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCInjectPosition](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_loc\\_SLQSLOCSetCradleMountConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCSetCradleMountConfig\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCSetCradleMountConfig](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_loc\\_SLQSLOCInjectSensorData](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCInjectSensorData\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCInjectSensorData](#) (uint8\_t \*pResp, uint16\_t respLen)

## 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.



9.6.1.28 `#define MAX_SENSOR_DATA_LEN 64`

9.6.1.29 `#define MAX_TEMP_DATA_LEN 64`

## 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>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

9.6.3.4 `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\_SLQSLOCInjectPosition ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_loc\_SLQSLOCInjectPosition\_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 parameter

#### 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\_SLQSLOCInjectSensorData ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_loc\_SLQSLOCInjectSensorData\_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 parameter

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.3.8** `int pack_loc_SLQSLOCInjectUTCTime ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCInjectUTCTime_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 parameter

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.3.9** `int pack_loc_SLQSLOCSetCradleMountConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCSetCradleMountConfig_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 parameter

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.3.10** `int pack_loc_Start ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Start_t * reqArg )`

LOC Start pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.6.3.11** 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

**9.6.3.12** int unpack\_loc\_BestAvailPos\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_BestAvailPos\_Ind\_t \* *pOutput* )

Loc Best Avail position Indication unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_XXX error values

**9.6.3.13** `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

**9.6.3.14** `int unpack_loc_DeleteAssistData_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_DeleteAssistData_Ind_t * pOutput )`

Unpack the status of delete the location engine assistance data

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_XXX error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_XXX error values

**9.6.3.15** `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

**9.6.3.16** 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

**9.6.3.17** int unpack\_loc\_GnssSvInfo\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_GnssSvInfo\_Ind\_t \* *pOutput* )

Unpack the GNSS SV Info Indication.

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.6.3.18 `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

9.6.3.19 `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

9.6.3.20 `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

**9.6.3.21** 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

**9.6.3.22** int unpack\_loc\_SetOperationMode\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SetOperationMode\_Ind\_t \* *pOutput* )

Unpack the engine to use the specified operation mode.

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.6.3.23 `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

9.6.3.24 `int unpack_loc_SLQSLOCInjectPosition ( uint8_t * pResp, uint16_t respLen )`

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

9.6.3.25 `int unpack_loc_SLQSLOCInjectSensorData ( uint8_t * pResp, uint16_t respLen )`

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

**9.6.3.26** int unpack\_loc\_SLQSLOCInjectUTCTime ( uint8\_t \* *pResp*, uint16\_t *respLen* )

**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

**9.6.3.27** int unpack\_loc\_SLQSLOCSetCradleMountConfig ( uint8\_t \* *pResp*, uint16\_t *respLen* )

**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

**9.6.3.28** 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

9.6.3.29 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\\_SLQSNasSwiIndicationRegister\\_t](#)
- struct [pack\\_nas\\_SLQSGetPLMNName\\_t](#)
- struct [unpack\\_nas\\_SLQSGetPLMNName\\_t](#)
- struct [nas\\_nmrCellInfo](#)
- struct [nas\\_GERANInfo](#)
- struct [nas\\_geranInstInfo](#)
- struct [nas\\_UMTSinstInfo](#)
- struct [nas\\_UMTSInfo](#)
- struct [nas\\_CDMAInfo](#)
- struct [nas\\_cellParams](#)
- struct [nas\\_LTEInfoIntrafreq](#)
- struct [nas\\_infoInterFreq](#)
- struct [nas\\_LTEInfoInterfreq](#)
- struct [nas\\_gsmCellInfo](#)

- struct [nas\\_lteGsmCellInfo](#)
- struct [nas\\_LTEInfoNeighboringGSM](#)
- struct [nas\\_wcdmaCellInfo](#)
- struct [nas\\_lteWcdmaCellInfo](#)
- struct [nas\\_LTEInfoNeighboringWCDMA](#)
- struct [nas\\_umtsLTENbrCell](#)
- struct [nas\\_WCDMAInfoLTENeighborCell](#)
- struct [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t](#)
- struct [nas\\_timeInfo](#)
- struct [unpack\\_nas\\_SLQSGetNetworkTime\\_t](#)
- struct [nas\\_UniversalTime](#)
- struct [unpack\\_nas\\_SLQSNasNetworkTimeCallBack\\_ind\\_t](#)
- struct [nas\\_PhyCaAggScellIndType](#)
- struct [nas\\_PhyCaAggScellIDBw](#)
- struct [nas\\_PhyCaAggScellInfo](#)
- struct [nas\\_PhyCaAggPcellInfo](#)
- struct [nas\\_PhyCaAggScellIndex](#)
- struct [unpack\\_nas\\_SetNasLTECphyCaIndCallback\\_ind\\_t](#)
- struct [nas\\_RxSigInfo](#)
- struct [nas\\_SccRxInfo](#)
- struct [unpack\\_nas\\_SLQSSwiGetLteSccRxInfo\\_t](#)
- struct [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_t](#)

## 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\\_SLQSSetSysInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSSetSysInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSetSysInfo\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasSysInfoCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasSysInfoCallback\\_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\\_SLQSSetServingSystem](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSSetServingSystem](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSetServingSystem\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSSetSignalStrength](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint16\_t reqMask)
- int [unpack\\_nas\\_SLQSSetSignalStrength](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSetSignalStrength\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSSetSignalStrengthsCallback](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSSetSignalStrengthsCallback\\_t](#) \*pReqParam)



- int [unpack\\_nas\\_SLQSSetSignalStrengthsCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SetRFInfoCallback](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint8\_t \*pBenable)
- int [unpack\\_nas\\_SetRFInfoCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SetLURejectCallback](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint8\_t \*pBenable)
- int [unpack\\_nas\\_SetLURejectCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_nas\\_SetEventReportInd](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetEventReportInd\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetCDMANetworkParameters](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetCDMANetworkParameters](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_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\\_SLQSNasSwiIndicationRegister](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasSwiIndicationRegister\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNasSwiIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen)
- 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\\_SLQSSwiGetLteScRxInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReq, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSSwiGetLteScRxInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiGetLteScRxInfo\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasTimerCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_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	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	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	<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

9.7.3.7 `int pack_nas_GetServingNetwork ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.7.3.8 `int pack_nas_GetServingNetworkCapabilities ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.7.3.9 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\_SLQSNasSwiIndicationRegister ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_nas\_SLQSNasSwiIndicationRegister\_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

**9.7.3.28** int pack\_nas\_SLQSNasSmiModemStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.7.3.29** 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\_SLQSSwiGetLteScsRxInfo ( 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_SetNasLTECphyCalIndCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCalIndCallback_ind_t * pOutput )`

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.7.3.49** 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_SLQSIInitiateNetworkRegistration ( 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
Generated by Doxygen	<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\_SLQSNasSwiIndicationRegister ( 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

**9.7.3.68** int unpack\_nas\_SLQSNasSwiModemStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasSwiModemStatus\_t \* *pOutput* )

**Parameters**

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



**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.7.3.69 `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_SLQSNasTimerCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSNasTimerCallback_ind_t * pOutput )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	network timer indication unpacked

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### 9.7.3.72 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

### 9.7.3.73 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

### 9.7.3.74 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

9.7.3.75 `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

9.7.3.76 `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

9.7.3.77 `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
Generated by Doxygen	<i>pOutput</i>	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 Coding Group Bits 7..4(0000)

- 0x01 - QMI\_VOICE\_REASON\_FWD\_UNCONDITIONAL - Unconditional call forwarding
- 0x02 - QMI\_VOICE\_REASON\_FWD\_MOBILEBUSY - Forward when the mobile is busy
- 0x03 - QMI\_VOICE\_REASON\_FWD\_NOREPLY - Forward when there is no reply
- 0x04 - QMI\_VOICE\_REASON\_FWD\_UNREACHABLE - Forward when the call is unreachable
- 0x05 - QMI\_VOICE\_REASON\_FWD\_ALLFORWARDING - All forwarding
- 0x06 - QMI\_VOICE\_REASON\_FWD\_ALLCONDITIONAL - All conditional forwarding
- 0x07 - QMI\_VOICE\_REASON\_BARR\_ALLOUTGOING - All outgoing
- 0x08 - QMI\_VOICE\_REASON\_BARR\_OUTGOINGINT - Outgoing internal
- 0x09 - QMI\_VOICE\_REASON\_BARR\_OUTGOINGINTEXTOHOM - Outgoing external to home
- 0x0A - QMI\_VOICE\_REASON\_BARR\_ALLINCOMING - All incoming
- 0x0B - QMI\_VOICE\_REASON\_BARR\_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI\_VOICE\_REASON\_BARR\_ALLBARRING - All calls are barred
- 0x0D - QMI\_VOICE\_REASON\_BARR\_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI\_VOICE\_REASON\_BARR\_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI\_VOICE\_REASON\_CALLWAITING - Call waiting
- 0x10 - VOICE\_CC\_SUPS\_RESULT\_REASON\_CLIP - CLIP

- 0x11 - VOICE\_CC\_SUPS\_RESULT\_REASON\_CLIR - CLIR
- 0x12 - VOICE\_CC\_SUPS\_RESULT\_REASON\_COLP - COLP
- 0x13 - VOICE\_CC\_SUPS\_RESULT\_REASON\_COLR - COLR
- 0x14 - VOICE\_CC\_SUPS\_RESULT\_REASON\_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.10 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 CDMA

- 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 SNDPC(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 CDMA

- 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 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CONN REL
- 1049 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE RLF
- 1050 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CTRL NOT CONN
- 1051 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE
- 1052 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE ABORTED
- 1053 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE ACCESS BARRED
- 1054 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRRC(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRRC(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRRC(LTE Radio Resource Control) CONN REL RLF
- 1063 - LRRRC(LTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRRC(LTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRRC(LTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRRC(LTE Radio Resource Control) CONN REL SIB READ ERROR
- 1067 - DETACH WITH REATTACH LTE NW DETACH
- 1068 - DETACH WITH OUT REATTACH LTE NW DETACH
- 1069 - ESM(EPS Session Management) PROC TIME OUT
- 1070 - MESSAGE EXCEED MAX L2 LIMIT
- 1500 - CD GEN OR BUSY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of general or network busy
- 1501 - CD BILL OR AUTH, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to the reception of a Connection Deny message with a deny code of billing failure or authentication failure
- 1502 - CHG HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since there is a change to HDR system due to redirection or PRL not preferred
- 1503 - EXIT HDR, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device exited HDR due to redirection or PRL not preferred
- 1504 - HDR NO SESSION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have a HDR session

- 1505 - HDR ORIG DURING GPS FIX, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since it is ending an HDR call origination in favor of a GPS fix
- 1506 - HDR CS TIMEOUT, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since connection setup on HDR system timed out
- 1507 - HDR RELEASED BY CM, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when it wants to release a HDR call so a 1X call can continue
- 1508 - COLLOC ACQ FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) when device failed to acquire co-located HDR for origination
- 1509 - OTASP COMMIT IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since an OTASP commit is in progress
- 1510 - NO HYBR HDR SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device has no Hybrid HDR service
- 1511 - HDR NO LOCK GRANTED, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module could not get the RF lock
- 1512 - HOLD OTHER IN PROG, this error code is returned when data call is brought down by CM(Call Manager) because DBM or SMS is in progress
- 1513 - HDR FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since HDR module released the call due to fade
- 1514 - HDR ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to HDR system Access Failure
- 2000 - CLIENT END, this error code is returned when client ends the data call
- 2001 - NO SRV, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since device has no service
- 2002 - FADE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device lost the system due to fade
- 2003 - REL NORMAL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with no reason
- 2004 - ACC IN PROG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access attempt already in progress
- 2005 - ACC FAIL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to Access Failure
- 2006 - REDIR OR HANDOFF, this error code is returned when data call is brought down because device is in the process of redirecting/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-3GPP 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 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

Copyright: © 2011-2014 Sierra Wireless, Inc. all rights reserved

## 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 Coding Group Bits 7..4(0000)

### 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\_DATAACIRCUITSYNC
- 0X20 - CLASS\_DATAACIRCUITASYNC
- 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\_EFFECT,
  - eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE,

```

eQCWWAN_ERR_QMI_WIDTH = 0xFFFF }
• enum qm_wds_ds_profile_extended_err_codes {
eWDS_ERR_PROFILE_REG_RESULT_FAIL = 1,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_HNDL,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_OP,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_TYPE,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_PROFILE_NUM,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_IDENT,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID,
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED,
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID,
eWDS_ERR_PROFILE_REG_RESULT_LIST_END,
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVALID_SUBS_ID,
eWDS_ERR_PROFILE_REG_INVALID_PROFILE_FAMILY,
eWDS_ERR_PROFILE_REG_3GPP_INVALID_PROFILE_FAMILY = 1001,
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR,
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED,
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET,
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET,
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES,
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE = 1101,
eWDS_ERR_PROFILE_REG_END }

```

## 9.22.1 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 interace 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\_FW\_INVALID\_SLOT\_INDEX** 0xE80F - invalid slot index  
**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\\_SLQSQosSwiReadApnExtraParams\\_t](#)
- struct [pack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#)
- struct [unpack\\_qosFlowStat\\_t](#)
- struct [unpack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosNWStatusCallback\\_ind\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosStatusCallback\\_ind\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosPriEventCallback\\_ind\\_t](#)
- struct [pack\\_qos\\_SLQSSetQosEventCallback\\_t](#)
- struct [unpack\\_qos\\_qosFlowInfoState\\_t](#)
- struct [unpack\\_qos\\_dataRate\\_t](#)
- struct [unpack\\_qos\\_tokenBucket\\_t](#)
- struct [unpack\\_qos\\_pktErrRate\\_t](#)
- struct [unpack\\_qos\\_swiQosFlow\\_t](#)
- struct [unpack\\_qos\\_IPv4Addr\\_t](#)
- struct [unpack\\_qos\\_Tos\\_t](#)
- struct [unpack\\_qos\\_IPv6Addr\\_t](#)
- struct [unpack\\_qos\\_IPv6TrafCls\\_t](#)
- struct [unpack\\_qos\\_Port\\_t](#)
- struct [unpack\\_qos\\_swiQosFilter\\_t](#)
- struct [unpack\\_qos\\_qosFlowInfo\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind\\_t](#)

## Macros

- `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`
- `#define LIBPACK_MAX_QOS_FILTERS 25`
- `#define LIBPACK_MAX_QOS_FLOWS 8`

## Functions

- int [pack\\_qos\\_SLQSQosGetNetworkStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_qos\\_SLQSQosGetNetworkStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSQosGetNetworkStatus\\_t](#) \*pOutput)
- int [pack\\_qos\\_SLQSQosSwiReadApnExtraParams](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_SLQSQosSwiReadApnExtraParams\\_t](#) reqParam)
- int [unpack\\_qos\\_SLQSQosSwiReadApnExtraParams](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSQosSwiReadApnExtraParams\\_t](#) \*pOutput)
- int [pack\\_qos\\_SLQSQosSwiReadDataStats](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#) reqParam)
- int [unpack\\_qos\\_SLQSQosSwiReadDataStats](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosNWStatusCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosNWStatusCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosStatusCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosStatusCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosPriEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosPriEventCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_qos\\_SLQSSetQosEventCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_SLQSSetQosEventCallback\\_t](#) reqParam)
- int [unpack\\_qos\\_SLQSSetQosEventCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind\\_t](#) \*pOutput)

## 9.23.1 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"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>

### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

### Note

- Timeout: 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"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of <i>pReqBuf</i></li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qos_SLQSQosSwiReadApnExtraParams_t</a> for more information</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

- Timeout: 2 seconds
  - PDN Specific: Yes

**9.23.2.3** `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"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of <i>pReqBuf</i></li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qos_SLQSQosSwiReadDataStats_t</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

- Timeout: 2 seconds
  - PDN Specific: Yes

9.23.2.4 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"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qos_SLQSSetQosEventCallback_t</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

- Timeout: 2 seconds
  - PDN Specific: Yes

9.23.2.5 `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"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">unpack_qos_SLQSQosGetNetworkStatus_t</a> for more information</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.23.2.6 `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"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">unpack_qos_SLQSQosSwiReadApnExtraParams_t</a> for more information</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.23.2.7 `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"><li>• Response from modem</li></ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"><li>• Length of pResp from modem</li></ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"><li>• See <a href="#">unpack_qos_SLQSQosSwiReadDataStats_t</a> for more information</li></ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.23.2.8 `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"><li>• Response from modem</li></ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"><li>• Length of pResp from modem</li></ul>

#### 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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosEventCallback_ind_t</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_NETWORK\_STATUS\_IND indication to identify this event from QOS service read function

9.23.2.10 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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosNWStatusCallback_ind_t</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise



## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

- Technology Supported: CDMA
- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_NETWORK\_STATUS\_IND indication to identify this event from QOS service read function

9.23.2.11 `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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosPriEventCallback_ind_t</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_PRIMARY\_QOS\_EVENT\_IND indication to identify this event from QOS service read function
- This is only generated when the primary flow is modified by the host

9.23.2.12 `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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosStatusCallback_ind_t</a> for more information</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_FLOW\_STATUS\_IND indication to identify this event from QOS service read function

## 9.24 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 [sSMSMessageMode](#) [sSMSMessageModelInfo](#)
- 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\\_SLQSDDeleteSMS](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSDDeleteSMS\\_t](#) \*p↵  
Output)
- 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\\_Set](#)↵  
[NewSMSCallback\\_t](#) reqParam)
- int [unpack\\_sms\\_SetNewSMSCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SetNewSMS](#)↵  
[Callback\\_t](#) \*Output)
- int [unpack\\_sms\\_SetNewSMSCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SetNewSMS](#)↵  
[Callback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SL](#)↵  
[QSWmsMemoryFullCallBack\\_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"> <li>• Number of sets of following element</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>• SMSC address</li> </ul>

9.24.2.2 typedef struct `sMSEtwsMessage` `sMSEtwsMessageInfo`

### Parameters

<i>notificationType</i>	<ul style="list-style-type: none"> <li>• Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS</li> </ul>
-------------------------	--

<i>length</i>	<ul style="list-style-type: none"> <li>Number of sets of following elements</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>Raw message data</li> </ul>

#### 9.24.2.3 typedef struct **sMSEtwsPlmn sMSEtwsPlmnInfo**

##### Parameters

<i>mobileCountry↔ Code</i>	<ul style="list-style-type: none"> <li>16 bit representation of MCC value range : 0 -999</li> </ul>
<i>mobile↔ NetworkCode</i>	<ul style="list-style-type: none"> <li>16 bit representation of MNC value range : 0 -999</li> </ul>

#### 9.24.2.4 typedef struct **sMSMessageMode sMSMessageModeInfo**

##### 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"> <li>Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK</li> </ul>
---------------------	---

<i>transactionID</i>	<ul style="list-style-type: none"> <li>Transaction ID of the message</li> </ul>
<i>format</i>	<ul style="list-style-type: none"> <li>Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC</li> </ul>
<i>length</i>	<ul style="list-style-type: none"> <li>Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes</li> </ul>
<i>data</i>	<ul style="list-style-type: none"> <li>Raw message data</li> </ul>

### 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

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

9.24.4.2 `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_SLQSMModifySMSStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,  
pack_sms_SLQSMModifySMSStatus_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\_SLQSMModifySMSStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSMModifySMSStatus\_t \* *pOutput* )

modify sms status unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.24.4.14 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

<code>in</code>	<code><i>pResp</i></code>	qmi response from modem
<code>in</code>	<code><i>respLen</i></code>	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\\_swioama\\_SLQSOMADMStartSession\\_t](#)
- struct [unpack\\_swioama\\_SLQSOMADMStartSession\\_t](#)
- struct [pack\\_swioama\\_SLQSOMADMCancelSession\\_t](#)
- struct [unpack\\_swioama\\_SLQSOMADMGetSettings\\_t](#)
- struct [pack\\_swioama\\_SLQSOMADMSetSettings\\_t](#)
- struct [pack\\_swioama\\_SLQSOMADMSelectSelection\\_t](#)
- struct [pack\\_swioama\\_SLQSOMADMGetSessionInfo\\_t](#)
- struct [unpack\\_swioama\\_SLQSOMADMGetSessionInfo\\_t](#)
- struct [unpack\\_omaDmFotaTlv\\_t](#)
- struct [unpack\\_omaDmConfigTlv\\_t](#)
- struct [unpack\\_omaDmNotificationsTlv\\_t](#)
- struct [unpack\\_swioama\\_SLQSOMADMAAlertCallback\\_ind\\_t](#)

### Macros

- `#define LIBPACK_MAX_SWIOMA_STR_LEN 255`

## Functions

- int [pack\\_swisma\\_SLQSOMADMStartSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swisma\\_SLQSOMADMStartSession\\_t](#) reqParam)
- int [unpack\\_swisma\\_SLQSOMADMStartSession](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swisma\\_SLQSOMADMStartSession\\_t](#) \*pOutput)
- int [pack\\_swisma\\_SLQSOMADMCancelSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swisma\\_SLQSOMADMCancelSession\\_t](#) reqParam)
- int [unpack\\_swisma\\_SLQSOMADMCancelSession](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_swisma\\_SLQSOMADMGetSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swisma\\_SLQSOMADMGetSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swisma\\_SLQSOMADMGetSettings\\_t](#) \*pOutput)
- int [pack\\_swisma\\_SLQSOMADMSetSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swisma\\_SLQSOMADMSetSettings\\_t](#) reqParam)
- int [unpack\\_swisma\\_SLQSOMADMSetSettings](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_swisma\\_SLQSOMADMSendSelection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swisma\\_SLQSOMADMSendSelection\\_t](#) reqParam)
- int [unpack\\_swisma\\_SLQSOMADMSendSelection](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_swisma\\_SLQSOMADMGetSessionInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swisma\\_SLQSOMADMGetSessionInfo\\_t](#) reqParam)
- int [unpack\\_swisma\\_SLQSOMADMGetSessionInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swisma\\_SLQSOMADMGetSessionInfo\\_t](#) \*pOutput)
- int [pack\\_swisma\\_SLQSOMADMAAlertCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swisma\\_SLQSOMADMAAlertCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_swisma\\_SLQSOMADMAAlertCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swisma\\_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_swisma_SLQSOMADMAAlertCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

#### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>



## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

9.27.2.2 `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"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swioma_SLQSOMADMCancelSession_t</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

9.27.2.3 `int pack_swima_SLQSOMADMGetSessionInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_SLQSOMADMGetSessionInfo_t reqParam )`

Function to pack QMI command to return information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

#### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swima_SLQSOMADMGetSessionInfo_t</a> for more information</li> </ul>

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

Timeout: 20 seconds

9.27.2.4 `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"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
	Generated by Doxygen

## 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"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swioma_SLQSOMADMSendSelection_t</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

9.27.2.6 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"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_swima_SLQSOMADMSetSettings_t</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

Timeout: 20 seconds

**9.27.2.7** `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"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_swima_SLQSOMADMStartSession_t</a> for more information</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

## Note

Timeout: 20 seconds

### 9.27.2.8 int unpack\_swioma\_SLQSOMADMAAlertCallback ( uint8\_t \* pResp, uint16\_t respLen )

Function to unpack response of QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>

## 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"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN] Generated by Doxygen	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">unpack_swioma_SLQSOMADMAAlertCallback_ind_t</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.27.2.10 `int unpack_swima_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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

**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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swima_SLQSOMADMGetSessionInfo_t</a> for more information</li> </ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.27.2.12 `int unpack_swioma_SLQSOMADMGetSettings ( uint8_t * pResp, uint16_t respLen, unpack_swioma_SLQSOMADMGetSettings_t * pOutput )`

Function to unpack OMA-DM get settings response from modem This maps to SLQSOMADMGetSettings2

**Parameters**

<i>pResp[IN]</i>	<ul style="list-style-type: none"><li>• Response from modem</li></ul>
<i>respLen[IN]</i>	<ul style="list-style-type: none"><li>• Length of pResp from modem</li></ul>
<i>pOutput[OUT]</i>	<ul style="list-style-type: none"><li>• See <a href="#">unpack_swioma_SLQSOMADMGetSettings_t</a> for more information</li></ul>

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.27.2.13 `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[IN]</i>	<ul style="list-style-type: none"><li>• Response from modem</li></ul>
<i>respLen[IN]</i>	<ul style="list-style-type: none"><li>• Length of pResp from modem</li></ul>

**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_swima_SLQSOMADMSetSettings ( uint8_t * pResp, uint16_t respLen )`

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

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"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swima_SLQSOMADMStartSession_t</a> for more information</li> </ul>

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values



## 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\\_SLQSUIEventRegister\\_t](#)
- struct [unpack\\_uim\\_SLQSUIEventRegister\\_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\\_SLQSUIPowerUp\\_t](#)
- struct [pack\\_uim\\_SLQSUIPowerDown\\_t](#)

### Macros

- [#define UIM\\_UINT8\\_MAX\\_STRING\\_SZ 255](#)
- [#define UIM\\_MAX\\_DESCRIPTION\\_LENGTH 255](#)
- [#define UIM\\_MAX\\_NO\\_OF\\_SLOTS 5](#)
- [#define UIM\\_MAX\\_NO\\_OF\\_APPLICATIONS 10](#)
- [#define MAX\\_NO\\_OF\\_SLOTS 5](#)
- [#define MAX\\_NO\\_OF\\_APPLICATIONS 10](#)
- [#define MAX\\_DESCRIPTION\\_LENGTH 255](#)
- [#define MAX\\_SLOTS\\_STATUS 255](#)
- [#define MAX\\_ICCID\\_LENGTH 255](#)

## Functions

- int [pack\\_uim\\_GetCardStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_uim\\_GetCardStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_GetCardStatus\\_t](#) \*pOutput)
- int [pack\\_uim\\_VerifyPin](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_VerifyPin\\_t](#) \*reqArg)
- int [unpack\\_uim\\_VerifyPin](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_VerifyPin\\_t](#) \*pOutput)
- int [pack\\_uim\\_UnblockPin](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_UnblockPin\\_t](#) \*reqArg)
- int [unpack\\_uim\\_UnblockPin](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_UnblockPin\\_t](#) \*pOutput)
- int [pack\\_uim\\_SetPinProtection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SetPinProtection\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SetPinProtection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SetPinProtection\\_t](#) \*pOutput)
- int [pack\\_uim\\_ChangePin](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_ChangePin\\_t](#) \*reqArg)
- int [unpack\\_uim\\_ChangePin](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_ChangePin\\_t](#) \*pOutput)
- int [pack\\_uim\\_ReadTransparent](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_ReadTransparent\\_t](#) \*reqArg)
- int [unpack\\_uim\\_ReadTransparent](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_ReadTransparent\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMEventRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMEventRegister\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMEventRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMEventRegister\\_t](#) \*pOutput)
- int [unpack\\_uim\\_SLQSUIMSetStatusChangeCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMSetStatusChangeCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMGetSlotsStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_uim\\_SLQSUIMGetSlotsStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMGetSlotsStatus\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMSwitchSlot](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMSwitchSlot\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMSwitchSlot](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMPowerUp](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMPowerUp\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMPowerUp](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_uim\\_SLQSUIMPowerDown](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMPowerDown\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMPowerDown](#) (uint8\_t \*pResp, uint16\_t respLen)

## 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\_SLQSUIEventRegister ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen,  
pack\_uim\_SLQSUIEventRegister\_t \* reqArg )

UIM Status Change callback enable pack

## Parameters

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

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.29.2.6 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\_SLQSUIEventRegister to subscribe

**9.29.2.17** int unpack\_uim\_SLQSUIEventRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIEvent↔  
Register\_t \* *pOutput* )

UIM Status Change callback enable unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.29.2.18 int unpack\_uim\_SLQSUIGetSlotsStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SLQSUIGetSlotsStatus\_t \* *pOutput* )

get slot status unpack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.29.2.19 int unpack\_uim\_SLQSUIPowerDown ( 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\_SLQSUIPowerUp ( 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\_SLQSUIMEventRegister to subscribe

**9.29.2.22** int unpack\_uim\_SLQSUIMSwitchSlot ( uint8\_t \* *pResp*, uint16\_t *respLen* )

switch slot unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.29.2.23 `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\\_SLQSSStartDataSession\\_t](#)
- struct [unpack\\_wds\\_SLQSSStartDataSession\\_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 [rmTrasferStaticsReq](#)

- struct [pack\\_wds\\_RMSetTransferStatistics\\_t](#)
- struct [unpack\\_wds\\_RMSetTransferStatistics\\_t](#)
- struct [pack\\_wds\\_SetMobileIPProfile\\_t](#)
- struct [unpack\\_wds\\_SetMobileIPProfile\\_t](#)
- struct [pack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings\\_t](#)
- struct [ipv6AddressInfo](#)
- struct [unpack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings\\_t](#)
- struct [transferStatInd](#)
- struct [pack\\_wds\\_SLQSGetDUNCallInfo\\_t](#)
- struct [connectionStatus](#)
- struct [dunchannelRate](#)
- struct [unpack\\_wds\\_SLQSGetDUNCallInfo\\_t](#)
- struct [qmiWSDDataBearerTechnology](#)
- struct [unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t](#)
- struct [pack\\_wds\\_SLQSGetDataBearerTechnology\\_t](#)
- struct [pack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#)
- struct [unpack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#)
- struct [pack\\_wds\\_SetDefaultProfileNum\\_t](#)
- struct [pack\\_wds\\_GetDefaultProfileNum\\_t](#)
- struct [unpack\\_wds\\_GetDefaultProfileNum\\_t](#)
- struct [wdsDhcpv4ProfileId](#)
- struct [wdsDhcpv4HwConfig](#)
- struct [wdsDhcpv4Option](#)
- struct [wdsDhcpv4OptionList](#)
- struct [pack\\_wds\\_SLQSSGetDHCPv4ClientConfig\\_t](#)
- struct [unpack\\_wds\\_SLQSSGetDHCPv4ClientConfig\\_t](#)
- struct [pack\\_wds\\_GetPacketStatistics\\_t](#)
- struct [unpack\\_wds\\_GetPacketStatistics\\_t](#)
- struct [unpack\\_wds\\_GetByteTotals\\_t](#)
- struct [unpack\\_wds\\_SLQSGetCurrentChannelRate\\_t](#)
- struct [unpack\\_wds\\_SLQSSGetLoopback\\_t](#)
- struct [pack\\_wds\\_SLQSSSetLoopback\\_t](#)
- struct [wds\\_DataULongTlv](#)
- struct [wds\\_DataULongLongTlv](#)
- struct [unpack\\_RMTransferStatistics\\_ind\\_t](#)
- struct [pack\\_wds\\_DHCPv4ClientLeaseChange\\_t](#)
- struct [wds\\_DHCPProfileIdTlv](#)
- struct [wds\\_DHCPLeaseStateTlv](#)
- struct [wds\\_IPv4AdTlv](#)
- struct [wds\\_DHCPOpt](#)
- struct [wds\\_DHCPLeaseOptTlv](#)
- struct [unpack\\_wds\\_DHCPv4ClientLease\\_ind\\_t](#)
- struct [pack\\_wds\\_SetMobileIP\\_t](#)
- struct [pack\\_wds\\_SetMobileIPParameters\\_t](#)
- struct [pack\\_wds\\_SetAutoconnect\\_t](#)
- struct [unpack\\_wds\\_GetAutoconnect\\_t](#)
- struct [wds\\_TrStatInd](#)
- struct [pack\\_wds\\_SLQSWdsSetEventReport\\_t](#)
- struct [wds\\_DHCPv4ProfileId](#)
- struct [wds\\_DHCPv4HWConfig](#)
- struct [wds\\_DHCPv4Option](#)
- struct [wds\\_DHCPv4OptionList](#)
- struct [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t](#)
- struct [unpack\\_wds\\_GetDataBearerTechnology\\_t](#)

## Macros

- `#define IPV6_ADDRESS_ARRAY_SIZE 8`
- `#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24`
- `#define PACK_WDS_IPV4 4`
- `#define PACK_WDS_IPV6 6`
- `#define BYT_STAT_STAT_MASK 0X000000C0`
- `#define WDS_DHCP_MAX_NUM_OPTIONS 30`
- `#define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */`

## Typedefs

- typedef union `unpackWdsProfileParam UnpackQmiProfileInfo`

## Functions

- `int pack_wds_SLQSSStartDataSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSStartDataSession_t *reqArg)`
- `int unpack_wds_SLQSSStartDataSession (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSStartDataSession_t *pOutput)`
- `int unpack_wds_SLQSSetPacketSrvStatusCallback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetPacketSrvStatusCallback_t *pOutput)`
- `int pack_wds_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\\_SLQSSetDUNCallInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetDUNCallInfo\\_t](#) \*reqParam)
- int [unpack\\_wds\\_SLQSSetDUNCallInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetDUNCallInfo\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetDataBearerTechnology](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetDataBearerTechnology\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_SLQSSetDataBearerTechnology](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetDataBearerTechnology\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetIPFamilyPreference](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_SLQSSetIPFamilyPreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#) \*pOutput)
- int [pack\\_wds\\_SetDefaultProfileNum](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetDefaultProfileNum\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_SetDefaultProfileNum](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_GetDefaultProfileNum](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetDefaultProfileNum\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_GetDefaultProfileNum](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetDefaultProfileNum\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetDHCPv4ClientConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#) \*pReq)
- int [unpack\\_wds\\_SLQSSetDHCPv4ClientConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#) \*pOutput)
- 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\\_SLQSSetCurrentChannelRate](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSSetCurrentChannelRate](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetCurrentChannelRate\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetLoopback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSSetLoopback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetLoopback\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetLoopback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetLoopback\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSSetLoopback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_wds\\_RMTransferStatistics\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_RMTransferStatistics\\_ind\\_t](#) \*pOutput)
- int [pack\\_wds\\_DHCPv4ClientLeaseChange](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_DHCPv4ClientLeaseChange\\_t](#) \*reqArg)
- int [unpack\\_wds\\_DHCPv4ClientLeaseChange](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_wds\\_DHCPv4ClientLease\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_DHCPv4ClientLease\\_ind\\_t](#) \*pOutput)
- int [pack\\_wds\\_SetMobileIP](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetMobileIP\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SetMobileIP](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SetMobileIPParameters](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetMobileIPParameters\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SetMobileIPParameters](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SetAutoconnect](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetAutoconnect\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SetAutoconnect](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_GetAutoconnect](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)

- int [unpack\\_wds\\_GetAutoconnect](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetAutoconnect\\_t](#) \*p↔  
Output)
- int [pack\\_wds\\_SLQSWdsSetEventReport](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_↔  
\\_SLQSWdsSetEventReport\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSWdsSetEventReport](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SLQSWdsGoDormant](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSWdsGoDormant](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SLQSWdsGoActive](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSWdsGoActive](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SLQSRResetPacketStatics](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSRResetPacketStatics](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_↔  
\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSSSetDHCPv4ClientConfig](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_GetDataBearerTechnology](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_GetDataBearerTechnology](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetData\\_↔  
BearerTechnology\\_t](#) \*pOutput)

### 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`

9.30.1.6 `#define WDS_DHCP_MAX_NUM_OPTIONS 30`

9.30.1.7 `#define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */`

### 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_DHCPv4ClientLeaseChange ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,  
pack_wds_DHCPv4ClientLeaseChange_t * reqArg )`

DHCPv4 lease state changes pack

#### Parameters

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

9.30.3.2 `int pack_wds_GetAutoconnect ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets auto connect data session setting pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

PDN Specific: Yes

9.30.3.3 `int pack_wds_GetByteTotals ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get Rx/Tx byte counts since the start of the last packet data session pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>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_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

9.30.3.5 `int pack_wds_GetDataBearerTechnology ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get current data bearer technology pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.6 `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

9.30.3.7 `int pack_wds_GetDefaultProfileNum ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDefaultProfileNum_t * pReqParam )`

get default profile number pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.8 `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

9.30.3.9 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

9.30.3.10 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

9.30.3.11 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

9.30.3.12 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

**9.30.3.13** 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

**9.30.3.14** 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

9.30.3.15 `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

9.30.3.16 `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

```
9.30.3.17 int pack_wds_SetAutoconnect ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
    pack_wds_SetAutoconnect_t * reqArg )
```

Auto connect data session parameters pack

**Parameters**

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

9.30.3.19 `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

9.30.3.20 `int pack_wds_SetMobileIP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SetMobileIP_t * reqArg )`

Sets the current mobile IP setting pack

#### Parameters

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

9.30.3.21 `int pack_wds_SetMobileIPParameters ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SetMobileIPParameters_t * reqArg )`

Sets the specified mobile IP parameters pack

#### Parameters

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

9.30.3.22 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

9.30.3.23 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

9.30.3.24 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

9.30.3.25 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

9.30.3.26 `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

9.30.3.27 `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

```
9.30.3.28 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

```
9.30.3.29 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

9.30.3.30 `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

9.30.3.31 `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

9.30.3.32 `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

9.30.3.33 int pack\_wds\_SLQSResetPacketStatics ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Reset packet data transfer statistics pack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**Note**

PDN Specific: Yes

9.30.3.34 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

**9.30.3.35** 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

**9.30.3.36** 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

9.30.3.37 int pack\_wds\_SLQSSGetDHCPv4ClientConfig ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSSGetDHCPv4ClientConfig\_t \* *pReq* )

get DHCPv4 Client Config pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>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

9.30.3.38 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

```
9.30.3.39 int pack_wds_SLQSSSetDHCPv4ClientConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
pack_wds_SLQSSSetDHCPv4ClientConfig_t * reqArg )
```

Gets the DHCP Client V4 Configuration pack

## Parameters

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

```
9.30.3.40 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

9.30.3.41 `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

9.30.3.42 `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

9.30.3.43 `int pack_wds_SLQSWdsGoActive ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets the device into Active state pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

PDN Specific: Yes

9.30.3.44 `int pack_wds_SLQSWdsGoDormant ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets the device into dormant state pack

#### Parameters

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

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### Note

PDN Specific: Yes

9.30.3.45 `int pack_wds_SLQSWdsSetEventReport ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSWdsSetEventReport_t * reqArg )`

Sets the event report parameters pack

#### Parameters

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

9.30.3.46 **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

9.30.3.47 **int** unpack\_wds\_DHCPv4ClientLease\_ind ( **uint8\_t** \* *pResp*, **uint16\_t** *respLen*, **unpack\_wds\_DHCPv4ClientLease\_ind\_t** \* *pOutput* )

DHCP lease state has changed indication unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.30.3.48 int unpack\_wds\_DHCPv4ClientLeaseChange ( uint8\_t \* *pResp*, uint16\_t *respLen* )

DHCPv4 lease state changes 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\_GetAutoconnect ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetAutoconnect\_t \* *pOutput* )

Gets auto connect data session setting unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values



9.30.3.50 `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

9.30.3.51 `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

9.30.3.52 `int unpack_wds_GetDataBearerTechnology ( uint8_t * pResp, uint16_t respLen, unpack_wds_GetDataBearerTechnology_t * pOutput )`

get current data bearer technology unpack

#### Parameters

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.30.3.53 `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

9.30.3.54 `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

9.30.3.55 `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

9.30.3.56 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

9.30.3.57 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

9.30.3.58 `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

9.30.3.59 `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

9.30.3.60 `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

**9.30.3.61** 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

**9.30.3.62** 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

9.30.3.63 `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

9.30.3.64 `int unpack_wds_RMTransferStatistics_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_RMTransferStatistics_ind_t * pOutput )`

RM transfer statistics indication unpack

Parameters

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

Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.30.3.65 `int unpack_wds_SetAutoconnect ( uint8_t * pResp, uint16_t respLen )`

Auto connect data session parameters unpack

Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.30.3.66 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

**9.30.3.67 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

**9.30.3.68** `int unpack_wds_SetMobileIP ( uint8_t * pResp, uint16_t respLen )`

Sets the current mobile IP setting 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

**9.30.3.69** `int unpack_wds_SetMobileIPParameters ( uint8_t * pResp, uint16_t respLen )`

Sets the specified mobile IP parameters unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

**9.30.3.70** `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

**9.30.3.71** `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

**9.30.3.72** `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

**9.30.3.73** `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

9.30.3.74 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

9.30.3.75 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

**9.30.3.76** 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

**9.30.3.77** 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

**9.30.3.78** 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
Generated by Doxygen	<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

**9.30.3.79** 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

**9.30.3.80** 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

**9.30.3.81** int unpack\_wds\_SLQSResetPacketStatics ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Reset packet data 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

**9.30.3.82** 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

**9.30.3.83** 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

**9.30.3.84** 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

**9.30.3.85 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

**9.30.3.86 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

9.30.3.87 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

9.30.3.88 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

9.30.3.89 int unpack\_wds\_SLQSSSetDHCPv4ClientConfig ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Gets the DHCP Client V4 Configuration 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

9.30.3.90 int unpack\_wds\_SLQSSetLoopback ( 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

9.30.3.91 int unpack\_wds\_SLQSStartDataSession ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSStartData↵  
Session\_t \* *pOutput* )

start data session unpack

**Parameters**

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

**Returns**

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

**See also**

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.30.3.92 int unpack\_wds\_SLQSStopDataSession ( uint8\_t \* *pResp*, uint16\_t *respLen* )

stop data session unpack



## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

## See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

#### 9.30.3.93 int unpack\_wds\_SLQSWdsGoActive ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Gets the device into Active state 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

#### 9.30.3.94 int unpack\_wds\_SLQSWdsGoDormant ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Gets the device into dormant state 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

9.30.3.95 int unpack\_wds\_SLQSWdsSetEventReport ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Sets the event report parameters unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_xxx error value otherwise

#### See also

See [qmerrno.h](#) for eQCWWAN\_xxx error values

9.30.3.96 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, 456](#)
  - [\\_\\_GOBI\\_API\\_CODING\\_SCHEME\\_H\\_\\_](#)
    - [qaGobiApiTableCodingScheme.h, 636](#)
  - [\\_libSDP\\_FirmwareInfo\\_, 23](#)
    - [szCarrier\\_str, 24](#)
    - [szCarrierPriversion\\_str, 24](#)
    - [szFwversion\\_str, 24](#)
    - [szModelid\\_str, 24](#)
    - [szPackageid\\_str, 24](#)
    - [szSku\\_str, 24](#)
- [AAASPI](#)
  - [unpack\\_wds\\_GetMobileIPProfile\\_t, 449](#)
- [AAASState](#)
  - [unpack\\_wds\\_GetMobileIPProfile\\_t, 449](#)
- [ACT\\_CODE\\_MAX\\_SIZE](#)
  - [dms.h, 500](#)
- [AMSSString](#)
  - [unpack\\_dms\\_GetFirmwareRevision\\_t, 315](#)
  - [unpack\\_dms\\_GetFirmwareRevisions\\_t, 315](#)
- [APNName](#)
  - [unpack\\_wds\\_SLQSGetRuntimeSettings\\_t, 461](#)
- [accelTemp](#)
  - [pack\\_loc\\_SLQSLOCInjectSensorData\\_t, 214](#)
- [acceleroData](#)
  - [pack\\_loc\\_SLQSLOCInjectSensorData\\_t, 214](#)
- [acceleroTimeSrc](#)
  - [pack\\_loc\\_SLQSLOCInjectSensorData\\_t, 214](#)
- [accolc](#)
  - [pack\\_nas\\_SetACCOLC\\_t, 219](#)
- [ackIndicator](#)
  - [sMSTransferRouteMTMessage, 289](#)
- [acqOrdeLen](#)
  - [nas\\_acqOrderPref, 87](#)
- [acroamsetting](#)
  - [pack\\_wds\\_SetAutoconnect\\_t, 258](#)
- [acsetting](#)
  - [pack\\_wds\\_SetAutoconnect\\_t, 258](#)
- [actCode](#)
  - [pack\\_dms\\_ActivateAutomatic\\_t, 189](#)
- [activationState](#)
  - [pack\\_dms\\_SetActivationStatusCallback\\_t, 190](#)
- [activationStatus](#)
  - [dms\\_ActivationStatusTlv, 33](#)
- [ActivationStatusTlv](#)
  - [unpack\\_dms\\_SetEventReport\\_ind\\_t, 324](#)
- [activeBandClass](#)
  - [nas\\_RFInfoTlv, 147](#)
  - [RFBandInfoElements, 279](#)
- [activeChannel](#)
  - [nas\\_RFInfoTlv, 147](#)
  - [RFBandInfoElements, 279](#)
- [ActiveTechPref](#)
  - [unpack\\_nas\\_GetNetworkPreference\\_t, 367](#)
- [addr](#)
  - [unpack\\_qos\\_IPv4Addr\\_t, 402](#)
  - [unpack\\_qos\\_IPv6Addr\\_t, 403](#)
- [address](#)
  - [unpack\\_wds\\_GetMobileIPProfile\\_t, 449](#)
- [aid](#)
  - [uim\\_UIMSessionInformation, 306](#)
  - [uim\\_sessionInformation, 303](#)
- [aidLength](#)
  - [appStats, 27](#)
  - [uim\\_UIMSessionInformation, 306](#)
  - [uim\\_appStatus, 295](#)
  - [uim\\_sessionInformation, 303](#)
- [aidVal](#)
  - [appStats, 27](#)
  - [uim\\_appStatus, 295](#)
- [aidingIndicatorMask](#)
  - [loc\\_sensorDataUsage, 83](#)
- [alertmsg](#)
  - [unpack\\_omaDmConfigTlv\\_t, 398](#)
- [alertmsglength](#)
  - [unpack\\_omaDmConfigTlv\\_t, 398](#)
- [altSrcInfo\\_t, 24](#)
  - [coverage, 25](#)
  - [linkage, 25](#)
  - [source, 25](#)
- [altitudeAssumed](#)
  - [unpack\\_loc\\_GnssSvInfo\\_Ind\\_t, 354](#)
- [altitudeSrcInfo](#)
  - [pack\\_loc\\_SLQSLOCInjectPosition\\_t, 211](#)
- [altitudeWrtEllipsoid](#)
  - [pack\\_loc\\_SLQSLOCInjectPosition\\_t, 211](#)
- [altitudeWrtMeanSeaLevel](#)
  - [pack\\_loc\\_SLQSLOCInjectPosition\\_t, 212](#)
- [ambr\\_dl](#)
  - [unpack\\_qos\\_SLQSQosSviReadApnExtra↔ Params\\_t, 409](#)
- [ambr\\_dl\\_ext](#)
  - [unpack\\_qos\\_SLQSQosSviReadApnExtra↔ Params\\_t, 409](#)
- [ambr\\_dl\\_ext2](#)
  - [unpack\\_qos\\_SLQSQosSviReadApnExtra↔ Params\\_t, 409](#)
- [ambr\\_ul](#)

- unpack\_qos\_SLQSQosSwiReadApnExtra↔  
Params\_t, 409
- ambr\_ul\_ext
  - unpack\_qos\_SLQSQosSwiReadApnExtra↔  
Params\_t, 409
- ambr\_ul\_ext2
  - unpack\_qos\_SLQSQosSwiReadApnExtra↔  
Params\_t, 409
- amssSize
  - unpack\_dms\_GetFirmwareRevision\_t, 315
  - unpack\_dms\_GetFirmwareRevisions\_t, 315
- apdoxypages.c, 489
- apnId
  - pack\_qos\_SLQSQosSwiReadApnExtraParams\_t,  
237
  - pack\_qos\_SLQSQosSwiReadDataStats\_t, 237
  - unpack\_qos\_SLQSQosSwiReadApnExtra↔  
Params\_t, 409
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- apnName
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔  
\_t, 468
- apnname
  - unpack\_wds\_GetDefaultProfile\_t, 446
- apnsize
  - unpack\_wds\_GetDefaultProfile\_t, 446
- appNameLength
  - loc\_LocApplicationInfo, 79
- appProviderLength
  - loc\_LocApplicationInfo, 79
- appState
  - appStats, 27
  - uim\_appStatus, 295
- appStats, 25
  - aidLength, 27
  - aidVal, 27
  - appState, 27
  - appType, 28
  - persoFeature, 28
  - persoRetries, 28
  - persoState, 28
  - persoUnblockRetries, 28
  - pin1Retries, 28
  - pin1State, 28
  - pin2Retries, 28
  - pin2State, 28
  - puk1Retries, 28
  - puk2Retries, 28
  - univPin, 28
- AppStatus
  - slotInf, 284
  - uim\_slotInfo, 305
- appType
  - appStats, 28
  - uim\_appStatus, 295
- appVersionLength
  - loc\_LocApplicationInfo, 79
- appVersionValid
  - loc\_LocApplicationInfo, 79
- Application
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- appversion\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- arfcn
  - nas\_GERANInfo, 107
  - nas\_gsmCellInfo, 109
- auth
  - unpack\_wds\_GetDefaultProfile\_t, 446
- Authentication
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- authentication
  - pack\_wds\_SetDefaultProfile\_t, 259
- Autosdm
  - unpack\_swima\_SLQSOMADMGetSettings\_t, 435
- avgPeriod
  - nas\_LTESigRptConfig, 130
- azimuth
  - loc\_satelliteInfo, 82
- bEnable
  - pack\_dms\_UIMSetPINProtection\_t, 199
  - pack\_nas\_SLQSSetSignalStrengthsCallback\_t,  
231
- bForceDownload
  - pack\_fms\_SetImagesPreference\_t, 203
- bICCIDLength
  - slot\_t, 282
- bICCID
  - slot\_t, 282
- bLogicalSlot
  - pack\_uim\_SLQSUIMSwitchSlot\_t, 253
  - slot\_t, 282
- bNumberOfPhySlots
  - unpack\_uim\_SetUimSlotStatusChangeCallback↔  
\_ind\_t, 440
- BOOL
  - SwiDataTypes.h, 677
- BPTlv
  - NASQmiCbkNasSystemSelPrefInd, 185
- bResetStatistics
  - rmTrasnferStaticsReq, 280
- BYT\_STAT\_STAT\_MASK
  - wds.h, 706
- BYTE
  - SwiDataTypes.h, 677
- band
  - nas\_LTEInfo, 122
- band1900
  - nas\_gsmCellInfo, 109
- band\_pref
  - NASBandPreferenceTlv, 176
- BandCapability
  - unpack\_dms\_GetBandCapability\_t, 309
- bandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, 329
- bandwidth
  - nas\_LTEInfo, 122

- baseId
  - nas\_CDMAInfo, 92
  - nas\_CDMA SysInfo, 95
- baseLat
  - nas\_CDMAInfo, 92
  - nas\_CDMA SysInfo, 95
- baseLong
  - nas\_CDMAInfo, 92
  - nas\_CDMA SysInfo, 95
- BasestationID
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- BasestationLatitude
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- BasestationLongitude
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- BearerID
  - unpack\_qos\_QosFlowInfo\_t, 406
- bearerID
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback↔\_t, 463
- bearerId
  - unpack\_QosFlowStat\_t, 423
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔\_t, 468
- bootSize
  - unpack\_dms\_GetFirmwareRevisions\_t, 315
- BootString
  - unpack\_dms\_GetFirmwareRevisions\_t, 315
- bootversion\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- Broadcast
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- bsInfoValid
  - nas\_CDMA SysInfo, 95
- bsPRev
  - nas\_CDMA SysInfo, 95
- bsPRevValid
  - nas\_CDMA SysInfo, 96
- bsic
  - nas\_GERANInfo, 107
- bsicId
  - nas\_gsmCellInfo, 109
- bucketSz
  - unpack\_qos\_tokenBucket\_t, 422
- buildIDLen
  - image\_info\_t, 43
- buildIDLength
  - FMSImageIdElement, 39
- buildID
  - FMSImageIdElement, 39
  - image\_info\_t, 43
- buildId
  - FMSImageElement, 38
- buildIdLength
  - FMSImageElement, 38
- ByteLoopbackMode
  - unpack\_wds\_SLQSSGetLoopback\_t, 466
- ByteLoopbackMultiplier
  - unpack\_wds\_SLQSSGetLoopback\_t, 466
- CDMA\_P\_Rev
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- CDMAECIOThreshListLen
  - nas\_CDMAECIOThresh, 91
- CDMARSSIOThreshListLen
  - nas\_CDMARSSIOThresh, 92
- CDMA SSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 388
- CDMA SystemInfoExt
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- CHAR
  - SwiDataTypes.h, 677
- CK\_MAX\_SIZE
  - dms.h, 500
- CQIValueCW0
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 393
- CQIValueCW1
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 393
- CSDomain
  - unpack\_nas\_GetServingNetwork\_t, 368
- CallBarStatus
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- callDuration
  - unpack\_wds\_GetSessionDuration\_t, 453
- callEndReason
  - unpack\_wds\_SLQSSGetDUNCallInfo\_t, 459
- cardState
  - slotInf, 284
  - uim\_slotInfo, 305
- carrier
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 332
- carrier\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- CarrierImage\_t, 28
  - m\_FwBuildId, 29
  - m\_FwImageId, 29
  - m\_PriBuildId, 29
  - m\_PriImageId, 29
  - m\_nCarrierId, 29
  - m\_nFolderId, 29
  - m\_nStorage, 29
- ccsSupported
  - nas\_CDMA SysInfo, 96
- ccsSupportedValid
  - nas\_CDMA SysInfo, 96
- cdma SSInfo, 29
  - ecio, 30
  - rsi, 30
- cdmaSysIdValid
  - nas\_CDMA SysInfo, 96
- cell\_resel\_priority
  - nas\_infoInterFreq, 119
- cellBroadcastCap
  - nas\_AddSysInfo, 88
- CellID
  - unpack\_nas\_SLQSSGetServingSystem\_t, 378
- cellID

- nas\_GERANInfo, 107
- nas\_UMTSInfo, 165
- cellId
  - nas\_GSMsSysInfo, 113
  - nas\_LTEsSysInfo, 133
  - nas\_WCDMASysInfo, 175
- cellIdValid
  - nas\_GSMsSysInfo, 113
  - nas\_LTEsSysInfo, 133
  - nas\_WCDMASysInfo, 175
  - nas\_gsmCellInfo, 109
- cellInterFreqParams
  - nas\_infoInterFreq, 119
- cellsTDD
  - nas\_umtsLTENbrCell, 168
- CellParams
  - nas\_LTEInfoIntrafreq, 125
- cellReselPriority
  - nas\_LTEInfoIntrafreq, 125
  - nas\_lteGsmCellInfo, 120
  - nas\_lteWcdmaCellInfo, 135
- cells\_len
  - nas\_infoInterFreq, 119
  - nas\_lteGsmCellInfo, 120
- cellsLen
  - nas\_LTEInfoIntrafreq, 125
  - nas\_lteWcdmaCellInfo, 135
- chaddr
  - wds\_DHCPv4HWConfig, 474
  - wdsDhcpv4HwConfig, 485
- chaddrLen
  - wds\_DHCPv4HWConfig, 474
  - wdsDhcpv4HwConfig, 485
- changePIN
  - pack\_uim\_ChangePin\_t, 248
- channelRate
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- clear
  - pack\_dms\_SLQSSwiGetCrashInfo\_t, 194
- common.h, 489
  - DEAULT\_LOC\_TIMEOUT\_IN\_SEC, 491
  - eCAT, 492
  - eCTL, 492
  - eDMS, 492
  - eIND, 492
  - eLOG\_DEBUG, 491
  - eLOG\_FATAL, 491
  - eLOG\_INFO, 491
  - eLOG\_LEVEL, 491
  - eLOG\_WARN, 491
  - eLOC, 492
  - eNAS, 492
  - eQMI\_SVC, 491
  - eQOS, 492
  - eREQ, 492
  - eRSP, 492
  - eSMS, 492
  - eSWILOC, 492
  - eSWIOMA, 492
  - eTIMEOUT\_10\_S, 492
  - eTIMEOUT\_20\_S, 492
  - eTIMEOUT\_2\_S, 492
  - eTIMEOUT\_300\_S, 492
  - eTIMEOUT\_30\_S, 492
  - eTIMEOUT\_5\_S, 492
  - eTIMEOUT\_60\_S, 492
  - eTIMEOUT\_8\_S, 492
  - eTIMEOUT\_DEFAULT, 492
  - eTMD, 492
  - eTimeout, 492
  - eUIM, 492
  - eWDS, 492
  - fill\_pack\_ctx, 493
  - fill\_sdu\_hdr, 493
  - get\_version, 493
  - glog, 494
  - gloglvl, 494
  - helper\_get\_resp\_ctx, 493
  - helper\_get\_xid, 493
  - helper\_set\_log\_func, 493
  - helper\_set\_log\_lvl, 493
  - libpack\_GetVersion, 493
  - libpack\_log, 493
  - logger, 491
  - MINREQBKLEN, 491
  - MSGID\_AND\_LEN, 491
  - MSGID\_DONT\_CARE, 491
  - msgtype, 492
  - SDK\_VALIDATE\_INPUT\_PACK\_PARAM\_AND←\_FILL\_XID, 491
  - SDK\_VALIDATE\_INPUT\_PACK\_PARAM, 491
  - SDU\_HDR\_LEN, 491
  - UNUSEDPARAM, 491
  - unpack\_result\_code\_only, 494
- commonInfo
  - unpack\_nas\_SLQSNasSwiModemStatus\_t, 391
- ConcSvcInfo
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- confidence
  - pack\_loc\_SLQSLOCSetCradleMountConfig\_t, 216
- conn\_status
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback←\_t, 463
- connectionStatus, 30
  - MDMCallDuration, 30
  - MDMConnStatus, 30
  - unpack\_wds\_GetSessionState\_t, 453
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- content
  - uim\_readResult, 301
- contentLen
  - uim\_readResult, 301
- contextId
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 276

- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔  
\_t, 468
- contextType
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t,  
276
- coverage
  - altSrcInfo\_t, 25
- cpich\_ecno
  - nas\_wcdmaCellInfo, 170
- cpich\_rscp
  - nas\_wcdmaCellInfo, 170
- crashAction
  - pack\_dms\_SetCrashAction\_t, 191
- crashData
  - crashInformation, 32
- crashId
  - crashInformation, 32
- crashInfo
  - crashInfoParams, 31
- crashInfoParam
  - unpack\_dms\_SLQSSwiGetCrashInfo\_t, 330
- crashInfoParams, 30
  - crashInfo, 31
  - crashStatus, 31
- crashInformation, 31
  - crashData, 32
  - crashId, 32
  - crashString, 32
  - crashStrlen, 32
  - gcdumpString, 32
  - gcdumpStrlen, 32
  - numCrashes, 32
- crashStatus
  - crashInfoParams, 31
- crashString
  - crashInformation, 32
- crashStrlen
  - crashInformation, 32
- csAttachState
  - NASServingSystemInfo, 187
  - nas\_servSystem, 153
- csBarStatus
  - nas\_CallBarringSysInfo, 89
  - nas\_callBarStatus, 90
- cur\_carr\_name
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- cur\_carr\_rev
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- curDataBearerTechnology
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t,  
458
- curProfile
  - pack\_wds\_SLQSMModifyProfile\_t, 268
  - UnPackGetProfileSettingOut, 469
- CurrChanRxRate
  - dunchannelRate, 37
- CurrChanTxRate
  - dunchannelRate, 37
- currDBTechAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t,  
465
- currNWInfo
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t,  
465
- currNetworkInfo, 32
  - NetworkType, 32
  - RATMask, 32
  - SOMask, 32
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_↔  
t, 456
- current\_channel\_rx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_↔  
t, 457
- current\_channel\_tx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_↔  
t, 457
- currentChannelRXRate
  - unpack\_wds\_GetConnectionRate\_t, 445
- currentChannelTXRate
  - unpack\_wds\_GetConnectionRate\_t, 445
- currentDataBearer
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 271
- currentNetwork
  - qmiWSDDataBearerTechnology, 279
- CurrentPLMN
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- cust\_attr
  - DMScustSettingInfo, 34
- cust\_id
  - DMScustSettingInfo, 34
  - DMSgetCustomInput, 36
  - pack\_dms\_GetCustFeaturesV2\_t, 189
  - pack\_dms\_SetCustFeaturesV2\_t, 192
- cust\_value
  - DMScustSettingInfo, 34
  - pack\_dms\_SetCustFeaturesV2\_t, 192
- custSetting
  - DMScustSettingList, 35
- CustomSCP
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- dBTechAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t,  
465
- dBTechnology
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t,  
465
- DEAULT\_LOC\_TIMEOUT\_IN\_SEC
  - common.h, 491
- DHCPRelayEnabled
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- DHCPv4LeaseOptTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, 443
- DHCPv4LeaseStateTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, 443
- DMS\_IMGDETAILS\_LEN

- dms.h, [500](#)
- DMS\_MAX\_CUST\_ID\_LEN
  - dms.h, [500](#)
- DMS\_MAX\_CUST\_VALUE\_LEN
  - dms.h, [500](#)
- DMS\_MAX\_FWUPDATE\_LOG\_STR\_SZ
  - dms.h, [500](#)
- DMS\_MAX\_FWUPDATE\_REF\_STR\_SZ
  - dms.h, [501](#)
- DMS\_PM\_FACTORY
  - dms.h, [501](#)
- DMS\_PM\_LOW
  - dms.h, [501](#)
- DMS\_PM\_OFFLINE
  - dms.h, [501](#)
- DMS\_PM\_ONLINE
  - dms.h, [501](#)
- DMS\_PM\_PERSISTENT\_LOW
  - dms.h, [501](#)
- DMS\_PM\_RESET
  - dms.h, [501](#)
- DMS\_PM\_SHUT\_DOWN
  - dms.h, [501](#)
- DMS\_SET\_REPORT\_DISABLE
  - dms.h, [501](#)
- DMS\_SET\_REPORT\_ENABLE
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_APPVERSION\_SZ
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_BOOTVERSION\_SZ
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_CARRIER\_SZ
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_CUR\_CARR\_NAME
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_CUR\_CARR\_REV
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_MODELID\_SZ
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_PACKAGEID\_SZ
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_PRIVERSION\_SZ
  - dms.h, [501](#)
- DMS\_SLQSFWINFO\_SKU\_SZ
  - dms.h, [501](#)
- DMS\_SWI\_SET\_IND\_DISABLE
  - dms.h, [501](#)
- DMS\_SWI\_SET\_IND\_ENABLE
  - dms.h, [501](#)
- DMS\_UINT8\_MAX\_STRING\_SZ
  - dms.h, [501](#)
- DMScustSettingInfo, [34](#)
  - cust\_attr, [34](#)
  - cust\_id, [34](#)
  - cust\_value, [34](#)
  - id\_length, [35](#)
  - value\_length, [35](#)
- DMScustSettingList, [35](#)
  - custSetting, [35](#)
  - list\_type, [35](#)
  - num\_instances, [35](#)
- DMSgetCustomFeatureV2, [35](#)
  - pCustSettingInfo, [36](#)
  - pCustSettingList, [36](#)
  - pGetCustomInput, [36](#)
- DMSgetCustomInput, [36](#)
  - cust\_id, [36](#)
  - list\_type, [36](#)
- DTMInd
  - unpack\_nas\_SLQSGetservingSystem\_t, [378](#)
- data
  - sMSCAddress, [285](#)
  - sMSEtwSMessage, [286](#)
  - sMSTransferRouteMTMessage, [289](#)
- data\_buf
  - NASOTAMessageTlv, [179](#)
- data\_len
  - NASOTAMessageTlv, [179](#)
- dataBearer
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [271](#)
- dataBearerMask
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, [458](#)
- dataBearerTech
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [459](#)
- dataCapabilities
  - nas\_dataSrvCapabilities, [102](#)
- dataCapabilitiesLen
  - nas\_dataSrvCapabilities, [102](#)
- DataCaps
  - unpack\_nas\_GetServingNetwork\_t, [368](#)
  - unpack\_nas\_GetServingNetworkCapabilities\_←  
t, [369](#)
- dataCaps
  - unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, [371](#)
- DataCapsLen
  - unpack\_nas\_GetServingNetwork\_t, [368](#)
  - unpack\_nas\_GetServingNetworkCapabilities\_←  
t, [369](#)
- dataCapsSize
  - unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, [371](#)
- DataRate
  - unpack\_qos\_swiQoSFlow\_t, [420](#)
- dataRateMax
  - unpack\_qos\_dataRate\_t, [402](#)
- dataServiceCaCapability
  - unpack\_dms\_GetDeviceCapabilities\_t, [312](#)
- DataServiceCapability
  - unpack\_dms\_GetDeviceCap\_t, [311](#)
- DataSrvCapabilities
  - unpack\_nas\_SLQSGetservingSystem\_t, [378](#)
- dataSysStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [465](#)



- dataSystemStatus
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 271
- Date
  - unpack\_swioma\_SLQSOMADMGetSessionInfo←\_t, 433
- DateLength
  - unpack\_swioma\_SLQSOMADMGetSessionInfo←\_t, 433
- day
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- dayLtSavingAdj
  - nas\_timeInfo, 164
- dayOfWeek
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- daylightSavings
  - nas\_qaQmi3Gpp2TimeZone, 144
- defaultPDNEnabled
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 456
- DefaultRoamInd
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- delayClass
  - LibPackGPRSRequestedQoS, 44
  - wds\_GPRSQoS, 477
- deliveryErrSDU
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 484
- description
  - unpack\_omaDmFotaTlv\_t, 400
- descriptionlength
  - unpack\_omaDmFotaTlv\_t, 400
- Description
  - nas\_QmiNas3GppNetworkInfo, 144
- destPortRangeEnd
  - LibPackTFTIDParams, 70
- destPortRangeStart
  - LibPackTFTIDParams, 70
- DetailedSvcInfo
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- DevCrashState
  - unpack\_dms\_GetCrashAction\_t, 309
- DisableIMSI
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- dl\_bw\_value
  - NASPhyCaAggPcellInfo, 180
  - NASPhyCaAggScellIDIBw, 181
  - NASPhyCaAggScellInfo, 183
  - nas\_PhyCaAggPcellInfo, 138
  - nas\_PhyCaAggScellIDIBw, 139
  - nas\_PhyCaAggScellInfo, 143
- dms.h, 494
  - ACT\_CODE\_MAX\_SIZE, 500
  - CK\_MAX\_SIZE, 500
  - DMS\_IMGDETAILS\_LEN, 500
  - DMS\_MAX\_CUST\_ID\_LEN, 500
  - DMS\_MAX\_CUST\_VALUE\_LEN, 500
  - DMS\_MAX\_FWUPDATE\_LOG\_STR\_SZ, 500
  - DMS\_MAX\_FWUPDATE\_REF\_STR\_SZ, 501
  - DMS\_PM\_FACTORY, 501
  - DMS\_PM\_LOW, 501
  - DMS\_PM\_OFFLINE, 501
  - DMS\_PM\_ONLINE, 501
  - DMS\_PM\_PERSISTENT\_LOW, 501
  - DMS\_PM\_RESET, 501
  - DMS\_PM\_SHUT\_DOWN, 501
  - DMS\_SET\_REPORT\_DISABLE, 501
  - DMS\_SET\_REPORT\_ENABLE, 501
  - DMS\_SLQSFWINFO\_APPVERSION\_SZ, 501
  - DMS\_SLQSFWINFO\_BOOTVERSION\_SZ, 501
  - DMS\_SLQSFWINFO\_CARRIER\_SZ, 501
  - DMS\_SLQSFWINFO\_CUR\_CARR\_NAME, 501
  - DMS\_SLQSFWINFO\_CUR\_CARR\_REV, 501
  - DMS\_SLQSFWINFO\_MODELID\_SZ, 501
  - DMS\_SLQSFWINFO\_PACKAGEID\_SZ, 501
  - DMS\_SLQSFWINFO\_PRIVERSION\_SZ, 501
  - DMS\_SLQSFWINFO\_SKU\_SZ, 501
  - DMS\_SWI\_SET\_IND\_DISABLE, 501
  - DMS\_SWI\_SET\_IND\_ENABLE, 501
  - DMS\_UINT8\_MAX\_STRING\_SZ, 501
  - ERI\_DATA\_MAX\_SIZE, 501
  - MAX\_BUILD\_ID\_LEN, 502
  - MEID\_MAX\_SIZE, 502
  - pack\_dms\_ActivateAutomatic, 502
  - pack\_dms\_GetActivationState, 502
  - pack\_dms\_GetBandCapability, 503
  - pack\_dms\_GetCrashAction, 503
  - pack\_dms\_GetCustFeature, 503
  - pack\_dms\_GetCustFeaturesV2, 504
  - pack\_dms\_GetDeviceCap, 504
  - pack\_dms\_GetDeviceCapabilities, 504
  - pack\_dms\_GetDeviceHardwareRev, 505
  - pack\_dms\_GetDeviceMfr, 505
  - pack\_dms\_GetDeviceSerialNumbers, 506
  - pack\_dms\_GetFSN, 507
  - pack\_dms\_GetFirmwareInfo, 506
  - pack\_dms\_GetFirmwareRevision, 506
  - pack\_dms\_GetFirmwareRevisions, 507
  - pack\_dms\_GetHardwareRevision, 508
  - pack\_dms\_GetIMSI, 508
  - pack\_dms\_GetManufacturer, 508
  - pack\_dms\_GetModelID, 509
  - pack\_dms\_GetNetworkTime, 509
  - pack\_dms\_GetOfflineReason, 510
  - pack\_dms\_GetPRLVersion, 510
  - pack\_dms\_GetPower, 510
  - pack\_dms\_GetSerialNumbers, 511
  - pack\_dms\_GetUSBComp, 511
  - pack\_dms\_GetVoiceNumber, 512
  - pack\_dms\_ResetToFactoryDefaults, 512
  - pack\_dms\_SLQSDmsSwiGetResetInfo, 516
  - pack\_dms\_SLQSDmsSwiIndicationRegister, 516
  - pack\_dms\_SLQSGetBandCapability, 517
  - pack\_dms\_SLQSGetERIFile, 517

- pack\_dms\_SLQSSwiClearDyingGaspStatistics, 518
- pack\_dms\_SLQSSwiGetCrashInfo, 518
- pack\_dms\_SLQSSwiGetDyingGaspCfg, 518
- pack\_dms\_SLQSSwiGetDyingGaspStatistics, 519
- pack\_dms\_SLQSSwiGetFirmwareCurr, 519
- pack\_dms\_SLQSSwiGetFwUpdateStatus, 520
- pack\_dms\_SLQSSwiGetHostDevInfo, 520
- pack\_dms\_SLQSSwiGetOSInfo, 520
- pack\_dms\_SLQSSwiGetSerialNoExt, 521
- pack\_dms\_SLQSSwiSetDyingGaspCfg, 521
- pack\_dms\_SLQSSwiSetHostDevInfo, 522
- pack\_dms\_SLQSSwiSetOSInfo, 522
- pack\_dms\_SLQSUIMGetState, 522
- pack\_dms\_SetActivationStatusCallback, 512
- pack\_dms\_SetCrashAction, 513
- pack\_dms\_SetCustFeature, 513
- pack\_dms\_SetCustFeaturesV2, 514
- pack\_dms\_SetEventReport, 514
- pack\_dms\_SetFirmwarePreference, 515
- pack\_dms\_SetPower, 515
- pack\_dms\_SetUSBComp, 516
- pack\_dms\_UIMChangePIN, 523
- pack\_dms\_UIMGetControlKeyStatus, 523
- pack\_dms\_UIMGetICCID, 524
- pack\_dms\_UIMGetPINStatus, 524
- pack\_dms\_UIMSetControlKeyProtection, 524
- pack\_dms\_UIMSetPINProtection, 525
- pack\_dms\_UIMUnblockControlKey, 525
- pack\_dms\_UIMUnblockPIN, 526
- pack\_dms\_UIMVerifyPIN, 526
- pack\_dms\_ValidateSPC, 526
- SLQS\_MAX\_DYING\_GASP\_CFG\_SMS\_CONT↔  
ENT\_LENGTH, 502
- SLQS\_MAX\_DYING\_GASP\_CFG\_SMS\_NUMB↔  
ER\_LENGTH, 502
- SPC\_SIZE, 502
- UNIQUE\_ID\_LEN, 502
- unpack\_dms\_ActivateAutomatic, 527
- unpack\_dms\_GetActivationState, 527
- unpack\_dms\_GetBandCapability, 528
- unpack\_dms\_GetCrashAction, 528
- unpack\_dms\_GetCustFeature, 528
- unpack\_dms\_GetCustFeaturesV2, 529
- unpack\_dms\_GetDeviceCap, 529
- unpack\_dms\_GetDeviceCapabilities, 529
- unpack\_dms\_GetDeviceHardwareRev, 530
- unpack\_dms\_GetDeviceMfr, 530
- unpack\_dms\_GetDeviceSerialNumbers, 531
- unpack\_dms\_GetFSN, 532
- unpack\_dms\_GetFirmwareInfo, 531
- unpack\_dms\_GetFirmwareRevision, 531
- unpack\_dms\_GetFirmwareRevisions, 532
- unpack\_dms\_GetHardwareRevision, 533
- unpack\_dms\_GetIMSI, 533
- unpack\_dms\_GetManufacturer, 533
- unpack\_dms\_GetModelID, 534
- unpack\_dms\_GetNetworkTime, 534
- unpack\_dms\_GetOfflineReason, 535
- unpack\_dms\_GetPRLVersion, 535
- unpack\_dms\_GetPower, 535
- unpack\_dms\_GetSerialNumbers, 536
- unpack\_dms\_GetUSBComp, 536
- unpack\_dms\_GetVoiceNumber, 537
- unpack\_dms\_ResetToFactoryDefaults, 537
- unpack\_dms\_SLQSDmsSwiGetResetInfo, 541
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind, 541
- unpack\_dms\_SLQSDmsSwiIndicationRegister, 542
- unpack\_dms\_SLQSGetBandCapability, 542
- unpack\_dms\_SLQSGetERIFile, 543
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics, 543
- unpack\_dms\_SLQSSwiGetCrashInfo, 544
- unpack\_dms\_SLQSSwiGetDyingGaspCfg, 544
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics, 544
- unpack\_dms\_SLQSSwiGetFirmwareCurr, 545
- unpack\_dms\_SLQSSwiGetFwUpdateStatus, 545
- unpack\_dms\_SLQSSwiGetHostDevInfo, 546
- unpack\_dms\_SLQSSwiGetOSInfo, 546
- unpack\_dms\_SLQSSwiGetSerialNoExt, 546
- unpack\_dms\_SLQSSwiSetDyingGaspCfg, 547
- unpack\_dms\_SLQSSwiSetHostDevInfo, 547
- unpack\_dms\_SLQSSwiSetOSInfo, 548
- unpack\_dms\_SLQSUIMGetState, 548
- unpack\_dms\_SetActivationStatusCallback, 537
- unpack\_dms\_SetCrashAction, 538
- unpack\_dms\_SetCustFeature, 538
- unpack\_dms\_SetCustFeaturesV2, 539
- unpack\_dms\_SetEventReport, 539
- unpack\_dms\_SetEventReport\_ind, 539
- unpack\_dms\_SetFirmwarePreference, 540
- unpack\_dms\_SetPower, 540
- unpack\_dms\_SetUSBComp, 541
- unpack\_dms\_UIMChangePIN, 548
- unpack\_dms\_UIMGetControlKeyStatus, 549
- unpack\_dms\_UIMGetICCID, 549
- unpack\_dms\_UIMGetPINStatus, 550
- unpack\_dms\_UIMSetControlKeyProtection, 550
- unpack\_dms\_UIMSetPINProtection, 550
- unpack\_dms\_UIMUnblockControlKey, 551
- unpack\_dms\_UIMUnblockPIN, 551
- unpack\_dms\_UIMVerifyPIN, 552
- unpack\_dms\_ValidateSPC, 552
- dms\_ActivationStatusTlv, 32
  - activationStatus, 33
  - TlvPresent, 33
- dms\_OperatingModeTlv, 33
  - operatingMode, 34
  - TlvPresent, 34
- domain
  - wds\_DomainNameList, 477
- domainLen
  - wds\_Domain, 476

DomainList  
    unpack\_wds\_SLQSSetRuntimeSettings\_t, 461  
domainName  
    wds\_Domain, 476  
dormancyStatAvail  
    unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465  
dormancyState  
    unpack\_wds\_GetDormancyState\_t, 448  
dormancyStatus  
    pack\_wds\_SLQSSetWdsEventCallback\_t, 271  
    unpack\_wds\_SLQSSetDUNCallInfo\_t, 459  
    unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465  
dtmSupp  
    nas\_GSMSSysInfo, 113  
dtmSuppValid  
    nas\_GSMSSysInfo, 113  
dunchannelRate, 36  
    CurrChanRxRate, 37  
    CurrChanTxRate, 37  
    MaxChanRxRate, 37  
    MaxChanTxRate, 37  
Duration  
    pack\_nas\_SetNetworkPreference\_t, 220  
    unpack\_nas\_GetNetworkPreference\_t, 367  
eCAT  
    common.h, 492  
eCTL  
    common.h, 492  
eDMS  
    common.h, 492  
eFW\_TYPE\_CWE\_NVU  
    libsdp.h, 558  
eFW\_TYPE\_CWE  
    libsdp.h, 558  
eFW\_TYPE\_INVALID  
    libsdp.h, 558  
eFW\_TYPE\_MBN\_GOBI  
    libsdp.h, 558  
eFW\_TYPE\_MBN  
    libsdp.h, 558  
eFW\_TYPE\_NVU  
    libsdp.h, 558  
eFW\_TYPE\_SPK  
    libsdp.h, 558  
eIND  
    common.h, 492  
eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED  
    nas.h, 586  
eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED\_LITE  
    nas.h, 587  
eLOG\_DEBUG  
    common.h, 491  
eLOG\_FATAL  
    common.h, 491  
eLOG\_INFO  
    common.h, 491  
eLOG\_LEVEL  
    common.h, 491  
eLOG\_WARN  
    common.h, 491  
eLOC  
    common.h, 492  
EMTlv  
    NASQmiCbkNasSystemSelPrefInd, 185  
eModel\_9X15  
    libsdp.h, 559  
eModel\_9X30  
    libsdp.h, 559  
eModel\_Unknown  
    libsdp.h, 559  
eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100  
    nas.h, 587  
eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15  
    nas.h, 587  
eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25  
    nas.h, 587  
eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50  
    nas.h, 587  
eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6  
    nas.h, 587  
eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75  
    nas.h, 587  
eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED\_LITE  
    nas.h, 587  
eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED\_LITE  
    nas.h, 587  
eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED\_LITE  
    nas.h, 587  
eNAS  
    common.h, 492  
eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT  
    qmerrno.h, 653  
eQCWWAN\_ERR\_BUFFER\_SZ  
    qmerrno.h, 652

eQCWWAN\_ERR\_CANCEL\_OP  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_DRIVER  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_ENUM\_BEGIN  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_ENUM\_END  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_FILE\_COPY  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_FILE\_OPEN  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_GENERAL  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_INTERNAL  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_INVALID\_ARG  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_INVALID\_DEVID  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_INVALID\_FILE  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_INVALID\_QMI\_RSP  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_INVALID\_XID  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_MEMORY  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_MULTIPLE\_DEVICES  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPORTED  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_NO\_CANCELABLE\_OP  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_NO\_CONNECTION  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_NO\_DEVICE  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_NO\_SIGNAL  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_NONE  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_NULL\_TLV  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_OFFLINE  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_PDU\_GENERATION  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_ABORTED  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_CALL\_FAILED  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_CAT\_END  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_QMI\_CAT\_START  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_QMI\_CAUSE\_CODE  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_CONNECT  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_DISABLED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_ENCODING  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_GENERAL  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_IFACE  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE  
     qmerrno.h, [655](#)

eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCE  
     ES  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INTERNAL  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_ARG  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD  
     qmerrno.h, 656  
 eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_INDEX  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_ID  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_PINID  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_AC-  
     TION  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTI-  
     ON  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP  
     qmerrno.h, 656  
 eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS-  
     \_IN\_USE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN-  
     \_USE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_MAX  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAI-  
     LURE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_MISSING\_ARG  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_NO\_EFFECT  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_NO\_ENTRY  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_NO\_MEMORY  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_NO\_RADIO  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE  
     qmerrno.h, 654  
 eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED  
     qmerrno.h, 655  
 eQCWWAN\_ERR\_QMI\_OFFSET  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNUPPORT-  
     ED  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPO-  
     RTED  
     qmerrno.h, 653  
 eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE  
     qmerrno.h, 655



eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_REQ\_SCH  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_REQ\_TO  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSU↵  
     PPORTED  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_REQ  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_RSP\_TO  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_RSP  
     qmerrno.h, [652](#)  
 eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_SESSION\_INVALID  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_SMSC\_ADDR  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_TPDU\_TYPE  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSA↵  
     CTION  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_QMI\_UNKNOWN  
     qmerrno.h, [654](#)  
 eQCWWAN\_ERR\_QMI\_WIDTH  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_RESET  
     qmerrno.h, [653](#)  
 eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_END  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK↵  
     \_PROCESS  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION↵  
     \_ID  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION↵  
     \_ID  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPP↵  
     ORTED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPP↵  
     ORTED  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_S↵  
     SESSIONS  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_START  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_TIMEOUT  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP  
     qmerrno.h, [655](#)  
 eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOU↵  
     ND  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIDCS\_END  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIDCS\_START  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIIM\_END  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOW↵  
     NLOADED  
     qmerrno.h, [656](#)  
 eQCWWAN\_ERR\_SWIIM\_FW\_ENTER\_DOWNLOA↵  
     D\_MODE

- qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETE
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_INVALID\_SLOT\_IND↔EX
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MIS↔MATCH
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRE↔NT\_ACTIVE\_IMAGE
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWIIM\_START
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWISM\_END
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT↔FOUND
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRU↔PTED
  - qmerrno.h, [656](#)
- eQCWWAN\_ERR\_SWISMS\_START
  - qmerrno.h, [656](#)
- eQCWWANError
  - qmerrno.h, [652](#)
- eQMI\_LOC\_SESS\_STATUS\_FAILURE
  - loc.h, [569](#)
- eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS
  - loc.h, [569](#)
- eQMI\_LOC\_SESS\_STATUS\_SUCCESS
  - loc.h, [569](#)
- eQMI\_LOC\_SESS\_STATUS\_TIMEOUT
  - loc.h, [569](#)
- eQMI\_SVC
  - common.h, [491](#)
- eQOS
  - common.h, [492](#)
- eREQ
  - common.h, [492](#)
- ERI\_DATA\_MAX\_SIZE
  - dms.h, [501](#)
- eRSP
  - common.h, [492](#)
- eSDP\_FWDWL\_ERR\_END
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_FAIL
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_FW\_UPGRADE
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_FW\_VERSION\_FAIL
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_GENERAL
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_INVALID\_DEV
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_INVALID\_PATH
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_PATH\_NOT\_SPECIFIED
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_PATH\_TOO\_LONG
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_PRI\_FAIL
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_SDK
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_SET\_CBK
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_ERR\_TIMEOUT
  - libsdh.h, [559](#)
- eSDP\_FWDWL\_SUCCESS
  - libsdh.h, [559](#)
- eSMS
  - common.h, [492](#)
- ESNString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [313](#)
- eSWILOC
  - common.h, [492](#)
- eSWIOMA
  - common.h, [492](#)
- eTIMEOUT\_10\_S
  - common.h, [492](#)
- eTIMEOUT\_20\_S
  - common.h, [492](#)
- eTIMEOUT\_2\_S
  - common.h, [492](#)
- eTIMEOUT\_300\_S
  - common.h, [492](#)
- eTIMEOUT\_30\_S
  - common.h, [492](#)
- eTIMEOUT\_5\_S
  - common.h, [492](#)
- eTIMEOUT\_60\_S
  - common.h, [492](#)
- eTIMEOUT\_8\_S
  - common.h, [492](#)
- eTIMEOUT\_DEFAULT
  - common.h, [492](#)





- pack\_uim\_ChangePin\_t, 248
  - pack\_uim\_SetPinProtection\_t, 250
  - pack\_uim\_UnblockPin\_t, 253
- engineState
  - unpack\_loc\_EngineState\_Ind\_t, 353
- eqmiCbkSetStatus
  - sms.h, 670
- eriData
  - eriDataparams, 37
- eriDataLen
  - eriDataparams, 37
- eriDataparams, 37
  - eriData, 37
  - eriDataLen, 37
- eriFile
  - unpack\_dms\_SLQSGetERIFile\_t, 329
- error
  - unpack\_wds\_GetLastMobileIPError\_t, 448
- errorRate
  - nas\_errorRateListElement, 105
- errorRateInfo
  - nas\_SLQSSignalStrengthsInformation, 156
- errorRateList
  - unpack\_nas\_SLQSGetSignalStrength\_t, 380
- errorRateListLen
  - unpack\_nas\_SLQSGetSignalStrength\_t, 380
- errorState
  - slotInf, 284
  - uim\_slotInfo, 305
- esn
  - unpack\_dms\_GetSerialNumbers\_t, 320
- esnSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- EspSpi
  - unpack\_qos\_swiQosFilter\_t, 415
- EtwsMessageInfo
  - sMSEtwsMessageTlv, 286
- event
  - unpack\_qos\_SLQSSetQosPriEventCallback\_ind↔\_t, 412
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 413
- eventMask
  - pack\_uim\_SLQSUIEventRegister\_t, 251
  - unpack\_uim\_SLQSUIEventRegister\_t, 440
- eventRegister
  - pack\_loc\_EventRegister\_t, 206
- eventType
  - unpack\_swioma\_SLQSOMADMAAlertCallback↔\_ind\_t, 431
- executingImage
  - FMSImageIDEntries, 40
- exponent
  - unpack\_qos\_pktErrRate\_t, 404
- ExtErrorCode
  - PackCreateProfileOut, 277
- extPowerState
  - pack\_loc\_SetExtPowerState\_t, 206
- extendedErrorCode
  - unpack\_wds\_SLQSDeleteProfile\_t, 454
- FIRMWARE\_INFO\_STRING\_SIZE
  - libsdh.h, 557
- FLOAT
  - SwiDataTypes.h, 677
- FMS\_FW\_PRI\_BUILD\_MATCH\_LEN
  - fms.h, 553
- FMS\_GOBI\_LISTENTRIES\_MAX
  - fms.h, 553
- FMS\_GOBI\_MBN\_BUILD\_ID\_STR\_LEN
  - fms.h, 553
- FMS\_GOBI\_MBN\_IMG\_ID\_STR\_LEN
  - fms.h, 553
- FMS\_IMAGE\_ID\_BUILD\_ID\_LEN
  - fms.h, 554
- FMS\_IMAGE\_ID\_IMG\_ID\_LEN
  - fms.h, 554
- FMS\_IMAGE\_ID\_MAX\_ENTRIES
  - fms.h, 554
- FMS\_IMAGE\_ID\_PRI\_IMGTYPE
  - fms.h, 554
- FMS\_MAX\_IMAGE\_ID\_ELEMENT
  - fms.h, 554
- FMS\_MAX\_IMAGE\_PREFERENCE\_IMAGE\_SIZE
  - fms.h, 554
- FMSImageElement, 38
  - buildId, 38
  - buildIdLength, 38
  - imageId, 39
  - imageType, 39
- FMSImageIDEntries, 40
  - executingImage, 40
  - imageIDElement, 40
  - imageIDSize, 40
  - imageType, 40
  - maxImages, 41
- FMSImageIDElement, 39
  - buildIDLength, 39
  - buildID, 39
  - failureCount, 39
  - imageID, 39
  - storageIndex, 40
- FMSImageList, 41
  - imageIDEntries, 41
  - listSize, 41
- FMSPrefImageList, 41
  - listEntries, 42
  - listSize, 42
- FOTAUpdate
  - pack\_swioma\_SLQSOMADMSetSettings\_t, 247
  - unpack\_swioma\_SLQSOMADMGetSettings\_t, 435
- FOTAdownload
  - pack\_swioma\_SLQSOMADMSetSettings\_t, 246
  - unpack\_swioma\_SLQSOMADMGetSettings\_t, 435
- facility
  - pack\_dms\_UIMGetControlKeyStatus\_t, 197
  - pack\_dms\_UIMSetControlKeyProtection\_t, 198

- pack\_dms\_UIMUnblockControlKey\_t, 200
- facilityCk
  - pack\_dms\_UIMSetControlKeyProtection\_t, 198
  - pack\_dms\_UIMUnblockControlKey\_t, 200
- facilityState
  - pack\_dms\_UIMSetControlKeyProtection\_t, 198
  - unpack\_dms\_UIMGetControlKeyStatus\_t, 339
- failureCount
  - FMSImageIdElement, 39
- family
  - pack\_wds\_GetDefaultProfileNum\_t, 256
  - pack\_wds\_SetDefaultProfileNum\_t, 260
- fileID
  - uim\_fileInfo, 299
- fileIndex
  - pack\_uim\_ReadTransparent\_t, 249
- fill\_pack\_ctx
  - common.h, 493
- fill\_sdu\_hdr
  - common.h, 493
- filterId
  - LibPackTFTIDParams, 70
- fix\_rate
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 244
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 429
- fix\_rate\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 429
- fix\_type
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 244
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 429
- fix\_type\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- flags
  - sensorData\_t, 281
- flowLabel
  - LibPackTFTIDParams, 70
- fms.h, 552
  - FMS\_FW\_PRI\_BUILD\_MATCH\_LEN, 553
  - FMS\_GOBI\_LISTENTRIES\_MAX, 553
  - FMS\_GOBI\_MBN\_BUILD\_ID\_STR\_LEN, 553
  - FMS\_GOBI\_MBN\_IMG\_ID\_STR\_LEN, 553
  - FMS\_IMAGE\_ID\_BUILD\_ID\_LEN, 554
  - FMS\_IMAGE\_ID\_IMG\_ID\_LEN, 554
  - FMS\_IMAGE\_ID\_MAX\_ENTRIES, 554
  - FMS\_IMAGE\_ID\_PRI\_IMGTYPE, 554
  - FMS\_MAX\_IMAGE\_ID\_ELEMENT, 554
  - FMS\_MAX\_IMAGE\_PREFERENCE\_IMAGE\_SIZE, 554
  - GetValidFwPriCombinations, 554
  - pack\_fms\_GetImagesPreference, 554
  - pack\_fms\_GetStoredImages, 555
  - pack\_fms\_SetImagesPreference, 555
  - unpack\_fms\_GetImagesPreference, 555
  - unpack\_fms\_GetStoredImages, 555
  - unpack\_fms\_SetImagesPreference, 556
- Forbidden
  - nas\_QmiNas3GppNetworkInfo, 144
- ForceRev0
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- format
  - sMSTransferRouteMTMessage, 289
- fqdnAddr
  - wds\_PCSCFFQDNAddress, 479
- fqdnLen
  - wds\_PCSCFFQDNAddress, 480
- freq
  - NASPhyCaAggPcellInfo, 180
  - NASPhyCaAggScellIndType, 182
  - NASPhyCaAggScellInfo, 183
  - nas\_PhyCaAggPcellInfo, 138
  - nas\_PhyCaAggScellIndType, 140
  - nas\_PhyCaAggScellInfo, 143
- freqsLen
  - nas\_LTEInfoInterfreq, 123
  - nas\_LTEInfoNeighboringGSM, 126
  - nas\_LTEInfoNeighboringWCDMA, 127
- function
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 244
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- function\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- FwAutoCheck
  - unpack\_swima\_SLQSOMADMGetSettings\_t, 435
- FwAvailability
  - unpack\_swima\_SLQSOMADMStartSession\_t, 436
- fwloadsize
  - unpack\_omaDmFotaTlv\_t, 400
- fwloadComplete
  - unpack\_omaDmFotaTlv\_t, 400
- fwvers
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 332
- gDIBitRate
  - LibPackQosClassID, 68
- GPRSGrantedQoS
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- GPSPM
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- GPSSel
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- GSMRSSIThreshListLen
  - nas\_GSMRSSIThresh, 109
- GSMSSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 388
- gUIBitRate
  - LibPackQosClassID, 68
- GWAOPTiv
  - NASQmiCbkNasSystemSelPrefInd, 185
- GWAcqOrderPref
  - NASGWAcqOrderPrefTlv, 177
- GWAddressV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- gcdumpString
  - crashInformation, 32

- gcdumpStrlen
  - crashInformation, 32
- geoSysIdx
  - nas\_AddCDMASysInfo, 88
  - nas\_AddSysInfo, 88
- geranArfcn
  - nas\_geranInstInfo, 108
- geranBsicBcc
  - nas\_geranInstInfo, 108
- geranBsicNcc
  - nas\_geranInstInfo, 108
- geranInst
  - nas\_UMTSInfo, 166
- GeranInstInfo
  - nas\_UMTSInfo, 166
- geranRssi
  - nas\_geranInstInfo, 108
- get\_version
  - common.h, 493
- GetCustomFeatureV2
  - unpack\_dms\_GetCustFeaturesV2\_t, 310
- GetValidFwPriCombinations
  - fms.h, 554
- globalCellId
  - nas\_LTEInfoIntrafreq, 125
- glog
  - common.h, 494
- gloglvl
  - common.h, 494
- gnssSvId
  - loc\_satelliteInfo, 82
- gnssSvUsedList
  - loc\_svUsedforFix, 85
- gnssSvUsedList\_len
  - loc\_svUsedforFix, 85
- Gpp2TimeZone
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- GppNetworkDSTAdjustment
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- GppTimeZone
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- GpsEnable
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- gpsTimeOfWeekMs
  - loc\_gpsTime, 78
- gpsWeek
  - loc\_gpsTime, 78
- grntDownlinkBitrate
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 484
- grntUplinkBitrate
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- GsmCellInfo
  - nas\_lteGsmCellInfo, 120
- gsmUmtsDI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- gsmUmtsUI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- guaranteedRate
  - unpack\_qos\_dataRate\_t, 402
- gwAddressV6
  - wds\_IPV6GWAddressInfo, 479
- gwV6PrefixLen
  - wds\_IPV6GWAddressInfo, 479
- gyroData
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 214
- gyroTemp
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- gyroTimeSrc
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- HASPI
  - unpack\_wds\_GetMobileIPProfile\_t, 449
- HASState
  - unpack\_wds\_GetMobileIPProfile\_t, 449
- HDOP
  - loc\_precisionDilution, 80
- HDRECIOTreshListLen
  - nas\_HDRECIOTresh, 114
- HDRIOTreshListLen
  - nas\_HDRIOTresh, 114
- HDRRSSITreshListLen
  - nas\_HDRRSSITresh, 115
- HDRSINRTreshListLen
  - nas\_HDRSINRThreshold, 116
- HDRSSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 388
- HardwareControlledMode
  - unpack\_dms\_GetPower\_t, 319
- has\_accelTemp
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_acceleroTimeSrc
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_accleroData
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_altitudeSrcInfo
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_altitudeWrtEllipsoid
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_altitudeWrtMeanSeaLevel
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_confidence
  - pack\_loc\_SLQSLOCSetCradleMountConfig\_t, 216
- has\_gyroData
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_gyroTemp
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_gyroTimeSrc
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_horConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_horReliability
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_horUncCircular
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212

- has\_latitude
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_longitude
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_opaqueuld
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- has\_positionSrc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_rawHorConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_rawHorUncCircular
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_timestampAge
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_timestampUtc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_vertConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_vertRelicability
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- has\_vertUnc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- hdrActiveProt
  - nas\_HDRSysInfo, 118
- hdrActiveProtValid
  - nas\_HDRSysInfo, 118
- hdrHybrid
  - nas\_detailSvcInfo, 103
- HdrPersonality
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- hdrPersonality
  - NASServingSystemInfo, 187
  - nas\_HDRSysInfo, 118
- hdrPersonalityValid
  - nas\_HDRSysInfo, 118
- hdrSSInfo, 42
  - ecio, 42
  - io, 42
  - rsi, 42
  - sinr, 42
- hdrSrvStatus
  - nas\_detailSvcInfo, 103
- healthStatus
  - loc\_satelliteInfo, 82
- helper\_get\_resp\_ctx
  - common.h, 493
- helper\_get\_xid
  - common.h, 493
- helper\_set\_log\_func
  - common.h, 493
- helper\_set\_log\_lvl
  - common.h, 493
- horConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- horReliability
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- horUncCircular
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- hotSwap
  - uim\_hotSwapStatus, 300
- hotSwapLength
  - uim\_hotSwapStatus, 300
- hour
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- hsCallStatus
  - nas\_WCDMASysInfo, 175
- hsCallStatusValid
  - nas\_WCDMASysInfo, 175
- hsInd
  - nas\_WCDMASysInfo, 175
- hsIndValid
  - nas\_WCDMASysInfo, 175
- hwType
  - wds\_DHCPv4HWConfig, 474
  - wdsDhcpv4HwConfig, 486
- hwVer
  - unpack\_dms\_GetHardwareRevision\_t, 316
- iLTEbandValue
  - NASPhyCaAggPcellInfo, 180
  - NASPhyCaAggScellInfo, 183
  - nas\_PhyCaAggPcellInfo, 138
  - nas\_PhyCaAggScellInfo, 143
- IMCNflag
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- IMEIString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- IMG\_MASK\_CLEAR
  - libsdh.h, 557
- IMG\_MASK\_GENERIC
  - libsdh.h, 557
- IMG\_MASK\_MDM
  - libsdh.h, 557
- IMG\_MASK\_PRI
  - libsdh.h, 557
- IMSInfo
  - sMSOnIMSTlv, 289
- IMSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 426
- INT32
  - SwiDataTypes.h, 677
- INT8
  - SwiDataTypes.h, 677
- IPAddressV6
  - ipv6AddressInfo, 43
  - wds\_IPV6AddressInfo, 478
- IPFamSupport
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- IPFamilyPreference
  - pack\_wds\_SLQSSetIPFamilyPreference\_t, 270
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- IPSECspi
  - LibPackTFTIDParams, 70
- IPV6\_ADDRESS\_ARRAY\_SIZE
  - wds.h, 706

- IPv6AddrInfo
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- IPv6GWAddrInfo
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- IPv6PrefixLen
  - ipv6AddressInfo, 43
  - wds\_IPV6AddressInfo, 478
- IPv4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- IPv4Addr
  - wds\_IPv4AdTlv, 478
- IPv4AddrTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, 443
- IPv4DstAddr
  - unpack\_qos\_swiQosFilter\_t, 416
- IPv4SrcAddr
  - unpack\_qos\_swiQosFilter\_t, 416
- IPv4Tos
  - unpack\_qos\_swiQosFilter\_t, 416
- IPv6DstAddr
  - unpack\_qos\_swiQosFilter\_t, 416
- IPv6Label
  - unpack\_qos\_swiQosFilter\_t, 416
- IPv6SrcAddr
  - unpack\_qos\_swiQosFilter\_t, 416
- IPv6TrafCls
  - unpack\_qos\_swiQosFilter\_t, 416
- Id
  - unpack\_qos\_swiQosFilter\_t, 415
- id
  - loc\_BdsSV, 74
  - loc\_SV, 84
  - nas\_CSGID, 100
  - pack\_dms\_UIMChangePIN\_t, 196
  - pack\_dms\_UIMSetPINProtection\_t, 199
  - pack\_dms\_UIMUnblockPIN\_t, 200
  - pack\_dms\_UIMVerifyPIN\_t, 201
  - unpack\_qos\_QosFlowInfoState\_t, 407
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 413
- id\_length
  - DMScustSettingInfo, 35
- image\_info\_t, 42
  - buildIDLen, 43
  - buildID, 43
  - imageType, 43
  - uniqueID, 43
- imageDElement
  - FMSImageIDEntries, 40
- imageIDEntries
  - FMSImageList, 41
- imageIDSize
  - FMSImageIDEntries, 40
- imageID
  - FMSImageIDElement, 39
- imageId
  - FMSImageElement, 39
- imageList
  - unpack\_fms\_GetStoredImages\_t, 344
- ImageListSize
  - unpack\_fms\_GetImagesPreference\_t, 343
- imageListSize
  - pack\_fms\_SetImagesPreference\_t, 203
- imageType
  - FMSImageElement, 39
  - FMSImageIDEntries, 40
  - image\_info\_t, 43
- ImageTypes
  - unpack\_fms\_SetImagesPreference\_t, 345
- ImageTypesSize
  - unpack\_fms\_SetImagesPreference\_t, 345
- imagelistSize
  - unpack\_fms\_GetStoredImages\_t, 344
- imei\_no
  - unpack\_dms\_GetSerialNumbers\_t, 320
- imeiSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- imeiSvnSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- IimeiSvnString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- imeisv\_svn
  - unpack\_dms\_GetSerialNumbers\_t, 321
- imgType
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 334
- imsRegState
  - nas\_CommInfo, 99
- imsi
  - unpack\_dms\_GetIMSI\_t, 316
- imsi\_11\_12
  - nas\_CDMASysInfoExt, 97
- InUse
  - nas\_QmiNas3GppNetworkInfo, 144
- includes\_pcs\_digit
  - nas\_QmisNasPcsDigit, 146
- index
  - pack\_wds\_GetMobileIPProfile\_t, 257
  - pack\_wds\_SetDefaultProfileNum\_t, 260
  - pack\_wds\_SetMobileIPProfile\_t, 263
  - unpack\_qos\_swiQosFilter\_t, 416
  - unpack\_qos\_swiQosFlow\_t, 420
  - unpack\_wds\_GetDefaultProfileNum\_t, 447
- index1xPri
  - uim\_cardStatus, 297
- index1xSec
  - uim\_cardStatus, 297
- indexGwPri
  - uim\_cardStatus, 297
- indexGwSec
  - uim\_cardStatus, 297
- Info
  - unpack\_nas\_SLQSNasSwtOTAMessageCallback↔\_ind\_t, 391
  - unpack\_nas\_SLQSSetSysSelectionPrefCall↔\_Back\_ind\_t, 393
- InfoInterfreq

- nas\_LTEInfoInterfreq, 123
- insNmrCellInfo
  - nas\_GERANInfo, 107
- instancesSize
  - unpack\_nas\_GetRFInfo\_t, 367
- interval
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 271
- Io
  - unpack\_nas\_SLQSGetSignalStrength\_t, 380
- io
  - hdrSSInfo, 42
  - nas\_SLQSSignalStrengthsInformation, 156
- ioDelta
  - nas\_SLQSSignalStrengthsIndReq, 155
- ipAddress
  - pack\_wds\_SetDefaultProfile\_t, 259
- ipFamily
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback↔\_t, 463
- ipVersion
  - LibPackTFTIDParams, 70
- ipaddr
  - unpack\_wds\_GetDefaultProfile\_t, 446
- ipaddrv6
  - unpack\_wds\_GetDefaultProfile\_t, 447
- ipv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔\_t, 468
- ipv4GWAddress
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔\_t, 468
- ipv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔\_t, 468
- ipv6AddressInfo, 43
  - IPAddressV6, 43
  - IPv6PrefixLen, 43
- ipv6GWAddress
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔\_t, 468
- is856SysId
  - nas\_HDRSysInfo, 118
- is856SysIdValid
  - nas\_HDRSysInfo, 118
- is\_DataRate\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_EspSpi\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv4DstAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv4SrcAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv4Tos\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv6DstAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv6Label\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv6SrcAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_IPv6TrafCls\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_Id\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_Jitter\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_Latency\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_LteBandCapability\_Available
  - unpack\_dms\_SLQSGetBandCapability\_t, 329
- is\_LteQci\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_MaxAllowedPktSz\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_MinPolicedPktSz\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_NxtHdrProto\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_PktErrRate\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_Precedence\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_ProfileId3GPP2\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_RxQFlowGranted\_Available
  - unpack\_qos\_QosFlowInfo\_t, 406
- is\_TCPDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_TCPSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_TdsBandCapability\_Available
  - unpack\_dms\_SLQSGetBandCapability\_t, 329
- is-TokenBucket\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_TrafficClass\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_TranDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_TranSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 416
- is\_TxQFlowGranted\_Available
  - unpack\_qos\_QosFlowInfo\_t, 406
- is\_UDPDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 417
- is\_UDPSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 417
- is\_val\_3GPP2Pri\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_val\_3GPPImCn\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_val\_3GPPResResidualBER\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_val\_3GPPSigInd\_Available
  - unpack\_qos\_swiQosFlow\_t, 420
- is\_val\_3GPPTraHdlPri\_Available
  - unpack\_qos\_swiQosFlow\_t, 420



- isNewFlow
  - unpack\_qos\_QosFlowInfoState\_t, [407](#)
- isPrefDataPath
  - nas\_GSMSrvStatusInfo, [110](#)
  - nas\_SrvStatusInfo, [157](#)
- isRadioTuned
  - nas\_RxSigInfo, [150](#)
- isSysForbidden
  - nas\_detailSvcInfo, [103](#)
  - nas\_sysInfoCommon, [159](#)
- isSysForbiddenValid
  - nas\_sysInfoCommon, [159](#)
- isSysPriMatch
  - nas\_CDMA SysInfo, [96](#)
  - nas\_HDR SysInfo, [118](#)
- isSysPriMatchValid
  - nas\_CDMA SysInfo, [96](#)
  - nas\_HDR SysInfo, [118](#)
- IsVoiceEnabled
  - pack\_dms\_SetCustFeature\_t, [191](#)
  - unpack\_dms\_GetCustFeature\_t, [310](#)
- Jitter
  - unpack\_qos\_swiQosFlow\_t, [420](#)
- LBPTlv
  - NASQmiCbkNasSystemSelPrefInd, [185](#)
- LIBPACK\_MAX\_QOS\_FILTERS
  - qos.h, [659](#)
- LIBPACK\_MAX\_QOS\_FLOW\_PER\_APN\_STATS
  - qos.h, [659](#)
- LIBPACK\_MAX\_QOS\_FLOWS
  - qos.h, [659](#)
- LIBPACK\_MAX\_SWIOMA\_STR\_LEN
  - swioma.h, [680](#)
- LIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB
  - nas.h, [586](#)
- LIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE
  - nas.h, [586](#)
- LIBPACK\_QMI\_CBK\_PARAM\_NOCHANGE
  - sms.h, [670](#)
- LIBPACK\_QMI\_CBK\_PARAM\_RESET
  - sms.h, [670](#)
- LIBPACK\_QMI\_CBK\_PARAM\_SET
  - sms.h, [670](#)
- LIBSDP\_CARRIER\_PACKAGE\_SKU
  - libsdp.h, [557](#)
- LIBSDP\_SKU\_STRING\_LENGTH
  - libsdp.h, [557](#)
- LOC\_UINT8\_MAX\_STRING\_SZ
  - loc.h, [566](#)
- LOCEVENTMASKBATCHFULLNOTIFICATION
  - loc.h, [566](#)
- LOCEVENTMASKENGINESTATE
  - loc.h, [566](#)
- LOCEVENTMASKFIXSESSIONSTATE
  - loc.h, [566](#)
- LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION
  - loc.h, [566](#)
- LOCEVENTMASKGEOFENCEBREACHNOTIFICATION
  - loc.h, [566](#)
- LOCEVENTMASKGEOFENCEGENALERT
  - loc.h, [566](#)
- LOCEVENTMASKGNSSMEASUREMENTREPORT
  - loc.h, [566](#)
- LOCEVENTMASKGNSSSVINFO
  - loc.h, [566](#)
- LOCEVENTMASKINJECTPOSITIONREQ
  - loc.h, [566](#)
- LOCEVENTMASKINJECTPREDICTEDORBITSREQ
  - loc.h, [567](#)
- LOCEVENTMASKINJECTTIMERREQ
  - loc.h, [567](#)
- LOCEVENTMASKINJECTWIFIAPDATAREQ
  - loc.h, [567](#)
- LOCEVENTMASKINVALIDVALUE
  - loc.h, [567](#)
- LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT
  - loc.h, [567](#)
- LOCEVENTMASKLOCATIONSERVERCONNECTIO↔NREQ
  - loc.h, [567](#)
- LOCEVENTMASKMOTIONDATACONTROL
  - loc.h, [567](#)
- LOCEVENTMASKNIGEOFENCENOTIFICATION
  - loc.h, [567](#)
- LOCEVENTMASKNINOTIFYVERIFYREQ
  - loc.h, [567](#)
- LOCEVENTMASKNMEA
  - loc.h, [568](#)
- LOCEVENTMASKPEDOMETERCONTROL
  - loc.h, [568](#)
- LOCEVENTMASKPOSITIONREPORT
  - loc.h, [568](#)
- LOCEVENTMASKSENSORSTREAMINGREADYST↔ATUS
  - loc.h, [568](#)
- LOCEVENTMASKSETSPISTREAMINGREPORT
  - loc.h, [568](#)
- LOCEVENTMASKTIMESYNCREQ
  - loc.h, [568](#)
- LOCEVENTMASKVEHICLEDATAREADYSTATUS
  - loc.h, [568](#)
- LOCEVENTMASKWIFIREQ
  - loc.h, [568](#)
- LPCSTR
  - SwiDataTypes.h, [677](#)
- LTEAttachProfile
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [456](#)
- LTEAttachProfileList
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [456](#)
- LTEAttachProfileListLen
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [269](#)
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [456](#)
- LTEBandPref

- NASLTEBandPreferenceTlv, 178
- LTECphyCAInfo
  - unpack\_nas\_SlqsGetLTECphyCAInfo\_t, 375
- LTERSRPThreshListLen
  - nas\_LTERSRPThresh, 128
- LTERSRQThreshListLen
  - nas\_LTERSRQThresh, 129
- LTERRSIThreshListLen
  - nas\_LTERRSIThresh, 129
- LTESNRThreshListLen
  - nas\_LTESNRThreshold, 131
- LTESSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 388
- LTEWCDMACellInfo
  - nas\_LTEInfoNeighboringWCDMA, 127
- Lac
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- lac
  - nas\_GERANInfo, 107
  - nas\_GSMSysInfo, 113
  - nas\_LTESysInfo, 133
  - nas\_UMTSInfo, 166
  - nas\_WCDMASysInfo, 175
- lacValid
  - nas\_GSMSysInfo, 113
  - nas\_LTESysInfo, 133
  - nas\_WCDMASysInfo, 175
- lastCallDataBearerTech
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- lastCallDataBearerTechnology
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, 458
- lastCallRXOKBytesCnt
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- lastCallTXOKBytesCnt
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- Latency
  - unpack\_qos\_swiQosFlow\_t, 421
- latitude
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- leapSeconds
  - nas\_qaQmi3Gpp2TimeZone, 144
- leaseState
  - wds\_DHCPLeaseStateTlv, 472
- len
  - loc\_BdsSVInfo, 74
  - loc\_SVInfo, 85
  - unpack\_nas\_GetSignalStrengths\_t, 370
- length
  - sMSCAddress, 285
  - sMSEtwSMessage, 286
  - sMSTransferRouteMTMessage, 289
  - uim\_readTransparentInfo, 301
- LibPackGPRSRequestedQoS, 43
  - delayClass, 44
  - meanThroughputClass, 44
  - peakThroughputClass, 44
  - precedenceClass, 44
  - reliabilityClass, 44
- LibPackQosClassID, 67
  - gDIBitRate, 68
  - gUIBitRate, 68
  - maxDIBitRate, 68
  - maxUIBitRate, 68
  - QCI, 68
- LibPackTFTIDParams, 68
  - destPortRangeEnd, 70
  - destPortRangeStart, 70
  - eValid, 70
  - filterId, 70
  - flowLabel, 70
  - IPSECSPi, 70
  - ipVersion, 70
  - nextHeader, 70
  - pSourceIP, 70
  - sourceIPMask, 70
  - srcPortRangeEnd, 70
  - srcPortRangeStart, 70
  - tosMask, 70
- LibPackUMTSQoS, 70
  - deliveryErrSDU, 72
  - grntDownlinkBitrate, 72
  - grntUplinkBitrate, 72
  - maxDownlinkBitrate, 72
  - maxSDUSize, 72
  - maxUplinkBitrate, 72
  - qosDeliveryOrder, 72
  - resBerRatio, 72
  - sduErrorRatio, 72
  - trafficClass, 72
  - trafficPriority, 72
  - transferDelay, 72
- LibPackUMTSReqQoSsigInd, 72
  - SigInd, 73
  - UMTSReqQoS, 73
- LibPackprofile\_3GPP2, 62
  - pAPNClass3GPP2, 66
  - pAPNEnabled3GPP2, 66
  - pAllowLinger, 66
  - pApnString, 66
  - pApnStringSize, 66
  - pAppPriority, 66
  - pAppType, 66
  - pAuthPassword, 66
  - pAuthPassword\_tSize, 66
  - pAuthProtocol, 66
  - pAuthRetryCount, 66
  - pAuthTimeout, 66
  - pDataMode, 66
  - pDataRate, 66
  - plpcpAckTimeout, 66
  - plpcpCreqRetryCount, 66
  - plsPscsfAddressNedded, 67
  - pLcpAckTimeout, 67
  - pLcpCreqRetryCount, 67
  - pNegoDnsSrvrPref, 67



- pPDNInactivTimeout3GPP2, 67
- pPdnType, 67
- pPppSessCloseTimer1x, 67
- pPppSessCloseTimerDO, 67
- pPriV6DnsAddress, 67
- pPrimaryV4DnsAddress, 67
- pRATType, 67
- pSecV6DnsAddress, 67
- pSecondaryV4DnsAddress, 67
- pUserId, 67
- pUserIdSize, 67
- LibPackprofile\_3GPP, 56
  - pAPNClass, 60
  - pAPNDisabledFlag, 60
  - pAPNName, 60
  - pAPNnameSize, 60
  - pAddrAllocPref, 60
  - pAuthenticationPref, 60
  - pGPRSMinimumQoS, 60
  - pGPRSRequestedQoS, 61
  - pIPv4AddrPref, 61
  - pIPv6AddPref, 61
  - pImCnFlag, 61
  - pPDNInactivTimeout, 61
  - pPDPtype, 61
  - pPassword, 61
  - pPasswordSize, 61
  - pPcscfAddrUsingDhcp, 61
  - pPcscfAddrUsingPCO, 61
  - pPdpAccessConFlag, 61
  - pPdpContext, 61
  - pPdpDataCompType, 61
  - pPdpHdrCompType, 61
  - pPriDNSIPv4AddPref, 61
  - pPriDNSIPv6addpref, 61
  - pPrimaryID, 61
  - pProfilename, 61
  - pProfilenameSize, 61
  - pQoSClassID, 61
  - pSecDNSIPv4AddPref, 61
  - pSecDNSIPv6addpref, 61
  - pSecondaryFlag, 61
  - pTFTID1Params, 62
  - pTFTID2Params, 62
  - pUMTSMinQoS, 62
  - pUMTSMinQoSSigInd, 62
  - pUMTSReqQoSsigInd, 62
  - pUMTSReqQoS, 62
  - pUsername, 62
  - pUsernameSize, 62
- libSDP\_BuildImagesPreferenceRequest
  - libsdp.h, 559
- libSDP\_CalculateImageMask
  - libsdp.h, 560
- libSDP\_CheckValidFirmwareInfo
  - libsdp.h, 560
- libSDP\_DownloadFW
  - libsdp.h, 560
- libSDP\_ExtractFirmwareParametersByPath
  - libsdp.h, 561
- libSDP\_FirmwareInfo
  - libsdp.h, 557
- libSDP\_Fw\_Type
  - libsdp.h, 558
- libSDP\_GetModelFamily
  - libsdp.h, 562
- libSDP\_GetVersion
  - libsdp.h, 562
- libSDP\_Models
  - libsdp.h, 559
- libSDP\_fwdwl\_error\_codes
  - libsdp.h, 558
- libSDP\_getFileType
  - libsdp.h, 561
- libpack\_GetVersion
  - common.h, 493
- libpack\_log
  - common.h, 493
- LibpackProfile3GPP2, 50
  - pAPNClass3GPP2, 54
  - pAPNEnabled3GPP2, 55
  - pAllowLinger, 54
  - pApnString, 55
  - pApnStringSize, 55
  - pAppPriority, 55
  - pAppType, 55
  - pAuthPassword, 55
  - pAuthPasswordSize, 55
  - pAuthProtocol, 55
  - pAuthRetryCount, 55
  - pAuthTimeout, 55
  - pDataMode, 55
  - pDataRate, 55
  - plpcpAckTimeout, 55
  - plpcpCreqRetryCount, 55
  - plsPcscfAddressNedded, 55
  - pLcpAckTimeout, 55
  - pLcpCreqRetryCount, 55
  - pNegoDnsSrvrPref, 55
  - pPDNInactivTimeout3GPP2, 55
  - pPdnType, 55
  - pPppSessCloseTimer1x, 55
  - pPppSessCloseTimerDO, 55
  - pPriV6DnsAddress, 56
  - pPrimaryV4DnsAddress, 55
  - pRATType, 56
  - pSecV6DnsAddress, 56
  - pSecondaryV4DnsAddress, 56
  - pUserId, 56
  - pUserIdSize, 56
- LibpackProfile3GPP, 44
  - pAPNClass, 49
  - pAPNDisabledFlag, 49
  - pAPNName, 49
  - pAPNnameSize, 49
  - pAddrAllocPref, 49

- pAuthenticationPref, 49
- pGPRSMinimumQoS, 49
- pGPRSRequestedQos, 49
- pIPv4AddrPref, 49
- pIPv6AddrPref, 49
- pImCnFlag, 49
- pPDNInactivTimeout, 49
- pPDPtype, 50
- pPassword, 49
- pPasswordSize, 49
- pPcscfAddrUsingDhcp, 49
- pPcscfAddrUsingPCO, 49
- pPdpAccessConFlag, 49
- pPdpContext, 49
- pPdpDataCompType, 49
- pPdpHdrCompType, 50
- pPriDNSIPv4AddPref, 50
- pPriDNSIPv6addpref, 50
- pPrimaryID, 50
- pProfilename, 50
- pProfilenameSize, 50
- pQosClassID, 50
- pSecDNSIPv4AddPref, 50
- pSecDNSIPv6addpref, 50
- pSecondaryFlag, 50
- pTFTID1Params, 50
- pTFTID2Params, 50
- pUMTSMinQoS, 50
- pUMTSMinQoSsigInd, 50
- pUMTSReqQoSsigInd, 50
- pUMTSReqQoS, 50
- pUsername, 50
- pUsernameSize, 50
- libsdh, 556
  - eFW\_TYPE\_CWE\_NVU, 558
  - eFW\_TYPE\_CWE, 558
  - eFW\_TYPE\_INVALID, 558
  - eFW\_TYPE\_MBN\_GOBI, 558
  - eFW\_TYPE\_MBN, 558
  - eFW\_TYPE\_NVU, 558
  - eFW\_TYPE\_SPK, 558
  - eModel\_9X15, 559
  - eModel\_9X30, 559
  - eModel\_Unknown, 559
  - eSDP\_FWDWL\_ERR\_END, 559
  - eSDP\_FWDWL\_ERR\_FAIL, 559
  - eSDP\_FWDWL\_ERR\_FW\_UPGRADE, 559
  - eSDP\_FWDWL\_ERR\_FW\_VERSION\_FAIL, 559
  - eSDP\_FWDWL\_ERR\_GENERAL, 559
  - eSDP\_FWDWL\_ERR\_INVALID\_DEV, 559
  - eSDP\_FWDWL\_ERR\_INVALID\_PATH, 559
  - eSDP\_FWDWL\_ERR\_PATH\_NOT\_SPECIFIED, 559
  - eSDP\_FWDWL\_ERR\_PATH\_TOO\_LONG, 559
  - eSDP\_FWDWL\_ERR\_PRI\_FAIL, 559
  - eSDP\_FWDWL\_ERR\_SDK, 559
  - eSDP\_FWDWL\_ERR\_SET\_CBK, 559
  - eSDP\_FWDWL\_ERR\_TIMEOUT, 559
  - eSDP\_FWDWL\_SUCCESS, 559
  - FIRMWARE\_INFO\_STRING\_SIZE, 557
  - IMG\_MASK\_CLEAR, 557
  - IMG\_MASK\_GENERIC, 557
  - IMG\_MASK\_MDM, 557
  - IMG\_MASK\_PRI, 557
  - LIBSDP\_CARRIER\_PACKAGE\_SKU, 557
  - LIBSDP\_SKU\_STRING\_LENGTH, 557
  - libSDP\_BuildImagesPreferenceRequest, 559
  - libSDP\_CalculateImageMask, 560
  - libSDP\_CheckValidFirmwareInfo, 560
  - libSDP\_DownloadFW, 560
  - libSDP\_ExtractFirmwareParametersByPath, 561
  - libSDP\_FirmwareInfo, 557
  - libSDP\_Fw\_Type, 558
  - libSDP\_GetModelFamily, 562
  - libSDP\_GetVersion, 562
  - libSDP\_Models, 559
  - libSDP\_fwdwl\_error\_codes, 558
  - libSDP\_getFileType, 561
  - libsdh\_SetReadBlockSize, 563
  - libsdh\_set\_log\_func, 562
  - libsdhlogger, 558
  - libsdh\_SetReadBlockSize
    - libsdh.h, 563
  - libsdh\_set\_log\_func
    - libsdh.h, 562
  - libsdhlogger
    - libsdh.h, 558
  - linkage
    - altSrcInfo\_t, 25
  - list\_type
    - DMScustSettingList, 35
    - DMSgetCustomInput, 36
    - pack\_dms\_GetCustFeaturesV2\_t, 189
  - listEntries
    - FMSPrefImageList, 42
  - listSize
    - FMSImageList, 41
    - FMSPrefImageList, 42
- loc.h, 563
  - eQMI\_LOC\_SESS\_STATUS\_FAILURE, 569
  - eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS, 569
  - eQMI\_LOC\_SESS\_STATUS\_SUCCESS, 569
  - eQMI\_LOC\_SESS\_STATUS\_TIMEOUT, 569
  - LOC\_UINT8\_MAX\_STRING\_SZ, 566
  - LOCEVENTMASKBATCHFULLNOTIFICATION, 566
  - LOCEVENTMASKENGINESTATE, 566
  - LOCEVENTMASKFIXSESSIONSTATE, 566
  - LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION, 566
  - LOCEVENTMASKGEOFENCEBREACHNOTIFICATION, 566
  - LOCEVENTMASKGEOFENCEGENERALALERT, 566
  - LOCEVENTMASKGNSSMEASUREMENTREPORT, 566

- LOCEVENTMASKGNSSSVINFO, 566
- LOCEVENTMASKINJECTPOSITIONREQ, 566
- LOCEVENTMASKINJECTPREDICTEDORBITS↔  
REQ, 567
- LOCEVENTMASKINJECTTIMEREQ, 567
- LOCEVENTMASKINJECTWIFIAPDATAREQ, 567
- LOCEVENTMASKINVALIDVALUE, 567
- LOCEVENTMASKLIVEBATCHEDPOSITIONRE↔  
PORT, 567
- LOCEVENTMASKLOCATIONSERVERCONNE↔  
CTIONREQ, 567
- LOCEVENTMASKMOTIONDATACONTROL, 567
- LOCEVENTMASKNIGEOFENCENOTIFICATION,  
567
- LOCEVENTMASKNINOTIFYVERIFYREQ, 567
- LOCEVENTMASKNMEA, 568
- LOCEVENTMASKPEDOMETERCONTROL, 568
- LOCEVENTMASKPOSITIONREPORT, 568
- LOCEVENTMASKSENSORSTREAMINGREAD↔  
YSTATUS, 568
- LOCEVENTMASKSETSPSTREAMINGREPORT,  
568
- LOCEVENTMASKTIMESYNCREQ, 568
- LOCEVENTMASKVEHICLEDATAAREADYSTAT↔  
US, 568
- LOCEVENTMASKWIFIREQ, 568
- MAX\_SENSOR\_DATA\_LEN, 568
- MAX\_TEMP\_DATA\_LEN, 569
- pack\_loc\_DeleteAssistData, 569
- pack\_loc\_EventRegister, 569
- pack\_loc\_SLQSLOCGetBestAvailPos, 570
- pack\_loc\_SLQSLOCInjectPosition, 571
- pack\_loc\_SLQSLOCInjectSensorData, 571
- pack\_loc\_SLQSLOCInjectUTCTime, 572
- pack\_loc\_SLQSLOCSetCradleMountConfig, 572
- pack\_loc\_SetExtPowerState, 570
- pack\_loc\_SetOperationMode, 570
- pack\_loc\_Start, 572
- pack\_loc\_Stop, 573
- unpack\_loc\_BestAvailPos\_Ind, 573
- unpack\_loc\_DeleteAssistData, 574
- unpack\_loc\_DeleteAssistData\_Ind, 574
- unpack\_loc\_EngineState\_Ind, 574
- unpack\_loc\_EventRegister, 575
- unpack\_loc\_GnssSvInfo\_Ind, 575
- unpack\_loc\_PositionRpt\_Ind, 576
- unpack\_loc\_SLQSLOCGetBestAvailPos, 578
- unpack\_loc\_SLQSLOCInjectPosition, 578
- unpack\_loc\_SLQSLOCInjectSensorData, 578
- unpack\_loc\_SLQSLOCInjectUTCTime, 579
- unpack\_loc\_SLQSLOCSetCradleMountConfig,  
579
- unpack\_loc\_SetExtPowerConfig\_Ind, 576
- unpack\_loc\_SetExtPowerState, 576
- unpack\_loc\_SetOperationMode, 577
- unpack\_loc\_SetOperationMode\_Ind, 577
- unpack\_loc\_Start, 579
- unpack\_loc\_Stop, 580
- loc\_BdsSVInfo, 74
  - len, 74
  - pSV, 74
- loc\_BdsSV, 73
  - id, 74
  - mask, 74
- loc\_CellDb, 74
  - mask, 75
- loc\_ClkInfo, 75
  - mask, 76
- loc\_GnssData, 76
  - mask, 77
- loc\_LocApplicationInfo, 78
  - appNameLength, 79
  - appProviderLength, 79
  - appVersionLength, 79
  - appVersionValid, 79
  - pAppName, 79
  - pAppProvider, 79
  - pAppVersion, 79
- loc\_SVInfo, 84
  - len, 85
  - pSV, 85
- loc\_SV, 83
  - id, 84
  - mask, 84
  - system, 84
- loc\_gpsTime, 78
  - gpsTimeOfWeekMs, 78
  - gpsWeek, 78
- loc\_precisionDilution, 79
  - HDOP, 80
  - PDOP, 80
  - VDOP, 80
- loc\_satelliteInfo, 80
  - azimuth, 82
  - elevation, 82
  - gnssSvId, 82
  - healthStatus, 82
  - snr, 82
  - svInfoMask, 82
  - svListLen, 82
  - svStatus, 82
  - system, 82
  - validMask, 82
- loc\_sensorDataUsage, 82
  - aidingIndicatorMask, 83
  - usageMask, 83
- loc\_svUsedforFix, 85
  - gnssSvUsedList, 85
  - gnssSvUsedList\_len, 85
- localTimeOffset
  - nas\_qaQmi3Gpp2TimeZone, 144
- logString
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 334
- logger
  - common.h, 491
- longName

- unpack\_nas\_SLQSGetPLMNName\_t, 376
- longNameCi
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- longNameEn
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- longNameLen
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- longNameSB
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- longitude
  - pack\_loc\_SLQSLOCInjectPosition\_t, 212
- loopbackMode
  - pack\_wds\_SLQSSSetLoopback\_t, 273
- loopbackMultiplier
  - pack\_wds\_SLQSSSetLoopback\_t, 273
- LteBandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, 329
- lteEmmDI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- lteEmmUI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- lteEsmDI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- lteEsmUI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- LteGsmCellInfo
  - nas\_LTEInfoNeighboringGSM, 126
- LteQci
  - unpack\_qos\_swiQosFlow\_t, 421
- lteRsrpDelta
  - nas\_SLQSSignalStrengthsIndReq, 155
- lteRsrpinfo
  - nas\_SLQSSignalStrengthsInformation, 156
- lteSSInfo, 85
  - rsrp, 86
  - rsrq, 86
  - rssi, 86
  - snr, 86
- lteSnrDelta
  - nas\_SLQSSignalStrengthsIndReq, 155
- lteSnrinfo
  - nas\_SLQSSignalStrengthsInformation, 156
- lter srp
  - unpack\_nas\_SLQSGetSignalStrength\_t, 380
- ltesnr
  - unpack\_nas\_SLQSGetSignalStrength\_t, 380
- m\_FwBuildId
  - CarrierImage\_t, 29
- m\_FwImageld
  - CarrierImage\_t, 29
- m\_PriBuildId
  - CarrierImage\_t, 29
- m\_PrImageld
  - CarrierImage\_t, 29
- m\_nCarrierId
  - CarrierImage\_t, 29
- m\_nFolderId
  - CarrierImage\_t, 29
- m\_nStorage
  - CarrierImage\_t, 29
- MAX\_BUILD\_ID\_LEN
  - dms.h, 502
- MAX\_CDMA\_ENC\_MO\_TXT\_MSG\_SIZE
  - sms.h, 668
- MAX\_DESCRIPTION\_LENGTH
  - uim.h, 690
- MAX\_ICCID\_LENGTH
  - uim.h, 690
- MAX\_MS\_TRANSFER\_ROUTE\_MSG
  - sms.h, 668
- MAX\_MSC\_ADDRESS\_SIZE
  - sms.h, 668
- MAX\_MSE\_TWS\_MSG
  - sms.h, 668
- MAX\_NO\_OF\_APPLICATIONS
  - uim.h, 690
- MAX\_NO\_OF\_SLOTS
  - uim.h, 690
- MAX\_SENSOR\_DATA\_LEN
  - loc.h, 568
- MAX\_SLOTS\_STATUS
  - uim.h, 690
- MAX\_SMS\_LIST\_SIZE
  - sms.h, 668
- MAX\_SMS\_MESSAGE\_SIZE
  - sms.h, 668
- MAX\_TEMP\_DATA\_LEN
  - loc.h, 569
- MAX\_WDS\_3GPP\_CONF\_LTE\_ATTACH\_PROFILE↔\_LIST\_SIZE
  - wds.h, 706
- MCC
  - nas\_CDMA SysInfo, 96
  - nas\_CDMA SysInfoExt, 97
  - nas\_GSM SysInfo, 113
  - nas\_LTE SysInfo, 133
  - nas\_QmiNas3GppNetworkInfo, 144
  - nas\_QmiNas3GppNetworkRAT, 145
  - nas\_QmisNasPcsDigit, 146
  - nas\_WCDMA SysInfo, 175
  - nas\_currentPLMN, 101
  - unpack\_nas\_GetServingNetwork\_t, 368
- MDMCallDuration
  - connectionStatus, 30
- MDMConnStatus
  - connectionStatus, 30
- MEID\_MAX\_SIZE
  - dms.h, 502
- MEIDString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- MINREQBKLEN
  - common.h, 491
- MIN
  - unpack\_dms\_GetVoiceNumber\_t, 321
- MMTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 426

- MNC
  - nas\_CDMA SysInfo, 96
  - nas\_GSM SysInfo, 113
  - nas\_LTE SysInfo, 133
  - nas\_QmiNas3GppNetworkInfo, 144
  - nas\_QmiNas3GppNetworkRAT, 145
  - nas\_QmisNasPcsDigit, 146
  - nas\_WCDMA SysInfo, 176
  - nas\_currentPLMN, 101
  - unpack\_nas\_GetServingNetwork\_t, 368
- MPTlv
  - NASQmiCbkNasSystemSelPrefInd, 185
- MSGID\_AND\_LEN
  - common.h, 491
- MSGID\_DONT\_CARE
  - common.h, 491
- MTMessageInfo
  - newMTMessageTlv, 188
- manString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, 195
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 335
- manufacturer
  - unpack\_dms\_GetManufacturer\_t, 317
- Mask
  - pack\_wds\_SLQSGetDUNCallInfo\_t, 265
- mask
  - loc\_BdsSV, 74
  - loc\_CellDb, 75
  - loc\_ClkInfo, 76
  - loc\_GnssData, 77
  - loc\_SV, 84
  - unpack\_qos\_IPv6TrafCls\_t, 403
  - unpack\_qos\_Tos\_t, 422
- max\_channel\_rx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_↔  
t, 457
- max\_channel\_tx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_↔  
t, 457
- max\_dist
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 244
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- max\_dist\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- max\_time
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 244
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- max\_time\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 430
- MaxAllowedPktSz
  - unpack\_qos\_swiQosFlow\_t, 421
- MaxChanRxRate
  - dunchannelRate, 37
- MaxChanTxRate
  - dunchannelRate, 37
- maxChannelRXRate
  - unpack\_wds\_GetConnectionRate\_t, 445
- maxChannelTXRate
  - unpack\_wds\_GetConnectionRate\_t, 445
- maxDIBitRate
  - LibPackQosClassID, 68
- maxDownlinkBitrate
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- maxImages
  - FMSImageIDEntries, 41
- MaxRXChannelRate
  - unpack\_dms\_GetDeviceCap\_t, 311
- maxRxChannelRate
  - unpack\_dms\_GetDeviceCapabilities\_t, 312
- maxSDUSize
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- MaxTXChannelRate
  - unpack\_dms\_GetDeviceCap\_t, 311
- maxTxChannelRate
  - unpack\_dms\_GetDeviceCapabilities\_t, 312
- maxUIBitRate
  - LibPackQosClassID, 68
- maxUplinkBitrate
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- mcc
  - nas\_CSGID, 100
  - nas\_MNRInfo, 135
  - nas\_netSelectionPref, 136
  - pack\_nas\_SLQSGetPLMNName\_t, 220
  - unpack\_nas\_GetHomeNetwork\_t, 366
- mdmCallDurationActive
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- meanThroughputClass
  - LibPackGPRSRequestedQoS, 44
  - wds\_GPRSQoS, 477
- meid
  - unpack\_dms\_GetSerialNumbers\_t, 321
- meidSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
- meidString
  - unpack\_dms\_SLQSSwiGetSerialNoExt\_t, 336
- message
  - unpack\_sms\_SLQSGetSMS\_t, 427
- message\_type
  - NASOTAMessageTlv, 179
- messageFailureCode
  - unpack\_sms\_SendSMS\_t, 425
- messageFormat
  - pack\_sms\_SendSMS\_t, 239
  - unpack\_sms\_SLQSGetSMS\_t, 427
- messageID
  - unpack\_sms\_SendSMS\_t, 425
- messageIndex
  - pack\_sms\_SLQSGetSMS\_t, 241
  - pack\_sms\_SLQSMModifySMSStatus\_t, 242
  - qmiSmsMessageList, 278
  - sMSMTMessage, 288
- messageList

- unpack\_sms\_SLQSGetSMSList\_t, 427
- messageListSize
  - unpack\_sms\_SLQSGetSMSList\_t, 427
- messageMode
  - SMSSMessageMode, 287
  - unpack\_sms\_SLQSWmsMemoryFullCallback\_↵
    - ind\_t, 428
- MessageModelInfo
  - messageModeTlv, 86
- messageModeTlv, 86
  - MessageModelInfo, 86
  - TlvPresent, 86
- messageSize
  - pack\_sms\_SendSMS\_t, 239
  - unpack\_sms\_SLQSGetSMS\_t, 427
- messageTag
  - pack\_sms\_SLQSMModifySMSStatus\_t, 242
  - qmiSmsMessageList, 278
  - unpack\_sms\_SLQSGetSMS\_t, 427
- MinPolicedPktSz
  - unpack\_qos\_swiQosFlow\_t, 421
- minSize
  - unpack\_dms\_GetVoiceNumber\_t, 321
- minute
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- mipMode
  - unpack\_wds\_GetMobileIP\_t, 448
- mipStatus
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- mipstatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- mnc
  - nas\_CSGID, 100
  - nas\_MNRInfo, 135
  - nas\_netSelectionPref, 136
  - pack\_nas\_SLQSGetPLMNName\_t, 220
  - unpack\_nas\_GetHomeNetwork\_t, 366
- mncPcsDigits
  - nas\_CSGID, 100
- mobileCountryCode
  - sMSEtwsPlmn, 287
- mobileIP
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 271
- mobileNetworkCode
  - sMSEtwsPlmn, 287
- mode
  - pack\_dms\_SetEventReport\_t, 192
  - pack\_dms\_SetPower\_t, 193
  - pack\_loc\_SetOperationMode\_t, 207
  - pack\_wds\_SetMobileIP\_t, 260
- ModePref
  - NASModePreferenceTlv, 178
- modelString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, 195
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 335
- modelid
  - unpack\_dms\_GetModelID\_t, 317
- modelid\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- modemMode
  - nas\_CommInfo, 99
- modemindex
  - pack\_fms\_SetImagesPreference\_t, 203
- month
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- msgid
  - pack\_qmi\_t, 236
  - unpack\_qmi\_t, 401
- msgtype
  - common.h, 492
- Mtu
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 461
- multiplier
  - unpack\_qos\_pktErrRate\_t, 404
- NAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE
  - nas.h, 586
- NAS\_LTE\_CPHY\_SCELL\_STATE\_LITE
  - nas.h, 587
- NAS\_MAX\_DESCRIPTION\_LENGTH
  - nas.h, 586
- NAS\_MAX\_NUM\_NETWORKS
  - nas.h, 586
- NAS\_MAX\_SCC\_RX\_INFO\_INSTANCES
  - nas.h, 586
- NAS\_OTA\_MESSAGE\_MAX\_BUF\_SIZE
  - nas.h, 586
- NAS\_PLMN\_LENGTH
  - nas.h, 586
- NAS\_SERVING\_SYSTEM\_INFO\_MAX\_RADIO\_INT↵
  - ERFACE\_LIST
  - nas.h, 586
- NASBandPreferenceTlv, 176
  - band\_pref, 176
  - TlvPresent, 176
- NASEmergencyModeTlv, 176
  - EmerMode, 177
  - TlvPresent, 177
- NASGWAcqOrderPrefTlv, 177
  - GWAcqOrderPref, 177
  - TlvPresent, 177
- NASLTEBandPreferenceTlv, 177
  - LTETBandPref, 178
  - TlvPresent, 178
- NASLteNasReleaseInfoTlv, 178
  - nas\_major, 178
  - nas\_minor, 178
  - nas\_release, 178
  - TlvPresent, 178
- NASModePreferenceTlv, 178
  - ModePref, 178
  - TlvPresent, 178
- NASNetSelPreferenceTlv, 178



- NetSelPref, [179](#)
- TlvPresent, [179](#)
- NASOTAMessageTlv, [179](#)
  - data\_buf, [179](#)
  - data\_len, [179](#)
  - message\_type, [179](#)
  - TlvPresent, [179](#)
- NASPRLPreferenceTlv, [183](#)
  - PRLPref, [184](#)
  - TlvPresent, [184](#)
- NASPhyCaAggPcellInfo, [179](#)
  - dl\_bw\_value, [180](#)
  - freq, [180](#)
  - iLTEbandValue, [180](#)
  - pci, [180](#)
  - TlvPresent, [180](#)
- NASPhyCaAggScellIDBw, [180](#)
  - dl\_bw\_value, [181](#)
  - TlvPresent, [181](#)
- NASPhyCaAggScellIndType, [181](#)
  - freq, [182](#)
  - pci, [182](#)
  - scell\_state, [182](#)
  - TlvPresent, [182](#)
- NASPhyCaAggScellIndex, [181](#)
  - scell\_idx, [181](#)
  - TlvPresent, [181](#)
- NASPhyCaAggScellInfo, [182](#)
  - dl\_bw\_value, [183](#)
  - freq, [183](#)
  - iLTEbandValue, [183](#)
  - pci, [183](#)
  - scell\_state, [183](#)
  - TlvPresent, [183](#)
- NASQmiCbkNasSwiOTAMessageInd, [184](#)
  - nasRelInfoTlv, [184](#)
  - otaMsgTlv, [184](#)
  - timeTlv, [184](#)
- NASQmiCbkNasSystemSelPrefInd, [184](#)
  - BPTlv, [185](#)
  - EMTlv, [185](#)
  - GWAOPTlv, [185](#)
  - LBPTlv, [185](#)
  - MPTlv, [185](#)
  - NSPTlv, [185](#)
  - PRLPTlv, [185](#)
  - RPTlv, [185](#)
  - SDPTlv, [185](#)
- NASRoamPreferenceTlv, [185](#)
  - RoamPref, [185](#)
  - TlvPresent, [185](#)
- NASServDomainPrefTlv, [185](#)
  - SrvDomainPref, [186](#)
  - TlvPresent, [186](#)
- NASServingSystemInfo, [186](#)
  - csAttachState, [187](#)
  - hdrPersonality, [187](#)
  - psAttachState, [187](#)
  - radioInterfaceList, [187](#)
  - radioInterfaceNo, [187](#)
  - registrationState, [187](#)
  - selectedNetwork, [187](#)
- NASTimeInfoTlv, [187](#)
  - time, [188](#)
  - TlvPresent, [188](#)
- NAI
  - unpack\_wds\_GetMobileIPProfile\_t, [449](#)
- NSPTlv
  - NASQmiCbkNasSystemSelPrefInd, [185](#)
- NWQoSStatus
  - unpack\_qos\_SLQSQosGetNetworkStatus\_t, [407](#)
- naiSize
  - unpack\_wds\_GetMobileIPProfile\_t, [449](#)
- Name
  - unpack\_nas\_GetServingNetwork\_t, [369](#)
- name
  - unpack\_nas\_GetHomeNetwork\_t, [366](#)
  - unpack\_wds\_GetDefaultProfile\_t, [447](#)
- nameSize
  - unpack\_nas\_GetServingNetwork\_t, [369](#)
- nameString
  - pack\_dms\_SLQSSwiSetOSInfo\_t, [196](#)
  - unpack\_dms\_SLQSSwiGetOSInfo\_t, [335](#)
- namelength
  - unpack\_omaDmFotaTlv\_t, [400](#)
- namesize
  - unpack\_wds\_GetDefaultProfile\_t, [447](#)
- nas.h, [580](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_↔ CONFIGURED\_ACTIVATED, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_↔ CONFIGURED\_DEACTIVATED, [586](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_↔ DECONFIGURED, [586](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100, [587](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15, [587](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25, [587](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50, [587](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6, [587](#)
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75, [587](#)
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGU↔ RED\_ACTIVATED\_LITE, [587](#)
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGU↔ RED\_DEACTIVATED\_LITE, [587](#)

- eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFI↔  
GURED\_LITE, 587
- LIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB, 586
- LIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE, 586
- NAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE, 586
- NAS\_LTE\_CPHY\_SCELL\_STATE\_LITE, 587
- NAS\_MAX\_DESCRIPTION\_LENGTH, 586
- NAS\_MAX\_NUM\_NETWORKS, 586
- NAS\_MAX\_SCC\_RX\_INFO\_INSTANCES, 586
- NAS\_OTA\_MESSAGE\_MAX\_BUF\_SIZE, 586
- NAS\_PLMN\_LENGTH, 586
- NAS\_SERVING\_SYSTEM\_INFO\_MAX\_RADIO↔  
\_INTERFACE\_LIST, 586
- pack\_nas\_GetACCOLC, 587
- pack\_nas\_GetANAAAAAuthenticationStatus, 587
- pack\_nas\_GetCDMANetworkParameters, 588
- pack\_nas\_GetHomeNetwork, 588
- pack\_nas\_GetNetworkPreference, 588
- pack\_nas\_GetRFInfo, 589
- pack\_nas\_GetServingNetwork, 589
- pack\_nas\_GetServingNetworkCapabilities, 589
- pack\_nas\_GetSignalStrengths, 589
- pack\_nas\_PerformNetworkScan, 590
- pack\_nas\_SLQSGetNetworkTime, 592
- pack\_nas\_SLQSGetPLMNName, 592
- pack\_nas\_SLQSGetServingSystem, 592
- pack\_nas\_SLQSGetSignalStrength, 593
- pack\_nas\_SLQSGetSysInfo, 593
- pack\_nas\_SLQSGetSysSelectionPref, 593
- pack\_nas\_SLQSInitiateNetworkRegistration, 594
- pack\_nas\_SLQSNasConfigSigInfo2, 594
- pack\_nas\_SLQSNasGetCellLocationInfo, 594
- pack\_nas\_SLQSNasGetSigInfo, 595
- pack\_nas\_SLQSNasIndicationRegisterExt, 595
- pack\_nas\_SLQSNasSwiIndicationRegister, 595
- pack\_nas\_SLQSNasSwiModemStatus, 596
- pack\_nas\_SLQSSetBandPreference, 596
- pack\_nas\_SLQSSetSignalStrengthsCallback, 596
- pack\_nas\_SLQSSetSysSelectionPref, 597
- pack\_nas\_SLQSSwiGetLteCQI, 597
- pack\_nas\_SLQSSwiGetLteSccRxInfo, 598
- pack\_nas\_SetACCOLC, 590
- pack\_nas\_SetLURRejectCallback, 591
- pack\_nas\_SetNetworkPreference, 591
- pack\_nas\_SetRFInfoCallback, 591
- pack\_nas\_SlqsGetLTECphyCAInfo, 591
- unpack\_nas\_GetACCOLC, 598
- unpack\_nas\_GetANAAAAAuthenticationStatus, 598
- unpack\_nas\_GetCDMANetworkParameters, 599
- unpack\_nas\_GetHomeNetwork, 599
- unpack\_nas\_GetNetworkPreference, 599
- unpack\_nas\_GetRFInfo, 599
- unpack\_nas\_GetServingNetwork, 600
- unpack\_nas\_GetServingNetworkCapabilities, 600
- unpack\_nas\_GetSignalStrengths, 600
- unpack\_nas\_PerformNetworkScan, 601
- unpack\_nas\_SLQSGetNetworkTime, 604
- unpack\_nas\_SLQSGetPLMNName, 604
- unpack\_nas\_SLQSGetServingSystem, 604
- unpack\_nas\_SLQSGetSignalStrength, 605
- unpack\_nas\_SLQSGetSysInfo, 605
- unpack\_nas\_SLQSGetSysSelectionPref, 605
- unpack\_nas\_SLQSInitiateNetworkRegistration, 606
- unpack\_nas\_SLQSNasConfigSigInfo2, 606
- unpack\_nas\_SLQSNasGetCellLocationInfo, 606
- unpack\_nas\_SLQSNasGetSigInfo, 607
- unpack\_nas\_SLQSNasIndicationRegisterExt, 607
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind, 607
- unpack\_nas\_SLQSNasSigInfoCallback\_ind, 608
- unpack\_nas\_SLQSNasSwiIndicationRegister, 608
- unpack\_nas\_SLQSNasSwiModemStatus, 608
- unpack\_nas\_SLQSNasSwiOTAMessageCallback↔  
\_ind, 609
- unpack\_nas\_SLQSNasSysInfoCallback\_ind, 609
- unpack\_nas\_SLQSNasTimerCallback\_ind, 609
- unpack\_nas\_SLQSSetBandPreference, 610
- unpack\_nas\_SLQSSetSignalStrengthsCallback, 610
- unpack\_nas\_SLQSSetSysSelectionPref, 610
- unpack\_nas\_SLQSSetSysSelectionPrefCall↔  
Back\_ind, 611
- unpack\_nas\_SLQSSwiGetLteCQI, 611
- unpack\_nas\_SLQSSwiGetLteSccRxInfo, 611
- unpack\_nas\_SetACCOLC, 601
- unpack\_nas\_SetDataCapabilitiesCallback\_ind, 602
- unpack\_nas\_SetEventReportInd, 602
- unpack\_nas\_SetLURRejectCallback, 602
- unpack\_nas\_SetNasLTECphyCAIndCallback\_ind, 602
- unpack\_nas\_SetNetworkPreference, 603
- unpack\_nas\_SetRFInfoCallback, 603
- unpack\_nas\_SetRoamingIndicatorCallback\_ind, 603
- unpack\_nas\_SetServingSystemCallback\_ind, 603
- unpack\_nas\_SlqsGetLTECphyCAInfo, 603
- nas\_AddCDMASysInfo, 87
  - geoSysIdx, 88
  - regPrd, 88
- nas\_AddSysInfo, 88
  - cellBroadcastCap, 88
  - geoSysIdx, 88
- nas\_CDMAECIOThresh, 90
  - CDMAECIOThreshListLen, 91
  - pCDMAECIOThreshList, 91
- nas\_CDMAInfo, 91
  - baseId, 92
  - baseLat, 92
  - baseLong, 92
  - nid, 92
  - refpn, 92
  - sid, 92
- nas\_CDMARSSIThresh, 92
  - CDMARSSIThreshListLen, 92



- pCDMARSSIThreshList, 92
- nas\_CDMASysInfo, 92
  - baseId, 95
  - baseLat, 95
  - baseLong, 95
  - bsInfoValid, 95
  - bsPRev, 95
  - bsPRevValid, 96
  - ccsSupported, 96
  - ccsSupportedValid, 96
  - cdmaSysIdValid, 96
  - isSysPrIMatch, 96
  - isSysPrIMatchValid, 96
  - MCC, 96
  - MNC, 96
  - networkID, 96
  - networkIdValid, 96
  - pRevInUse, 96
  - pRevInUseValid, 96
  - packetZone, 96
  - packetZoneValid, 96
  - sysInfoCDMA, 96
  - systemID, 96
- nas\_CDMASysInfoExt, 96
  - imsi\_11\_12, 97
  - MCC, 97
- nas\_CSGID, 99
  - id, 100
  - mcc, 100
  - mnc, 100
  - mncPcsDigits, 100
  - rat, 100
- nas\_CallBarringSysInfo, 88
  - csBarStatus, 89
  - psBarStatus, 89
- nas\_CommInfo, 98
  - imsRegState, 99
  - modemMode, 99
  - psState, 99
  - systemMode, 99
  - temperature, 99
- nas\_GERANInfo, 105
  - arfcn, 107
  - bsic, 107
  - cellID, 107
  - insNmrCellInfo, 107
  - lac, 107
  - nmrInst, 107
  - plmn, 107
  - rxLev, 107
  - timingAdvance, 107
- nas\_GSMRSSIThresh, 109
  - GSMRSSIThreshListLen, 109
  - pGSMRSSIThreshList, 109
- nas\_GSMSrvStatusInfo, 109
  - isPrefDataPath, 110
  - srvStatus, 110
  - trueSrvStatus, 110
- nas\_GSMSysInfo, 110
  - cellId, 113
  - cellIdValid, 113
  - dtmSupp, 113
  - dtmSuppValid, 113
  - egprsSupp, 113
  - egprsSuppValid, 113
  - lac, 113
  - lacValid, 113
  - MCC, 113
  - MNC, 113
  - networkIdValid, 113
  - regRejectInfoValid, 113
  - rejCause, 113
  - rejectSrvDomain, 113
  - sysInfoGSM, 113
- nas\_HDRECIOThresh, 113
  - HDRECIOThreshListLen, 114
  - pHDRECIOThreshList, 114
- nas\_HDRIOThresh, 114
  - HDRIOThreshListLen, 114
  - pHDRIOThreshList, 114
- nas\_HDRRSSIThresh, 114
  - HDRRSSIThreshListLen, 115
  - pHDRRSSIThreshList, 115
- nas\_HDRSINRThreshold, 115
  - HDRSINRThresholdListLen, 116
  - pHDRSINRThresholdList, 116
- nas\_HDRSysInfo, 116
  - hdrActiveProt, 118
  - hdrActiveProtValid, 118
  - hdrPersonality, 118
  - hdrPersonalityValid, 118
  - is856SysId, 118
  - is856SysIdValid, 118
  - isSysPrIMatch, 118
  - isSysPrIMatchValid, 118
  - sysInfoHDR, 118
- nas\_LTEInfo, 121
  - band, 122
  - bandwidth, 122
  - emmConnState, 122
  - emmState, 122
  - emmSubState, 122
  - RXChan, 122
  - TXChan, 122
- nas\_LTEInfoInterfreq, 123
  - freqsLen, 123
  - InfoInterfreq, 123
  - ueInIdle, 123
- nas\_LTEInfoIntrafreq, 123
  - CellParams, 125
  - cellReselPriority, 125
  - cellsLen, 125
  - earfcn, 125
  - globalCellId, 125
  - plmn, 125
  - sIntraSearch, 125

- sNonIntraSearch, 125
- servingCellId, 125
- tac, 125
- threshServingLow, 125
- ueInIdle, 125
- nas\_LTEInfoNeighboringGSM, 126
  - freqsLen, 126
  - LteGsmCellInfo, 126
  - ueInIdle, 126
- nas\_LTEInfoNeighboringWCDMA, 126
  - freqsLen, 127
  - LTEWCDMACellInfo, 127
  - ueInIdle, 127
- nas\_LTERSRPThresh, 128
  - LTERSRPThreshListLen, 128
  - pLTERSRPThreshList, 128
- nas\_LTERSRQThresh, 128
  - LTERSRQThreshListLen, 129
  - pLTERSRQThreshList, 129
- nas\_LTERSSIThresh, 129
  - LTERSSIThreshListLen, 129
  - pLTERSSIThreshList, 129
- nas\_LTESNRThreshold, 131
  - LTESNRThreshListLen, 131
  - pLTESNRThreshList, 131
- nas\_LTESigRptConfig, 129
  - avgPeriod, 130
  - rptRate, 130
- nas\_LTESysInfo, 131
  - cellId, 133
  - cellIdValid, 133
  - lac, 133
  - lacValid, 133
  - MCC, 133
  - MNC, 133
  - networkIdValid, 133
  - regRejectInfoValid, 133
  - rejCause, 133
  - rejectSrvDomain, 134
  - sysInfoLTE, 134
  - tac, 134
  - tacValid, 134
- nas\_MNRInfo, 135
  - mcc, 135
  - mnc, 135
  - rat, 135
- nas\_PhyCaAggPcellInfo, 138
  - dl\_bw\_value, 138
  - freq, 138
  - iLTEbandValue, 138
  - pci, 138
  - TlvPresent, 139
- nas\_PhyCaAggScellIDBw, 139
  - dl\_bw\_value, 139
  - TlvPresent, 139
- nas\_PhyCaAggScellIndType, 140
  - freq, 140
  - pci, 140
  - scell\_state, 140
  - TlvPresent, 140
- nas\_PhyCaAggScellIndex, 139
  - scell\_idx, 140
  - TlvPresent, 140
- nas\_PhyCaAggScellInfo, 141
  - dl\_bw\_value, 143
  - freq, 143
  - iLTEbandValue, 143
  - pci, 143
  - scell\_state, 143
  - TlvPresent, 143
- nas\_QmiNas3GppNetworkInfo, 144
  - Description, 144
  - Forbidden, 144
  - InUse, 144
  - MCC, 144
  - MNC, 144
  - Preferred, 144
  - Roaming, 144
- nas\_QmiNas3GppNetworkRAT, 145
  - MCC, 145
  - MNC, 145
  - RAT, 145
- nas\_QmisNasPcsDigit, 145
  - includes\_pcs\_digit, 146
  - MCC, 146
  - MNC, 146
- nas\_RFInfoTlv, 147
  - activeBandClass, 147
  - activeChannel, 147
  - radioInterface, 147
  - radioInterfaceSize, 147
  - TlvPresent, 147
- nas\_RejectReasonTlv, 146
  - rejectCause, 146
  - serviceDomain, 146
  - TlvPresent, 146
- nas\_RxSigInfo, 149
  - isRadioTuned, 150
  - rsrp, 150
  - rxChainIndex, 150
  - rxPower, 150
- nas\_SLQSSignalStrengthsIndReq, 154
  - ecioDelta, 155
  - ecioThresholdList, 155
  - ecioThresholdListLen, 155
  - ioDelta, 155
  - lteRsrpDelta, 155
  - lteSnrDelta, 155
  - rsrqDelta, 155
  - rxSignalStrengthDelta, 155
  - sinrDelta, 155
  - sinrThresholdList, 155
  - sinrThresholdListLen, 155
- nas\_SLQSSignalStrengthsInformation, 155
  - ecioInfo, 156
  - errorRateInfo, 156

- io, 156
- lteRsrpinfo, 156
- lteSnrinfo, 156
- rsrqInfo, 156
- rxSignalStrengthInfo, 156
- sinr, 156
- nas\_SLQSSignalStrengthsTlv, 156
  - sSLQSSignalStrengthsInfo, 156
  - TlvPresent, 156
- nas\_SccRxInfo, 151
  - numInstances, 151
  - rsrq, 151
  - sigInfo, 151
  - snr, 151
  - TlvPresent, 152
- nas\_SignalStrengthTlv, 153
  - radiolInterface, 154
  - signalStrength, 154
  - TlvPresent, 154
- nas\_SrvStatusInfo, 156
  - isPrefDataPath, 157
  - srvStatus, 157
- nas\_TDSCDMAECIOThresh, 160
  - pTDSCDMAECIOThreshList, 160
  - TDSCDMAECIOThreshListLen, 160
- nas\_TDSCDMARSCPTThresh, 160
  - pTDSCDMARSCPTThreshList, 161
  - TDSCDMARSCPTThreshListLen, 161
- nas\_TDSCDMARSSIThresh, 161
  - pTDSCDMARSSIThreshList, 161
  - TDSCDMARSSIThreshListLen, 161
- nas\_TDSCDMASINRThresh, 161
  - pTDSCDMASINRThreshList, 162
  - TDSCDMASINRThreshListLen, 162
- nas\_UMTSInfo, 164
  - cellID, 165
  - ecio, 166
  - geranInst, 166
  - GeranInstInfo, 166
  - lac, 166
  - plmn, 166
  - psc, 166
  - rscp, 166
  - UMTSInstInfo, 166
  - uarfcn, 166
  - umtsInst, 166
- nas\_UMTSinstInfo, 166
  - umtsEcio, 167
  - umtsPsc, 167
  - umtsRscp, 167
  - umtsUarfcn, 167
- nas\_UniversalTime, 168
  - day, 169
  - dayOfWeek, 169
  - hour, 169
  - minute, 169
  - month, 169
  - second, 169
  - year, 169
- nas\_WCDMAECIOThresh, 170
  - pWCDMAECIOThreshList, 171
  - WCDMAECIOThreshListLen, 171
- nas\_WCDMAInfoLTENeighborCell, 171
  - UMTSLTENbrCell, 172
  - umtsLTENbrCellLen, 172
  - wcdmaRRCState, 172
- nas\_WCDMARSSIThresh, 172
  - pWCDMARSSIThreshList, 172
  - WCDMARSSIThreshListLen, 172
- nas\_WCDMASysInfo, 172
  - cellId, 175
  - cellIdValid, 175
  - hsCallStatus, 175
  - hsCallStatusValid, 175
  - hsInd, 175
  - hsIndValid, 175
  - lac, 175
  - lacValid, 175
  - MCC, 175
  - MNC, 176
  - networkIdValid, 176
  - psc, 176
  - pscValid, 176
  - regRejectInfoValid, 176
  - rejCause, 176
  - rejectSrvDomain, 176
  - sysInfoWCDMA, 176
- nas\_acqOrderPref, 86
  - acqOrdeLen, 87
  - pAcqOrder, 87
- nas\_callBarStatus, 89
  - csBarStatus, 90
  - psBarStatus, 90
- nas\_cellParams, 97
  - pci, 98
  - rsrp, 98
  - rsrq, 98
  - rssI, 98
  - srxlev, 98
- nas\_currentPLMN, 100
  - MCC, 101
  - MNC, 101
  - netDescr, 101
  - netDescrLength, 101
- nas\_dataSrvCapabilities, 101
  - dataCapabilities, 102
  - dataCapabilitiesLen, 102
- nas\_detailSvcInfo, 102
  - hdrHybrid, 103
  - hdrSrvStatus, 103
  - isSysForbidden, 103
  - srvCapability, 103
  - srvStatus, 103
- nas\_ecioListElement, 104
  - ecio, 104
  - radioIlf, 104

- nas\_errorRateListElement, 104
  - errorRate, 105
  - radiolf, 105
- nas\_geranInstInfo, 107
  - geranArfcn, 108
  - geranBsicBcc, 108
  - geranBsicNcc, 108
  - geranRssi, 108
- nas\_gsmCellInfo, 108
  - arfcn, 109
  - band1900, 109
  - bsicId, 109
  - cellIdValid, 109
  - rssi, 109
  - srxlev, 109
- nas\_infoInterFreq, 118
  - cell\_resel\_priority, 119
  - cellInterFreqParams, 119
  - cells\_len, 119
  - earfcn, 119
  - threshXHigh, 119
  - threshXLow, 119
- nas\_lteGsmCellInfo, 119
  - cellReselPriority, 120
  - cells\_len, 120
  - GsmCellInfo, 120
  - nccPermitted, 120
  - threshGsmHigh, 120
  - threshGsmLow, 120
- nas\_lteRsrpInformation, 127
  - rsrplevel, 127
- nas\_lteSnrinformation, 130
  - snrlevel, 131
- nas\_lteWcdmaCellInfo, 134
  - cellReselPriority, 135
  - cellsLen, 135
  - threshXhigh, 135
  - threshXlow, 135
  - uarfcn, 135
  - WCDMACellInfo, 135
- nas\_major
  - NASLteNasReleaseInfoTlv, 178
- nas\_minor
  - NASLteNasReleaseInfoTlv, 178
- nas\_netSelectionPref, 136
  - mcc, 136
  - mnc, 136
  - netReg, 136
- nas\_nmrCellInfo, 136
  - nmrArfcn, 137
  - nmrBsic, 137
  - nmrCellID, 137
  - nmrLac, 137
  - nmrPlmn, 137
  - nmrRxLev, 138
- nas\_qaQmi3Gpp2TimeZone, 143
  - daylightSavings, 144
  - leapSeconds, 144
  - localTimeOffset, 144
- nas\_release
  - NASLteNasReleaseInfoTlv, 178
- nas\_roamIndList, 147
  - numInstances, 148
  - radiolInterface, 148
  - roamIndicator, 148
- nas\_rsrqInformation, 148
  - radiolf, 149
  - rsrq, 149
- nas\_rxSignalStrengthListElement, 150
  - radiolf, 150
  - rxSignalStrength, 150
- nas\_servSystem, 152
  - csAttachState, 153
  - numRadioInterfaces, 153
  - psAttachState, 153
  - radiolInterface, 153
  - regState, 153
  - selNetwork, 153
- nas\_sysInfoCommon, 157
  - isSysForbidden, 159
  - isSysForbiddenValid, 159
  - roamStatus, 159
  - roamStatusValid, 159
  - srvCapability, 159
  - srvCapabilityValid, 159
  - srvDomain, 159
  - srvDomainValid, 160
- nas\_timeInfo, 162
  - day, 164
  - dayLtSavingAdj, 164
  - dayOfWeek, 164
  - hour, 164
  - minute, 164
  - month, 164
  - radiolInterface, 164
  - second, 164
  - timeZone, 164
  - TlvPresent, 164
  - year, 164
- nas\_umtsLTENbrCell, 167
  - cellsTDD, 168
  - earfcn, 168
  - pci, 168
  - rsrp, 168
  - rsrq, 168
  - srxlev, 168
- nas\_wcdmaCellInfo, 169
  - cpich\_ecno, 170
  - cpich\_rscp, 170
  - psc, 170
  - srxlev, 170
- NasGetLTECphyCaInfo, 177
  - PhyCaAggPcellInfo, 177
  - PhyCaAggScellIDBw, 177
  - PhyCaAggScellIndType, 177
  - PhyCaAggScellIndex, 177

- PhyCaAggScellInfo, 177
- nasRelInfoTlv
  - NASQmiCbkNasSwiOTAMessageInd, 184
- nccPermitted
  - nas\_lteGsmCellInfo, 120
- netDescr
  - nas\_currentPLMN, 101
- netDescrLength
  - nas\_currentPLMN, 101
- netInfoLen
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- netReg
  - nas\_netSelectionPref, 136
- NetSelPref
  - NASNetSelPreferenceTlv, 179
- NetworkID
  - unpack\_nas\_SLQSGetServingSystem\_t, 378
- networkID
  - nas\_CDMA SysInfo, 96
- networkIdValid
  - nas\_CDMA SysInfo, 96
  - nas\_GSM SysInfo, 113
  - nas\_LTE SysInfo, 133
  - nas\_WCDMA SysInfo, 176
- networkInfoLen
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, 456
- NetworkType
  - currNetworkInfo, 32
  - wds\_currNetworkInfo, 471
- NewMMTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 426
- newMTMessageTlv, 188
  - MTMessageInfo, 188
  - TlvPresent, 188
- newPINLen
  - uim\_unblockUIMPIN, 307
- newPINVal
  - uim\_unblockUIMPIN, 307
- newPin
  - pack\_dms\_UIMUnblockPIN\_t, 200
- newValue
  - pack\_dms\_UIMChangePIN\_t, 196
- nextHeader
  - LibPackTFTIDParams, 70
- nid
  - nas\_CDMA Info, 92
  - unpack\_nas\_GetHomeNetwork\_t, 366
- nmrArfcn
  - nas\_nmrCellInfo, 137
- nmrBsic
  - nas\_nmrCellInfo, 137
- nmrCellID
  - nas\_nmrCellInfo, 137
- nmrInst
  - nas\_GERAN Info, 107
- nmrLac
  - nas\_nmrCellInfo, 137
- nmrPlmn
  - nas\_nmrCellInfo, 137
- nmrRxLev
  - nas\_nmrCellInfo, 138
- notification
  - unpack\_omaDmNotificationsTlv\_t, 401
- notificationType
  - sMSEtwsMessage, 286
- notused
  - unpack\_dms\_SetCrashAction\_t, 323
- num\_instances
  - DMScustSettingList, 35
- numApp
  - slotInf, 284
  - uim\_slotInfo, 305
- numCrashes
  - crashInformation, 32
- numEntries
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 332
- NumFlows
  - unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, 411
- numInstances
  - nas\_SccRxInfo, 151
  - nas\_roamIndList, 148
  - wds\_DomainNameList, 477
  - wds\_PCSCFFQDNAddressList, 480
  - wds\_PCSCFIPv4ServerAddressList, 481
- numOpt
  - wds\_DHCPLeaseOptTlv, 472
  - wds\_DHCPv4OptionList, 475
  - wdsDhcpv4OptionList, 487
- numQosFlow
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- numRadioInterfaces
  - nas\_servSystem, 153
- NumRxFilters
  - unpack\_qos\_QosFlowInfo\_t, 406
- numSlot
  - uim\_cardStatus, 297
- NumSupUSBComps
  - unpack\_dms\_GetUSBComp\_t, 321
- NumTxFilters
  - unpack\_qos\_QosFlowInfo\_t, 406
- NxtHdrProto
  - unpack\_qos\_swiQosFilter\_t, 417
- OMADMEEnabled
  - unpack\_swioma\_SLQSOMADMGetSettings\_t, 435
- OfflineReason
  - unpack\_dms\_GetPower\_t, 320
- offset
  - uim\_readTransparentInfo, 301
- oldPINLen
  - uim\_changeUIMPIN, 298
- oldPINVal
  - uim\_changeUIMPIN, 298
- oldValue

- pack\_dms\_UIMChangePIN\_t, 196
- opaqueId
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 215
- operatingMode
  - dms\_OperatingModeTlv, 34
- OperatingModeTlv
  - unpack\_dms\_SetEventReport\_ind\_t, 324
- OperationMode
  - unpack\_dms\_GetPower\_t, 320
- optCode
  - wds\_DHCPopt, 473
  - wds\_DHCPv4Option, 474
  - wdsDhcpv4Option, 486
- optList
  - wds\_DHCPLeaseOptTlv, 472
- optListData
  - wds\_DHCPLeaseOptTlv, 472
- optVal
  - wds\_DHCPv4Option, 474
  - wdsDhcpv4Option, 486
- optValLen
  - wds\_DHCPopt, 473
  - wds\_DHCPv4Option, 474
  - wdsDhcpv4Option, 486
- otaMsgTlv
  - NASQmiCbkNasSwiOTAMessageInd, 184
- p1Status
  - unpack\_dms\_UIMGetPINStatus\_t, 341
- p1UnblockRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, 341
- p1VerifyRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, 341
- p2Status
  - unpack\_dms\_UIMGetPINStatus\_t, 341
- p2UnblockRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, 341
- p2VerifyRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, 341
- p3GPP2TimeInfo
  - unpack\_nas\_SLQSGetNetworkTime\_t, 375
- p3GPPTimeInfo
  - unpack\_nas\_SLQSGetNetworkTime\_t, 375
- p3GppNetworkInfoInstances
  - unpack\_nas\_PerformNetworkScan\_t, 370
- p3GppNetworkInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, 370
- p3gppRelease
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 269
- pAAASPI
  - pack\_wds\_SetMobileIPProfile\_t, 263
- PACK\_WDS\_IPV4
  - wds.h, 706
- PACK\_WDS\_IPV6
  - wds.h, 706
- pAPNClass
  - LibPackprofile\_3GPP, 60
  - LibpackProfile3GPP, 49
- pAPNClass3GPP2
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 54
- pAPNDisabledFlag
  - LibPackprofile\_3GPP, 60
  - LibpackProfile3GPP, 49
- pAPNEnabled3GPP2
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 55
- pAPNName
  - LibPackprofile\_3GPP, 60
  - LibpackProfile3GPP, 49
- pAPNnameSize
  - LibPackprofile\_3GPP, 60
  - LibpackProfile3GPP, 49
- pAcqOrder
  - nas\_acqOrderPref, 87
- pAcqOrderPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 235
- pAddCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 382
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 396
- pAddGSMSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 382
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 396
- pAddHDRSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 382
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 396
- pAddLTESysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 382
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 396
- pAddWCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 382
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 396
- pAddrAllocPref
  - LibPackprofile\_3GPP, 60
  - LibpackProfile3GPP, 49
- pAddress
  - pack\_wds\_SetMobileIPProfile\_t, 263
- pAllowLinger
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 54
- pAltitudeAssumed
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pAltitudeWrtEllipsoid
  - unpack\_loc\_BestAvailPos\_Ind\_t, 349
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pAltitudeWrtMeanSeaLevel
  - unpack\_loc\_BestAvailPos\_Ind\_t, 349
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pApnString
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 55
- pApnStringSize
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 55
- pApnname
  - pack\_wds\_SetDefaultProfile\_t, 259
- pAppName

- loc\_LocApplicationInfo, [79](#)
- pAppPriority
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pAppProvider
  - loc\_LocApplicationInfo, [79](#)
- pAppType
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pAppVersion
  - loc\_LocApplicationInfo, [79](#)
- pApplicationInfo
  - pack\_loc\_Start\_t, [217](#)
- pAuth
  - pack\_wds\_SLQSStartDataSession\_t, [274](#)
- pAuthPassword
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pAuthPassword\_tSize
  - LibPackprofile\_3GPP2, [66](#)
- pAuthPasswordSize
  - LibpackProfile3GPP2, [55](#)
- pAuthProtocol
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pAuthRetryCount
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pAuthTimeout
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pAuthenticationPref
  - LibPackprofile\_3GPP, [60](#)
  - LibpackProfile3GPP, [49](#)
- pAutosdm
  - pack\_swima\_SLQSOMADMSetsSettings\_t, [247](#)
- pBandPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [235](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [386](#)
- pBdsSVInfo
  - pack\_loc\_Delete\_Assist\_Data\_t, [204](#)
- pCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pCDMAECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pCDMAECIOThreshList
  - nas\_CDMAECIOThresh, [91](#)
- pCDMAInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [387](#)
- pCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pCDMARSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pCDMARSSIThreshList
  - nas\_CDMARSSIThresh, [92](#)
- pCDMASigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [390](#)
- pCDMASrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [382](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [396](#)
- pCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [382](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [396](#)
- PCSCFAddrPCO
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [461](#)
- PCSCFFQDNAddrList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [461](#)
- pCSGID
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [235](#)
- pCardResult
  - unpack\_uim\_ReadTransparent\_t, [438](#)
- pCardStatus
  - unpack\_uim\_GetCardStatus\_t, [437](#)
  - unpack\_uim\_SLQSUIMSetStatusChangeCall←  
Back\_ind\_t, [441](#)
- pCellDb
  - pack\_loc\_Delete\_Assist\_Data\_t, [204](#)
- pChangeDuration
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, [221](#)
- pChgDuration
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [235](#)
- pClkInfo
  - pack\_loc\_Delete\_Assist\_Data\_t, [204](#)
- pConfigAltitudeAssumed
  - pack\_loc\_Start\_t, [217](#)
- pCreateProfileOut
  - unpack\_wds\_SLQSCreateProfile\_t, [454](#)
- pCurProfile
  - pack\_wds\_SLQSCreateProfile\_t, [264](#)
- pCurrChannelRateInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pCurrDataBearerTechInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pCurrImgInfo
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [332](#)
- pCurrPrefDataSysInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pCustSettingInfo
  - DMSgetCustomFeatureV2, [36](#)
- pCustSettingList
  - DMSgetCustomFeatureV2, [36](#)
- pDDTMInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [229](#)
- PDOP
  - loc\_precisionDilution, [80](#)
- PDPTtype
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [461](#)
- pDataBearer
  - unpack\_wds\_GetDataBearerTechnology\_t, [446](#)
- pDataBearerTechInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pDataCallStatusChangeInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pDataMode
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)



- pDataRate
  - LibPackprofile\_3GPP2, [66](#)
  - LibpackProfile3GPP2, [55](#)
- pDataSystemStatusChangeInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pDayItSavAdj
  - unpack\_nas\_SLQSNasNetworkTimeCallBack\_↔  
ind\_t, [389](#)
- pDefaultPDNEnabled
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [270](#)
- pDeferTime
  - pack\_swima\_SLQSOMADMSelectSelection\_↔  
t, [246](#)
- pDestSMSContent
  - pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [194](#)
  - packgetDyingGaspCfg, [277](#)
- pDestSMSNum
  - pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [194](#)
  - packgetDyingGaspCfg, [277](#)
- pDormancyStatusInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pDualStandByPrefInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [229](#)
- pEVDOPageMonPerChangeInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [276](#)
- pEmerMode
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [235](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [386](#)
- pEnableNotification
  - pack\_wds\_DHCPv4ClientLeaseChange\_t, [255](#)
- pEnabled
  - pack\_wds\_SetMobileIPProfile\_t, [263](#)
- pEncryptData
  - pack\_uim\_ReadTransparent\_t, [249](#)
- pEncryptedData
  - unpack\_uim\_ReadTransparent\_t, [438](#)
- pEncryptedPIN1
  - pack\_uim\_VerifyPin\_t, [255](#)
  - unpack\_uim\_ChangePin\_t, [436](#)
  - unpack\_uim\_SetPinProtection\_t, [439](#)
  - unpack\_uim\_UnblockPin\_t, [442](#)
  - unpack\_uim\_VerifyPin\_t, [443](#)
- pErrorRateInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [229](#)
- pExtErrCode
  - UnPackGetProfileSettingOut, [469](#)
- pExtErrorCode
  - unpack\_wds\_SLQSMModifyProfile\_t, [462](#)
- pFailureReason
  - unpack\_wds\_SLQSStartDataSession\_t, [467](#)
- pFixId
  - unpack\_loc\_PositionRpt\_Ind\_t, [359](#)
- pFwAutoCheck
  - pack\_swima\_SLQSOMADMSetSettings\_t, [247](#)
- pGERANInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [387](#)
- pGPRSMinimumQoS
  - LibPackprofile\_3GPP, [60](#)
- LibpackProfile3GPP, [49](#)
- pGPRSRequestedQoS
  - LibPackprofile\_3GPP, [61](#)
  - LibpackProfile3GPP, [49](#)
- pGSMCallBarringSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [382](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [396](#)
- pGSMCipherDomainSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [382](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [396](#)
- pGSMRSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pGSMRSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pGSMRSSIThreshList
  - nas\_GSMRSSIThresh, [109](#)
- pGSMSigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [390](#)
- pGSMStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [382](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [396](#)
- pGSMStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [382](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [397](#)
- pGWAcqOrderPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [235](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [386](#)
- pGetCustomInput
  - DMSgetCustomFeatureV2, [36](#)
- pGetDyingGaspCfg
  - unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, [331](#)
- pGetDyingGaspStatistics
  - unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_↔  
\_t, [331](#)
- pGnssData
  - pack\_loc\_Delete\_Assist\_Data\_t, [204](#)
- pGpsTime
  - unpack\_loc\_BestAvailPos\_Ind\_t, [350](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [359](#)
- pHA2002bis
  - pack\_wds\_SetMobileIPParameters\_t, [261](#)
- pHAAAuthenticator
  - pack\_wds\_SetMobileIPParameters\_t, [262](#)
- pHASPI
  - pack\_wds\_SetMobileIPProfile\_t, [263](#)
- pHDRECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pHDRECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pHDRECIOThreshList
  - nas\_HDRECIOThresh, [114](#)
- pHDRIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pHDRIOTThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [225](#)
- pHDRIOTThreshList
  - nas\_HDRIOTThresh, [114](#)
- pHDRNewUATIAssInd



- pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pHRRSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 225
- pHRRSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 225
- pHRRSSIThreshList
  - nas\_HRRSSIThresh, 115
- pHRSINRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 225
- pHRSINRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 225
- pHRSINRThreshList
  - nas\_HRSINRThreshold, 116
- pHDRSessionCloseInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pHDRSigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 390
- pHDRSrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 382
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pHDRSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pHeading
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHeadingUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorCirConf
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
- pHorConfidence
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorEllpConf
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
- pHorReliability
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorUncCircular
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorUncEllipseOrientAzimuth
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorUncEllipseSemiMajor
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorUncEllipseSemiMinor
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pHorizontalAccuracyLvl
  - pack\_loc\_Start\_t, 218
- pHotSwapStatus
  - unpack\_uim\_GetCardStatus\_t, 437
- pHwConfig
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, 272
  - unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t, 466
- plPv4AddrPref
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- plPv6AddrPref
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- plgnoreHotSwapSwitch
  - pack\_uim\_SLQSUIMPowerUp\_t, 252
- plmCnFlag
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- plmImageList
  - pack\_fms\_SetImagesPreference\_t, 203
  - unpack\_fms\_GetImagesPreference\_t, 344
- plndicationToken
  - pack\_uim\_ChangePin\_t, 248
  - pack\_uim\_ReadTransparent\_t, 249
  - pack\_uim\_SetPinProtection\_t, 250
  - pack\_uim\_UnblockPin\_t, 254
  - pack\_uim\_VerifyPin\_t, 255
  - unpack\_uim\_ChangePin\_t, 436
  - unpack\_uim\_ReadTransparent\_t, 438
  - unpack\_uim\_SetPinProtection\_t, 439
  - unpack\_uim\_UnblockPin\_t, 442
  - unpack\_uim\_VerifyPin\_t, 443
- plIntermediateReportState
  - pack\_loc\_Start\_t, 218
- plpcpAckTimeout
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 55
- plpcpCreqRetryCount
  - LibPackprofile\_3GPP2, 66
  - LibpackProfile3GPP2, 55
- plsPcscfAddressNedded
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pKeyReferenceID
  - pack\_uim\_ChangePin\_t, 248
  - pack\_uim\_SetPinProtection\_t, 250
  - pack\_uim\_UnblockPin\_t, 254
  - pack\_uim\_VerifyPin\_t, 255
- plTEAttachProfile
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 270
- plTEAttachProfileList
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 270
- plTEBandPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 386
- plTECphyCa
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- plTEInfo
  - unpack\_nas\_SLQSNasSwiModemStatus\_t, 391
- plTEInfoInterfreq
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 387
- plTEInfoIntrafreq
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 388
- plTEInfoNeighboringGSM
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 388

- pLTEInfoNeighboringWCDMA
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 388
- pLTERSRPDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTERSRPThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTERSRPThreshList
  - nas\_LTERSRPThresh, 128
- pLTERSRQDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTERSRQThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTERSRQThreshList
  - nas\_LTERSRQThresh, 129
- pLTERSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTERSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTERSSIThreshList
  - nas\_LTERSSIThresh, 129
- pLTESNRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTESNRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTESNRThreshList
  - nas\_LTESNRThreshold, 131
- pLTESigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 390
- pLTESigRptConfig
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pLTESrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pLTESysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pLTEVoiceSupportSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pLatitude
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pLcpAckTimeout
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pLcpCreqRetryCount
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pLeapSeconds
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pLinktimer
  - pack\_sms\_SendSMS\_t, 239
- pLongitude
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pMIPStatusInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 276
- pMNAAA
  - pack\_wds\_SetMobileIPProfile\_t, 263
- pMNCIncPCSDigStat
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
- pMNHA
  - pack\_wds\_SetMobileIPProfile\_t, 263
- pMNRInfo
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, 221
- pMagneticDeviation
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pManagedRoamingInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pMessage
  - pack\_sms\_SendSMS\_t, 239
- pMessageIndex
  - pack\_sms\_SLQSDeleteSMS\_t, 240
- pMessageMode
  - pack\_sms\_SLQSDeleteSMS\_t, 240
  - pack\_sms\_SLQSGetSMS\_t, 241
  - pack\_sms\_SLQSGetSMSList\_t, 241
  - pack\_sms\_SLQSModifySMSStatus\_t, 242
- pMessageTag
  - pack\_sms\_SLQSDeleteSMS\_t, 240
- pMinIntervalTime
  - pack\_loc\_Start\_t, 218
- pMncPcsDigitStatus
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, 221
- pMncPcsStatus
  - pack\_nas\_SLQSGetPLMNName\_t, 220
- pMode
  - pack\_wds\_SetMobileIPParameters\_t, 262
- pModePref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 386
- pNAI
  - pack\_wds\_SetMobileIPProfile\_t, 263
- pName
  - pack\_wds\_SetDefaultProfile\_t, 259
- pNegoDnsSrvrPref
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pNetSelPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 386
- pNetworkTimeInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pNumberOfPhySlot
  - unpack\_uim\_SLQSUIMGetSlotsStatus\_t, 441
- pOptList
  - wds\_DHCPv4OptionList, 475
  - wdsDhcpv4OptionList, 487
- pOptVal
  - wds\_DHCPOpt, 473
- pPCSInstance
  - unpack\_nas\_PerformNetworkScan\_t, 370
- pPCSInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, 370
- pPDNInactivTimeout

- LibPackprofile\_3GPP, 61
- LibpackProfile3GPP, 49
- pPDNInactivTimeout3GPP2
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pPDPTYPE
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pPRLPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 386
- pPass
  - pack\_wds\_SLQSStartDataSession\_t, 274
- pPassword
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
  - pack\_wds\_SetDefaultProfile\_t, 259
- pPasswordSize
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- pPcscfAddrUsingDhcp
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- pPcscfAddrUsingPCO
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- pPdnType
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pPdpAccessConFlag
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- pPdpContext
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- pPdpDataCompType
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 49
- pPdpHdrCompType
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pPppSessCloseTimer1x
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pPppSessCloseTimerDO
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pPrecisionDilution
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 359
- pPriDNSIPv4AddPref
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pPriDNSIPv6addpref
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pPriV6DnsAddress
  - LibPackprofile\_3GPP2, 67
- LibpackProfile3GPP2, 56
- pPrimaryHA
  - pack\_wds\_SetMobileIPProfile\_t, 263
- pPrimaryID
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pPrimaryV4DnsAddress
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 55
- pProfileID
  - unpack\_wds\_SLQSCreateProfile\_t, 454
- pProfileId
  - pack\_wds\_SLQSCreateProfile\_t, 264
  - pack\_wds\_SLQSModifyProfile\_t, 268
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, 271
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, 272
- pProfileList
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 270
- pProfileSettings
  - unpack\_wds\_SLQSGetProfileSettings\_t, 460
- pProfileType
  - pack\_wds\_SLQSCreateProfile\_t, 264
  - pack\_wds\_SLQSModifyProfile\_t, 268
- pProfilename
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pProfilenameSize
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pQosClassID
  - LibPackprofile\_3GPP, 61
  - LibpackProfile3GPP, 50
- pRATInstance
  - unpack\_nas\_PerformNetworkScan\_t, 370
- pRATInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, 370
- pRATType
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 56
- pRAT
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
- PRIStrng
  - unpack\_dms\_GetFirmwareRevision\_t, 315
  - unpack\_dms\_GetFirmwareRevisions\_t, 315
- PRLInd
  - unpack\_nas\_SLQSGetServingSystem\_t, 379
- PRLPTlv
  - NASQmiCbkNasSystemSelPrefInd, 185
- PRLPref
  - NASPRLPreferenceTlv, 184
- pRXDroppedCount
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pRXOKBytesLastCall
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pRXOkBytesCount
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pRXPacketErrors
  - unpack\_wds\_GetPacketStatistics\_t, 451

- pRXPacketOverflows
  - unpack\_wds\_GetPacketStatistics\_t, [451](#)
- pRXPacketSuccesses
  - unpack\_wds\_GetPacketStatistics\_t, [451](#)
- pRXTotalBytes
  - unpack\_wds\_GetByteTotals\_t, [444](#)
- pRadioInterface
  - unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t, [389](#)
- pRankIndicatorInd
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [230](#)
- pReRegPeriod
  - pack\_wds\_SetMobileIPParameters\_t, [262](#)
- pReRegTraffic
  - pack\_wds\_SetMobileIPParameters\_t, [262](#)
- pReadResult
  - unpack\_uim\_ReadTransparent\_t, [438](#)
- pReasonMask
  - unpack\_dms\_GetOfflineReason\_t, [319](#)
- pRecurrenceType
  - pack\_loc\_Start\_t, [218](#)
- pRejectReason
  - pack\_swima\_SLQSOMADMSendSelection\_t, [246](#)
- pRemainingRetries
  - unpack\_uim\_ChangePin\_t, [437](#)
  - unpack\_uim\_SetPinProtection\_t, [439](#)
  - unpack\_uim\_UnblockPin\_t, [442](#)
  - unpack\_uim\_VerifyPin\_t, [443](#)
- pReportChannelRate
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [265](#)
- pReportConnStatus
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [265](#)
- pReportDataBearerTech
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [265](#)
- pReportDormStatus
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [265](#)
- pReqSettings
  - pack\_wds\_SLQSGetRuntimeSettings\_t, [267](#)
- pRequestOptionList
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [272](#)
  - unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [466](#)
- pRequestedTag
  - pack\_sms\_SLQSGetSMSList\_t, [241](#)
- pRetryInterval
  - pack\_wds\_SetMobileIPParameters\_t, [262](#)
- pRetryLimit
  - pack\_wds\_SetMobileIPParameters\_t, [262](#)
- pRevInUse
  - nas\_CDMASysInfo, [96](#)
- pRevInUseValid
  - nas\_CDMASysInfo, [96](#)
- pRevTunneling
  - pack\_wds\_SetMobileIPProfile\_t, [263](#)
- pRoamPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [236](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [386](#)
- pRscp
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [390](#)
- PSDomain
  - unpack\_nas\_GetServingNetwork\_t, [369](#)
- pSMSAttemptedFlag
  - packgetDyingGaspStatistics, [278](#)
- pSPC
  - pack\_wds\_SetMobileIPParameters\_t, [262](#)
- pSVInfo
  - pack\_loc\_Delete\_Assist\_Data\_t, [204](#)
- pSatelliteInfo
  - unpack\_loc\_GnssSvInfo\_Ind\_t, [354](#)
- pScanResult
  - unpack\_nas\_PerformNetworkScan\_t, [371](#)
- pSccRxInfo
  - unpack\_nas\_SLQSSwiGetLteSccRxInfo\_t, [394](#)
- pSecDNSIPv4AddPref
  - LibPackprofile\_3GPP, [61](#)
  - LibpackProfile3GPP, [50](#)
- pSecDNSIPv6addpref
  - LibPackprofile\_3GPP, [61](#)
  - LibpackProfile3GPP, [50](#)
- pSecV6DnsAddress
  - LibPackprofile\_3GPP2, [67](#)
  - LibpackProfile3GPP2, [56](#)
- pSecondaryFlag
  - LibPackprofile\_3GPP, [61](#)
  - LibpackProfile3GPP, [50](#)
- pSecondaryHA
  - pack\_wds\_SetMobileIPProfile\_t, [263](#)
- pSecondaryV4DnsAddress
  - LibPackprofile\_3GPP2, [67](#)
  - LibpackProfile3GPP2, [56](#)
- pSensorDataUsage
  - unpack\_loc\_BestAvailPos\_Ind\_t, [350](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [359](#)
- pServingSystemInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [229](#)
- pSigIndReq
  - pack\_nas\_SLQSSetSignalStrengthsCallback\_t, [231](#)
- pSignalStrengthInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [229](#)
- pSourceIP
  - LibPackTFTIDParams, [70](#)
- pSpeedHorizontal
  - unpack\_loc\_BestAvailPos\_Ind\_t, [350](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [359](#)
- pSpeedUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [350](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [360](#)
- pSpeedVertical
  - unpack\_loc\_BestAvailPos\_Ind\_t, [350](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [360](#)
- pSpeedVerticalUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [350](#)
- pSrvDomainPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [236](#)

- unpack\_nas\_SLQSGetSysSelectionPref\_t, 386
- pSrvRegRestriction
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
- pStatMask
  - pack\_wds\_GetPacketStatistics\_t, 257
- pSubscriptionInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pSV
  - loc\_BdsSVInfo, 74
  - loc\_SVInfo, 85
- pSvUsedForFix
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pSysInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pSysInfoNoChange
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pSystemSelectionInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 229
- pTDSCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMAECIOTresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMAECIOTreshList
  - nas\_TDSCDMAECIOTresh, 160
- pTDSCDMARSCPDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMARSCPThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMARSCPThreshList
  - nas\_TDSCDMARSCPThresh, 161
- pTDSCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMARSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMARSSIThreshList
  - nas\_TDSCDMARSSIThresh, 161
- pTDSCDMASINRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMASINRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pTDSCDMASINRThreshList
  - nas\_TDSCDMASINRThresh, 162
- pTDSCDMASigInfoExt
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 390
- pTFTID1Params
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pTFTID2Params
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pTXDroppedCount
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pTXOKBytesLastCall
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pTXOkBytesCount
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pTXPacketErrors
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pTXPacketOverflows
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pTXPacketSuccesses
  - unpack\_wds\_GetPacketStatistics\_t, 451
- pTXTotalBytes
  - unpack\_wds\_GetByteTotals\_t, 444
- pTdsdmaBandPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 236
- pTech
  - pack\_wds\_SLQSStartDataSession\_t, 274
- pTechnologyMask
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pTimeSrc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pTimeStamp
  - packgetDyingGaspStatistics, 278
- pTimeUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 351
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pTimeZone
  - unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t, 389
- pTimer
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, 230
- pTimestampUtc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 350
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pTransferStatInd
  - pack\_wds\_SLQSGetDUNCallInfo\_t, 265
  - pack\_wds\_SLQSWdsSetEventReport\_t, 276
- pUMTSCellID
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 388
- pUMTSInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 388
- pUMTSMinQoS
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pUMTSMinQoS SigInd
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pUMTSReqQoS SigInd
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pUMTSReqQoS
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pUimSlotsStatus
  - unpack\_uim\_SLQSUIMGetSlotsStatus\_t, 441
- pUser
  - pack\_wds\_SLQSStartDataSession\_t, 274
- pUserId
  - LibPackprofile\_3GPP2, 67
  - LibpackProfile3GPP2, 56
- pUserIdSize
  - LibPackprofile\_3GPP2, 67



- LibpackProfile3GPP2, 56
- pUsername
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
  - pack\_wds\_SetDefaultProfile\_t, 259
- pUsernameSize
  - LibPackprofile\_3GPP, 62
  - LibpackProfile3GPP, 50
- pVerboseFailReasonType
  - unpack\_wds\_SLQSSStartDataSession\_t, 467
- pVerboseFailureReason
  - unpack\_wds\_SLQSSStartDataSession\_t, 467
- pVertConfidence
  - unpack\_loc\_BestAvailPos\_Ind\_t, 351
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pVertReliability
  - unpack\_loc\_BestAvailPos\_Ind\_t, 351
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pVertUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 351
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- pWCDMACallBarringSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pWCDMACipherDomainSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pWCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pWCDMAECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pWCDMAECIOThreshList
  - nas\_WCDMAECIOThresh, 171
- pWCDMAInfoLTENeighborCell
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 388
- pWCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pWCDMARSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 226
- pWCDMARSSIThreshList
  - nas\_WCDMARSSIThresh, 172
- pWCDMASigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 390
- pWCDMASrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pWCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 383
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 397
- pXid
  - unpack\_loc\_BestAvailPos\_Ind\_t, 351
- pack\_dms\_ActivateAutomatic
  - dms.h, 502
- pack\_dms\_ActivateAutomatic\_t, 188
  - actCode, 189
- pack\_dms\_GetActivationState
  - dms.h, 502
- pack\_dms\_GetBandCapability
  - dms.h, 503
- pack\_dms\_GetCrashAction
  - dms.h, 503
- pack\_dms\_GetCustFeature
  - dms.h, 503
- pack\_dms\_GetCustFeaturesV2
  - dms.h, 504
- pack\_dms\_GetCustFeaturesV2\_t, 189
  - cust\_id, 189
  - list\_type, 189
  - Tlvresult, 189
- pack\_dms\_GetDeviceCap
  - dms.h, 504
- pack\_dms\_GetDeviceCapabilities
  - dms.h, 504
- pack\_dms\_GetDeviceHardwareRev
  - dms.h, 505
- pack\_dms\_GetDeviceMfr
  - dms.h, 505
- pack\_dms\_GetDeviceSerialNumbers
  - dms.h, 506
- pack\_dms\_GetFSN
  - dms.h, 507
- pack\_dms\_GetFirmwareInfo
  - dms.h, 506
- pack\_dms\_GetFirmwareRevision
  - dms.h, 506
- pack\_dms\_GetFirmwareRevisions
  - dms.h, 507
- pack\_dms\_GetHardwareRevision
  - dms.h, 508
- pack\_dms\_GetIMSI
  - dms.h, 508
- pack\_dms\_GetManufacturer
  - dms.h, 508
- pack\_dms\_GetModelID
  - dms.h, 509
- pack\_dms\_GetNetworkTime
  - dms.h, 509
- pack\_dms\_GetOfflineReason
  - dms.h, 510
- pack\_dms\_GetPRLVersion
  - dms.h, 510
- pack\_dms\_GetPower
  - dms.h, 510
- pack\_dms\_GetSerialNumbers
  - dms.h, 511
- pack\_dms\_GetUSBComp
  - dms.h, 511
- pack\_dms\_GetVoiceNumber
  - dms.h, 512
- pack\_dms\_ResetToFactoryDefaults
  - dms.h, 512
- pack\_dms\_ResetToFactoryDefaults\_t, 189
  - spc, 190
- pack\_dms\_SLQSDmsSwiGetResetInfo
  - dms.h, 516
- pack\_dms\_SLQSDmsSwiIndicationRegister

- dms.h, 516
- pack\_dms\_SLQSDmsSwiIndicationRegister\_t, 193
  - resetInfoInd, 193
- pack\_dms\_SLQSGetBandCapability
  - dms.h, 517
- pack\_dms\_SLQSGetERIFile
  - dms.h, 517
- pack\_dms\_SLQSSwiClearDyingGaspStatistics
  - dms.h, 518
- pack\_dms\_SLQSSwiGetCrashInfo
  - dms.h, 518
- pack\_dms\_SLQSSwiGetCrashInfo\_t, 193
  - clear, 194
- pack\_dms\_SLQSSwiGetDyingGaspCfg
  - dms.h, 518
- pack\_dms\_SLQSSwiGetDyingGaspStatistics
  - dms.h, 519
- pack\_dms\_SLQSSwiGetFirmwareCurr
  - dms.h, 519
- pack\_dms\_SLQSSwiGetFwUpdateStatus
  - dms.h, 520
- pack\_dms\_SLQSSwiGetHostDevInfo
  - dms.h, 520
- pack\_dms\_SLQSSwiGetOSInfo
  - dms.h, 520
- pack\_dms\_SLQSSwiGetSerialNoExt
  - dms.h, 521
- pack\_dms\_SLQSSwiSetDyingGaspCfg
  - dms.h, 521
- pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, 194
  - pDestSMSContent, 194
  - pDestSMSNum, 194
- pack\_dms\_SLQSSwiSetHostDevInfo
  - dms.h, 522
- pack\_dms\_SLQSSwiSetHostDevInfo\_t, 194
  - manString, 195
  - modelString, 195
  - plasmaIDString, 195
  - swVerString, 195
- pack\_dms\_SLQSSwiSetOSInfo
  - dms.h, 522
- pack\_dms\_SLQSSwiSetOSInfo\_t, 195
  - nameString, 196
  - versionString, 196
- pack\_dms\_SLQSUIMGetState
  - dms.h, 522
- pack\_dms\_SetActivationStatusCallback
  - dms.h, 512
- pack\_dms\_SetActivationStatusCallback\_t, 190
  - activationState, 190
- pack\_dms\_SetCrashAction
  - dms.h, 513
- pack\_dms\_SetCrashAction\_t, 190
  - crashAction, 191
- pack\_dms\_SetCustFeature
  - dms.h, 513
- pack\_dms\_SetCustFeature\_t, 191
  - DHCPRelayEnabled, 191
- DisableIMSI, 191
- GPSPMP, 191
- GPSSel, 191
- GpsEnable, 191
- IPFamSupport, 191
- IsVoiceEnabled, 191
- RMAutoConnect, 191
- SMSSupport, 191
- pack\_dms\_SetCustFeaturesV2
  - dms.h, 514
- pack\_dms\_SetCustFeaturesV2\_t, 191
  - cust\_id, 192
  - cust\_value, 192
  - Tlvresult, 192
  - value\_length, 192
- pack\_dms\_SetEventReport
  - dms.h, 514
- pack\_dms\_SetEventReport\_t, 192
  - mode, 192
- pack\_dms\_SetFirmwarePreference
  - dms.h, 515
- pack\_dms\_SetPower
  - dms.h, 515
- pack\_dms\_SetPower\_t, 192
  - mode, 193
  - Tlvresult, 193
- pack\_dms\_SetUSBComp
  - dms.h, 516
- pack\_dms\_SetUSBComp\_t, 193
  - Tlvresult, 193
  - USBComp, 193
- pack\_dms\_UIMChangePIN\_t, 196
  - id, 196
  - newValue, 196
  - oldValue, 196
- pack\_dms\_UIMChangePIN
  - dms.h, 523
- pack\_dms\_UIMGetControlKeyStatus
  - dms.h, 523
- pack\_dms\_UIMGetControlKeyStatus\_t, 196
  - facility, 197
- pack\_dms\_UIMGetICCID\_t, 197
  - Tlvresult, 197
- pack\_dms\_UIMGetICCID
  - dms.h, 524
- pack\_dms\_UIMGetPINStatus
  - dms.h, 524
- pack\_dms\_UIMSetControlKeyProtection
  - dms.h, 524
- pack\_dms\_UIMSetControlKeyProtection\_t, 197
  - facility, 198
  - facilityCk, 198
  - facilityState, 198
- pack\_dms\_UIMSetPINProtection
  - dms.h, 525
- pack\_dms\_UIMSetPINProtection\_t, 198
  - bEnable, 199
  - id, 199

- value, 199
- pack\_dms\_UIMUnblockControlKey
  - dms.h, 525
- pack\_dms\_UIMUnblockControlKey\_t, 199
  - facility, 200
  - facilityCk, 200
- pack\_dms\_UIMUnblockPIN\_t, 200
  - id, 200
  - newPin, 200
  - pukValue, 200
- pack\_dms\_UIMUnblockPIN
  - dms.h, 526
- pack\_dms\_UIMVerifyPIN\_t, 200
  - id, 201
  - value, 201
- pack\_dms\_UIMVerifyPIN
  - dms.h, 526
- pack\_dms\_ValidateSPC
  - dms.h, 526
- pack\_fms\_GetImagesPreference
  - fms.h, 554
- pack\_fms\_GetImagesPreference\_t, 201
  - Tlvresult, 201
- pack\_fms\_GetStoredImages
  - fms.h, 555
- pack\_fms\_GetStoredImages\_t, 201
  - Tlvresult, 202
- pack\_fms\_SetImagesPreference
  - fms.h, 555
- pack\_fms\_SetImagesPreference\_t, 202
  - bForceDownload, 203
  - imageListSize, 203
  - modemindex, 203
  - pImageList, 203
  - Tlvresult, 203
- pack\_loc\_Delete\_Assist\_Data\_t, 203
  - pBdsSVInfo, 204
  - pCellDb, 204
  - pClkInfo, 204
  - pGnssData, 204
  - pSVInfo, 204
  - Tlvresult, 204
- pack\_loc\_DeleteAssistData
  - loc.h, 569
- pack\_loc\_EventRegister
  - loc.h, 569
- pack\_loc\_EventRegister\_t, 204
  - eventRegister, 206
  - Tlvresult, 206
- pack\_loc\_SLQSLOCGetBestAvailPos
  - loc.h, 570
- pack\_loc\_SLQSLOCGetBestAvailPos\_t, 207
  - Tlvresult, 208
  - xid, 208
- pack\_loc\_SLQSLOCInjectPosition
  - loc.h, 571
- pack\_loc\_SLQSLOCInjectPosition\_t, 208
  - altitudeSrcInfo, 211
  - altitudeWrtEllipsoid, 211
  - altitudeWrtMeanSeaLevel, 212
  - has\_altitudeSrcInfo, 212
  - has\_altitudeWrtEllipsoid, 212
  - has\_altitudeWrtMeanSeaLevel, 212
  - has\_horConfidence, 212
  - has\_horReliability, 212
  - has\_horUncCircular, 212
  - has\_latitude, 212
  - has\_longitude, 212
  - has\_positionSrc, 212
  - has\_rawHorConfidence, 212
  - has\_rawHorUncCircular, 212
  - has\_timestampAge, 212
  - has\_timestampUtc, 212
  - has\_vertConfidence, 212
  - has\_vertRelicability, 212
  - has\_vertUnc, 212
  - horConfidence, 212
  - horReliability, 212
  - horUncCircular, 212
  - latitude, 212
  - longitude, 212
  - positionSrc, 212
  - rawHorConfidence, 213
  - rawHorUncCircular, 213
  - timestampAge, 213
  - timestampUtc, 213
  - vertConfidence, 213
  - vertReliability, 213
  - vertUnc, 213
- pack\_loc\_SLQSLOCInjectSensorData
  - loc.h, 571
- pack\_loc\_SLQSLOCInjectSensorData\_t, 213
  - accelTemp, 214
  - acceleroData, 214
  - acceleroTimeSrc, 214
  - gyroData, 214
  - gyroTemp, 215
  - gyroTimeSrc, 215
  - has\_accelTemp, 215
  - has\_acceleroTimeSrc, 215
  - has\_accleroData, 215
  - has\_gyroData, 215
  - has\_gyroTemp, 215
  - has\_gyroTimeSrc, 215
  - has\_opaqueId, 215
  - opaqueId, 215
- pack\_loc\_SLQSLOCInjectUTCTime
  - loc.h, 572
- pack\_loc\_SLQSLOCInjectUTCTime\_t, 215
  - timeMsec, 215
  - timeUncMsec, 215
- pack\_loc\_SLQSLOCSetCradleMountConfig
  - loc.h, 572
- pack\_loc\_SLQSLOCSetCradleMountConfig\_t, 216
  - confidence, 216
  - has\_confidence, 216



- state, 216
- pack\_loc\_SetExtPowerState
  - loc.h, 570
- pack\_loc\_SetExtPowerState\_t, 206
  - extPowerState, 206
  - Tlvresult, 206
- pack\_loc\_SetOperationMode
  - loc.h, 570
- pack\_loc\_SetOperationMode\_t, 207
  - mode, 207
  - Tlvresult, 207
- pack\_loc\_Start
  - loc.h, 572
- pack\_loc\_Start\_t, 216
  - pApplicationInfo, 217
  - pConfigAltitudeAssumed, 217
  - pHorizontalAccuracyLvl, 218
  - pIntermediateReportState, 218
  - pMinIntervalTime, 218
  - pRecurrenceType, 218
  - SessionId, 218
  - Tlvresult, 218
- pack\_loc\_Stop
  - loc.h, 573
- pack\_loc\_Stop\_t, 218
  - SessionId, 218
  - Tlvresult, 218
- pack\_nas\_GetACCOLC
  - nas.h, 587
- pack\_nas\_GetANAAAAuthenticationStatus
  - nas.h, 587
- pack\_nas\_GetCDMANetworkParameters
  - nas.h, 588
- pack\_nas\_GetHomeNetwork
  - nas.h, 588
- pack\_nas\_GetNetworkPreference
  - nas.h, 588
- pack\_nas\_GetRFInfo
  - nas.h, 589
- pack\_nas\_GetServingNetwork
  - nas.h, 589
- pack\_nas\_GetServingNetworkCapabilities
  - nas.h, 589
- pack\_nas\_GetSignalStrengths
  - nas.h, 589
- pack\_nas\_PerformNetworkScan
  - nas.h, 590
- pack\_nas\_SLQSGetNetworkTime
  - nas.h, 592
- pack\_nas\_SLQSGetPLMNName
  - nas.h, 592
- pack\_nas\_SLQSGetPLMNName\_t, 220
  - mcc, 220
  - mnc, 220
  - pMncPcsStatus, 220
- pack\_nas\_SLQSGetServingSystem
  - nas.h, 592
- pack\_nas\_SLQSGetSignalStrength
  - nas.h, 593
- pack\_nas\_SLQSGetSysInfo
  - nas.h, 593
- pack\_nas\_SLQSGetSysSelectionPref
  - nas.h, 593
- pack\_nas\_SLQSIInitiateNetworkRegistration
  - nas.h, 594
- pack\_nas\_SLQSIInitiateNetworkRegistration\_t, 221
  - pChangeDuration, 221
  - pMNRInfo, 221
  - pMncPcsDigitStatus, 221
  - regAction, 221
- pack\_nas\_SLQSNasConfigSigInfo2
  - nas.h, 594
- pack\_nas\_SLQSNasConfigSigInfo2\_t, 222
  - pCDMAECIODelta, 225
  - pCDMAECIOThresh, 225
  - pCDMARSSIDelta, 225
  - pCDMARSSIThresh, 225
  - pGSMRSSIDelta, 225
  - pGSMRSSIThresh, 225
  - pHDRECIODelta, 225
  - pHDRECIOThresh, 225
  - pHDRIODelta, 225
  - pHDRIOThresh, 225
  - pHDDRSSIDelta, 225
  - pHDDRSSIThresh, 225
  - pHDRSINRDelta, 225
  - pHDRSINRThresh, 225
  - pLTERSRPDelta, 226
  - pLTERSRPThresh, 226
  - pLTERSRQDelta, 226
  - pLTERSRQThresh, 226
  - pLTERSSIDelta, 226
  - pLTERSSIThresh, 226
  - pLTESNRDelta, 226
  - pLTESNRThresh, 226
  - pLTESigRptConfig, 226
  - pTDSCDMAECIODelta, 226
  - pTDSCDMAECIOThresh, 226
  - pTDSCDMARSCPDelta, 226
  - pTDSCDMARSCPThresh, 226
  - pTDSCDMARSSIDelta, 226
  - pTDSCDMARSSIThresh, 226
  - pTDSCDMASINRDelta, 226
  - pTDSCDMASINRThresh, 226
  - pWCDMAECIODelta, 226
  - pWCDMAECIOThresh, 226
  - pWCDMARSSIDelta, 226
  - pWCDMARSSIThresh, 226
- pack\_nas\_SLQSNasGetCellLocationInfo
  - nas.h, 594
- pack\_nas\_SLQSNasGetSigInfo
  - nas.h, 595
- pack\_nas\_SLQSNasIndicationRegisterExt
  - nas.h, 595
- pack\_nas\_SLQSNasIndicationRegisterExt\_t, 226
  - pDDTMInd, 229

- pDualStandByPrefInd, 229
- pErrorRateInd, 229
- pHDRNewUATIAssInd, 229
- pHDRSessionCloseInd, 229
- pLTECphyCa, 229
- pManagedRoamingInd, 229
- pNetworkTimeInd, 229
- pServingSystemInd, 229
- pSignalStrengthInd, 229
- pSubscriptionInfoInd, 229
- pSysInfoInd, 229
- pSystemSelectionInd, 229
- pack\_nas\_SLQSNasSwiIndicationRegister
  - nas.h, 595
- pack\_nas\_SLQSNasSwiIndicationRegister\_t, 229
  - gsmUmtsDI, 230
  - gsmUmtsUI, 230
  - lteEmmDI, 230
  - lteEmmUI, 230
  - lteEsmDI, 230
  - lteEsmUI, 230
  - pRankIndicatorInd, 230
  - pTimer, 230
- pack\_nas\_SLQSNasSwiModemStatus
  - nas.h, 596
- pack\_nas\_SLQSSetBandPreference
  - nas.h, 596
- pack\_nas\_SLQSSetSignalStrengthsCallback
  - nas.h, 596
- pack\_nas\_SLQSSetSignalStrengthsCallback\_t, 231
  - bEnable, 231
  - pSigIndReq, 231
- pack\_nas\_SLQSSetSysSelectionPref
  - nas.h, 597
- pack\_nas\_SLQSSetSysSelectionPref\_t, 231
  - pAcqOrderPref, 235
  - pBandPref, 235
  - pCSGID, 235
  - pChgDuration, 235
  - pEmerMode, 235
  - pGWAqOrderPref, 235
  - pLTETBandPref, 236
  - pMNCIncPCSDigStat, 236
  - pModePref, 236
  - pNetSelPref, 236
  - pPRLPref, 236
  - pRAT, 236
  - pRoamPref, 236
  - pSrvDomainPref, 236
  - pSrvRegRestriction, 236
  - pTdsdmaBandPref, 236
- pack\_nas\_SLQSSwiGetLteCQI
  - nas.h, 597
- pack\_nas\_SLQSSwiGetLteScCRxInfo
  - nas.h, 598
- pack\_nas\_SetACCOLC\_t, 218
  - accolc, 219
  - spc, 219
- pack\_nas\_SetACCOLC
  - nas.h, 590
- pack\_nas\_SetLURejectCallback
  - nas.h, 591
- pack\_nas\_SetNetworkPreference
  - nas.h, 591
- pack\_nas\_SetNetworkPreference\_t, 219
  - Duration, 220
  - TechnologyPref, 220
  - Tlvresult, 220
- pack\_nas\_SetRFInfoCallback
  - nas.h, 591
- pack\_nas\_SlqsGetLTECphyCAInfo
  - nas.h, 591
- pack\_qmi\_t, 236
  - msgid, 236
  - svc, 236
  - timeout, 237
  - xid, 237
- pack\_qos\_SLQSQosGetNetworkStatus
  - qos.h, 659
- pack\_qos\_SLQSQosSwiReadApnExtraParams
  - qos.h, 659
- pack\_qos\_SLQSQosSwiReadApnExtraParams\_t, 237
  - apnId, 237
- pack\_qos\_SLQSQosSwiReadDataStats
  - qos.h, 660
- pack\_qos\_SLQSQosSwiReadDataStats\_t, 237
  - apnId, 237
- pack\_qos\_SLQSSetQosEventCallback
  - qos.h, 661
- pack\_qos\_SLQSSetQosEventCallback\_t, 238
  - enable, 238
- pack\_sms\_SLQSDeleteSMS\_t, 239
  - pMessageIndex, 240
  - pMessageMode, 240
  - pMessageTag, 240
  - storageType, 240
- pack\_sms\_SLQSDeleteSMS
  - sms.h, 671
- pack\_sms\_SLQSGetSMS\_t, 240
  - messageIndex, 241
  - pMessageMode, 241
  - storageType, 241
- pack\_sms\_SLQSGetSMSList
  - sms.h, 672
- pack\_sms\_SLQSGetSMSList\_t, 241
  - pMessageMode, 241
  - pRequestedTag, 241
  - storageType, 241
- pack\_sms\_SLQSGetSMS
  - sms.h, 671
- pack\_sms\_SLQSModifySMSStatus
  - sms.h, 672
- pack\_sms\_SLQSModifySMSStatus\_t, 241
  - messageIndex, 242
  - messageTag, 242
  - pMessageMode, 242

- storageType, [242](#)
- pack\_sms\_SendSMS\_t, [238](#)
  - messageFormat, [239](#)
  - messageSize, [239](#)
  - pLinktimer, [239](#)
  - pMessage, [239](#)
- pack\_sms\_SendSMS
  - sms.h, [670](#)
- pack\_sms\_SetNewSMSCallback
  - sms.h, [670](#)
- pack\_sms\_SetNewSMSCallback\_t, [239](#)
  - status, [239](#)
- pack\_swiloc\_SwiLocGetAutoStart
  - swiloc.h, [677](#)
- pack\_swiloc\_SwiLocSetAutoStart
  - swiloc.h, [678](#)
- pack\_swiloc\_SwiLocSetAutoStart\_t, [242](#)
  - fix\_rate, [244](#)
  - fix\_type, [244](#)
  - function, [244](#)
  - max\_dist, [244](#)
  - max\_time, [244](#)
  - set\_fix\_rate, [244](#)
  - set\_fix\_type, [244](#)
  - set\_function, [244](#)
  - set\_max\_dist, [244](#)
  - set\_max\_time, [244](#)
- pack\_swioima\_SLQSOMADMAAlertCallback
  - swioima.h, [680](#)
- pack\_swioima\_SLQSOMADMCancelSession
  - swioima.h, [681](#)
- pack\_swioima\_SLQSOMADMCancelSession\_t, [244](#)
  - sessionType, [244](#)
- pack\_swioima\_SLQSOMADMGetSessionInfo
  - swioima.h, [681](#)
- pack\_swioima\_SLQSOMADMGetSessionInfo\_t, [245](#)
  - SessionType, [245](#)
- pack\_swioima\_SLQSOMADMGetSettings
  - swioima.h, [682](#)
- pack\_swioima\_SLQSOMADMSendSelection
  - swioima.h, [683](#)
- pack\_swioima\_SLQSOMADMSendSelection\_t, [245](#)
  - pDeferTime, [246](#)
  - pRejectReason, [246](#)
  - selection, [246](#)
- pack\_swioima\_SLQSOMADMSetSettings
  - swioima.h, [683](#)
- pack\_swioima\_SLQSOMADMSetSettings\_t, [246](#)
  - FOTAUpdate, [247](#)
  - FOTAdownload, [246](#)
  - pAutosdm, [247](#)
  - pFwAutoCheck, [247](#)
- pack\_swioima\_SLQSOMADMStartSession
  - swioima.h, [684](#)
- pack\_swioima\_SLQSOMADMStartSession\_t, [247](#)
  - sessionType, [247](#)
- pack\_uim\_ChangePin
  - uim.h, [691](#)
- pack\_uim\_ChangePin\_t, [247](#)
  - changePIN, [248](#)
  - EncryptedPIN1, [248](#)
  - pIndicationToken, [248](#)
  - pKeyReferenceID, [248](#)
  - sessionInfo, [248](#)
  - Tlvresult, [248](#)
- pack\_uim\_GetCardStatus
  - uim.h, [691](#)
- pack\_uim\_ReadTransparent
  - uim.h, [692](#)
- pack\_uim\_ReadTransparent\_t, [248](#)
  - fileIndex, [249](#)
  - pEncryptData, [249](#)
  - pIndicationToken, [249](#)
  - readTransparent, [249](#)
  - sessionInfo, [249](#)
  - Tlvresult, [249](#)
- pack\_uim\_SLQSUIEventRegister
  - uim.h, [692](#)
- pack\_uim\_SLQSUIEventRegister\_t, [250](#)
  - eventMask, [251](#)
- pack\_uim\_SLQSUIMGetSlotsStatus
  - uim.h, [693](#)
- pack\_uim\_SLQSUIPowerDown
  - uim.h, [693](#)
- pack\_uim\_SLQSUIPowerDown\_t, [251](#)
  - slot, [251](#)
- pack\_uim\_SLQSUIPowerUp
  - uim.h, [693](#)
- pack\_uim\_SLQSUIPowerUp\_t, [251](#)
  - plgnoreHotSwapSwitch, [252](#)
  - slot, [252](#)
- pack\_uim\_SLQSUIMSwitchSlot
  - uim.h, [694](#)
- pack\_uim\_SLQSUIMSwitchSlot\_t, [252](#)
  - bLogicalSlot, [253](#)
  - ulPhysicalSlot, [253](#)
- pack\_uim\_SetPinProtection
  - uim.h, [692](#)
- pack\_uim\_SetPinProtection\_t, [249](#)
  - EncryptedPIN1, [250](#)
  - pIndicationToken, [250](#)
  - pKeyReferenceID, [250](#)
  - pinProtection, [250](#)
  - sessionInfo, [250](#)
  - Tlvresult, [250](#)
- pack\_uim\_UnblockPin
  - uim.h, [694](#)
- pack\_uim\_UnblockPin\_t, [253](#)
  - EncryptedPIN1, [253](#)
  - pIndicationToken, [254](#)
  - pKeyReferenceID, [254](#)
  - pinProtection, [254](#)
  - sessionInfo, [254](#)
  - Tlvresult, [254](#)
- pack\_uim\_VerifyPin
  - uim.h, [695](#)

- pack\_uim\_VerifyPin\_t, 254
  - pEncryptedPIN1, 255
  - pIndicationToken, 255
  - pKeyReferenceID, 255
  - sessionInfo, 255
  - Tlvresult, 255
  - verifyPIN, 255
- pack\_wds\_DHCPv4ClientLeaseChange
  - wds.h, 706
- pack\_wds\_DHCPv4ClientLeaseChange\_t, 255
  - pEnableNotification, 255
- pack\_wds\_GetAutoconnect
  - wds.h, 707
- pack\_wds\_GetByteTotals
  - wds.h, 707
- pack\_wds\_GetConnectionRate
  - wds.h, 708
- pack\_wds\_GetDataBearerTechnology
  - wds.h, 708
- pack\_wds\_GetDefaultProfile
  - wds.h, 709
- pack\_wds\_GetDefaultProfile\_t, 255
  - profiletype, 256
- pack\_wds\_GetDefaultProfileNum
  - wds.h, 709
- pack\_wds\_GetDefaultProfileNum\_t, 256
  - family, 256
  - type, 256
- pack\_wds\_GetDormancyState
  - wds.h, 709
- pack\_wds\_GetDormancyState\_t, 256
- pack\_wds\_GetLastMobileIPError
  - wds.h, 710
- pack\_wds\_GetLastMobileIPError\_t, 256
- pack\_wds\_GetMobileIP\_t, 256
- pack\_wds\_GetMobileIPProfile
  - wds.h, 711
- pack\_wds\_GetMobileIPProfile\_t, 256
  - index, 257
- pack\_wds\_GetMobileIP
  - wds.h, 710
- pack\_wds\_GetPacketStatistics
  - wds.h, 711
- pack\_wds\_GetPacketStatistics\_t, 257
  - pStatMask, 257
- pack\_wds\_GetPacketStatus
  - wds.h, 712
- pack\_wds\_GetPacketStatus\_t, 257
  - statmask, 257
- pack\_wds\_GetSessionDuration
  - wds.h, 712
- pack\_wds\_GetSessionDuration\_t, 258
- pack\_wds\_GetSessionState
  - wds.h, 713
- pack\_wds\_RMSetTransferStatistics
  - wds.h, 713
- pack\_wds\_RMSetTransferStatistics\_t, 258
  - RmTrasnferStaticsReq, 258
- pack\_wds\_SLQSCreateProfile
  - wds.h, 716
- pack\_wds\_SLQSCreateProfile\_t, 263
  - pCurProfile, 264
  - pProfileId, 264
  - pProfileType, 264
- pack\_wds\_SLQSDeleteProfile
  - wds.h, 717
- pack\_wds\_SLQSDeleteProfile\_t, 264
  - profileIndex, 264
  - profileType, 264
- pack\_wds\_SLQSGet3GPPConfigItem
  - wds.h, 717
- pack\_wds\_SLQSGetCurrDataSystemStat
  - wds.h, 718
- pack\_wds\_SLQSGetCurrDataSystemStat\_t, 264
- pack\_wds\_SLQSGetCurrentChannelRate
  - wds.h, 718
- pack\_wds\_SLQSGetDUNCallInfo
  - wds.h, 719
- pack\_wds\_SLQSGetDUNCallInfo\_t, 264
  - Mask, 265
  - pReportChannelRate, 265
  - pReportConnStatus, 265
  - pReportDataBearerTech, 265
  - pReportDormStatus, 265
  - pTransferStatInd, 265
- pack\_wds\_SLQSGetDataBearerTechnology
  - wds.h, 719
- pack\_wds\_SLQSGetDataBearerTechnology\_t, 264
- pack\_wds\_SLQSGetProfileSettings
  - wds.h, 719
- pack\_wds\_SLQSGetProfileSettings\_t, 265
  - ProfileId, 266
  - ProfileType, 266
- pack\_wds\_SLQSGetRuntimeSettings
  - wds.h, 720
- pack\_wds\_SLQSGetRuntimeSettings\_t, 266
  - pReqSettings, 267
- pack\_wds\_SLQSModifyProfile
  - wds.h, 720
- pack\_wds\_SLQSModifyProfile\_t, 267
  - curProfile, 268
  - pProfileId, 268
  - pProfileType, 268
- pack\_wds\_SLQSResetPacketStatics
  - wds.h, 721
- pack\_wds\_SLQSSGetDHCPv4ClientConfig
  - wds.h, 723
- pack\_wds\_SLQSSGetDHCPv4ClientConfig\_t, 271
  - pProfileId, 271
- pack\_wds\_SLQSSGetLoopback
  - wds.h, 723
- pack\_wds\_SLQSSSetDHCPv4ClientConfig
  - wds.h, 724
- pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t, 271
  - pHwConfig, 272
  - pProfileId, 272

- pRequestOptionList, [272](#)
- pack\_wds\_SLQSSSetLoopback
  - wds.h, [724](#)
- pack\_wds\_SLQSSSetLoopback\_t, [272](#)
  - loopbackMode, [273](#)
  - loopbackMultiplier, [273](#)
- pack\_wds\_SLQSSSet3GPPConfigItem
  - wds.h, [721](#)
- pack\_wds\_SLQSSSet3GPPConfigItem\_t, [268](#)
  - LTEAttachProfileListLen, [269](#)
  - p3gppRelease, [269](#)
  - pDefaultPDNEnabled, [270](#)
  - pLTEAttachProfile, [270](#)
  - pLTEAttachProfileList, [270](#)
  - pProfileList, [270](#)
- pack\_wds\_SLQSSSetIPFamilyPreference
  - wds.h, [722](#)
- pack\_wds\_SLQSSSetIPFamilyPreference\_t, [270](#)
  - IPFamilyPreference, [270](#)
- pack\_wds\_SLQSSSetWdsEventCallback
  - wds.h, [722](#)
- pack\_wds\_SLQSSSetWdsEventCallback\_t, [270](#)
  - currentDataBearer, [271](#)
  - dataBearer, [271](#)
  - dataSystemStatus, [271](#)
  - dormancyStatus, [271](#)
  - interval, [271](#)
  - mobileIP, [271](#)
  - transferStats, [271](#)
- pack\_wds\_SLQSSStartDataSession
  - wds.h, [724](#)
- pack\_wds\_SLQSSStartDataSession\_t, [273](#)
  - pAuth, [274](#)
  - pPass, [274](#)
  - pTech, [274](#)
  - pUser, [274](#)
  - pprofileid3gpp, [274](#)
  - pprofileid3gpp2, [274](#)
- pack\_wds\_SLQSSStopDataSession
  - wds.h, [725](#)
- pack\_wds\_SLQSSStopDataSession\_t, [274](#)
  - psid, [274](#)
- pack\_wds\_SLQSWdsGoActive
  - wds.h, [725](#)
- pack\_wds\_SLQSWdsGoDormant
  - wds.h, [726](#)
- pack\_wds\_SLQSWdsSetEventReport
  - wds.h, [726](#)
- pack\_wds\_SLQSWdsSetEventReport\_t, [274](#)
  - pCurrChannelRateInd, [276](#)
  - pCurrDataBearerTechInd, [276](#)
  - pCurrPrefDataSysInd, [276](#)
  - pDataBearerTechInd, [276](#)
  - pDataCallStatusChangeInd, [276](#)
  - pDataSystemStatusChangeInd, [276](#)
  - pDormancyStatusInd, [276](#)
  - pEVDOPageMonPerChangeInd, [276](#)
  - pMIPStatusInd, [276](#)
  - pTransferStatInd, [276](#)
- pack\_wds\_SLQSWdsSwdPDPRuntimeSettings
  - wds.h, [727](#)
- pack\_wds\_SLQSWdsSwdPDPRuntimeSettings\_t, [276](#)
  - contextId, [276](#)
  - contextType, [276](#)
- pack\_wds\_SetAutoconnect
  - wds.h, [714](#)
- pack\_wds\_SetAutoconnect\_t, [258](#)
  - acroamsetting, [258](#)
  - acsetting, [258](#)
- pack\_wds\_SetDefaultProfile
  - wds.h, [714](#)
- pack\_wds\_SetDefaultProfile\_t, [259](#)
  - authentication, [259](#)
  - ipAddress, [259](#)
  - pApnname, [259](#)
  - pName, [259](#)
  - pPassword, [259](#)
  - pUsername, [259](#)
  - pdpType, [259](#)
  - primaryDNS, [259](#)
  - profileType, [259](#)
  - secondaryDNS, [260](#)
- pack\_wds\_SetDefaultProfileNum
  - wds.h, [714](#)
- pack\_wds\_SetDefaultProfileNum\_t, [260](#)
  - family, [260](#)
  - index, [260](#)
  - type, [260](#)
- pack\_wds\_SetMobileIP\_t, [260](#)
  - mode, [260](#)
- pack\_wds\_SetMobileIPParameters
  - wds.h, [715](#)
- pack\_wds\_SetMobileIPParameters\_t, [260](#)
  - pHA2002bis, [261](#)
  - pHAAAuthenticator, [262](#)
  - pMode, [262](#)
  - pReRegPeriod, [262](#)
  - pReRegTraffic, [262](#)
  - pRetryInterval, [262](#)
  - pRetryLimit, [262](#)
  - pSPC, [262](#)
- pack\_wds\_SetMobileIPProfile
  - wds.h, [716](#)
- pack\_wds\_SetMobileIPProfile\_t, [262](#)
  - index, [263](#)
  - pAAASPI, [263](#)
  - pAddress, [263](#)
  - pEnabled, [263](#)
  - pHASPI, [263](#)
  - pMNAHA, [263](#)
  - pMNAAA, [263](#)
  - pNAI, [263](#)
  - pPrimaryHA, [263](#)
  - pRevTunneling, [263](#)
  - pSecondaryHA, [263](#)
  - spc, [263](#)

- pack\_wds\_SetMobileIP
  - wds.h, 715
- PackCreateProfileOut, 277
  - ExtErrorCode, 277
  - ProfileIndex, 277
  - ProfileType, 277
- package\_name
  - unpack\_omaDmFotaTlv\_t, 400
- packageid\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 314
- packetZone
  - nas\_CDMASysInfo, 96
- packetZoneValid
  - nas\_CDMASysInfo, 96
- packgetDyingGaspCfg, 277
  - pDestSMSContent, 277
  - pDestSMSNum, 277
- packgetDyingGaspStatistics, 277
  - pSMSAttemptedFlag, 278
  - pTimeStamp, 278
- path
  - uim\_fileInfo, 299
- pathLen
  - uim\_fileInfo, 300
- pbPlatform
  - unpack\_dms\_GetOfflineReason\_t, 319
- pci
  - NASPhyCaAggPcellInfo, 180
  - NASPhyCaAggScellIndType, 182
  - NASPhyCaAggScellInfo, 183
  - nas\_PhyCaAggPcellInfo, 138
  - nas\_PhyCaAggScellIndType, 140
  - nas\_PhyCaAggScellInfo, 143
  - nas\_cellParams, 98
  - nas\_umtsLTENbrCell, 168
- pcsfQDNAddress
  - wds\_PCSCFFQDNAddressList, 480
- pdpType
  - pack\_wds\_SetDefaultProfile\_t, 259
- pdptype
  - unpack\_wds\_GetDefaultProfile\_t, 447
- peakRate
  - unpack\_qos\_tokenBucket\_t, 422
- peakThroughputClass
  - LibPackGPRSRequestedQoS, 44
  - wds\_GPRSQoS, 477
- PersistentTechPref
  - unpack\_nas\_GetNetworkPreference\_t, 367
- persoFeature
  - appStats, 28
  - uim\_appStatus, 295
- persoRetries
  - appStats, 28
  - uim\_appStatus, 295
- persoState
  - appStats, 28
  - uim\_appStatus, 295
- persoUnblockRetries
  - appStats, 28
  - uim\_appStatus, 295
- PhyCaAggPcellInfo
  - NasGetLTECphyCaInfo, 177
- PhyCaAggScellDIBw
  - NasGetLTECphyCaInfo, 177
- PhyCaAggScellIndType
  - NasGetLTECphyCaInfo, 177
- PhyCaAggScellIndex
  - NasGetLTECphyCaInfo, 177
- PhyCaAggScellInfo
  - NasGetLTECphyCaInfo, 177
- pin1Len
  - uim\_encryptedPIN1, 299
- pin1Retries
  - appStats, 28
  - uim\_appStatus, 295
- pin1State
  - appStats, 28
  - uim\_appStatus, 295
- pin1Val
  - uim\_encryptedPIN1, 299
- pin2Retries
  - appStats, 28
  - uim\_appStatus, 295
- pin2State
  - appStats, 28
  - uim\_appStatus, 295
- pinID
  - uim\_changeUIMPIN, 298
  - uim\_setPINProtection, 304
  - uim\_unblockUIMPIN, 307
  - uim\_verifyUIMPIN, 308
- pinLen
  - uim\_changeUIMPIN, 298
  - uim\_verifyUIMPIN, 308
- pinLength
  - uim\_setPINProtection, 304
- pinOperation
  - uim\_setPINProtection, 304
- pinProtection
  - pack\_uim\_SetPinProtection\_t, 250
  - pack\_uim\_UnblockPin\_t, 254
- pinVal
  - uim\_changeUIMPIN, 298
  - uim\_verifyUIMPIN, 308
- pinValue
  - uim\_setPINProtection, 304
- PkgDescLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo↔\_t, 433
- PkgDescription
  - unpack\_swima\_SLQSOMADMGetSessionInfo↔\_t, 433
- PkgName
  - unpack\_swima\_SLQSOMADMGetSessionInfo↔\_t, 433
- PkgNameLength



- unpack\_swioma\_SLQSOMADMGetSessionInfo↔  
\_t, [433](#)
- pkgver
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [332](#)
- PktErrRate
  - unpack\_qos\_swiQosFlow\_t, [421](#)
- plasmaIDString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, [195](#)
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, [335](#)
- plmn
  - nas\_GERANInfo, [107](#)
  - nas\_LTEInfoIntrafreq, [125](#)
  - nas\_UMTSInfo, [166](#)
- port
  - unpack\_qos\_Port\_t, [404](#)
- positionSrc
  - pack\_loc\_SLQSLOCInjectPosition\_t, [212](#)
- pprofileid3gpp
  - pack\_wds\_SLQSStartDataSession\_t, [274](#)
- pprofileid3gpp2
  - pack\_wds\_SLQSStartDataSession\_t, [274](#)
- prDNSIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔  
\_t, [468](#)
- prDNSIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔  
\_t, [468](#)
- prPCSCFIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔  
\_t, [468](#)
- prPCSCFIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings↔  
\_t, [468](#)
- Precedence
  - unpack\_qos\_swiQosFilter\_t, [417](#)
- precedenceClass
  - LibPackGPRSRequestedQoS, [44](#)
  - wds\_GPRSQoS, [477](#)
- prefNetwork
  - unpack\_wds\_SLQSGetCurrDataSystemStat↔  
t, [456](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t,  
[465](#)
- Preferred
  - nas\_QmiNas3GppNetworkInfo, [144](#)
- prefixLen
  - unpack\_qos\_IPv6Addr\_t, [403](#)
- priSize
  - unpack\_dms\_GetFirmwareRevisions\_t, [315](#)
- pridns
  - unpack\_wds\_GetDefaultProfile\_t, [447](#)
- pridnsv6
  - unpack\_wds\_GetDefaultProfile\_t, [447](#)
- PrimaryDNSV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [462](#)
- PrimaryDNSV6
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [462](#)
- primaryDNS
  - pack\_wds\_SetDefaultProfile\_t, [259](#)
- primaryHA
  - unpack\_wds\_GetMobileIPProfile\_t, [449](#)
- priver
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [332](#)
- priversion\_str
  - unpack\_dms\_GetFirmwareInfo\_t, [314](#)
- ProfileID
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [462](#)
- ProfileId
  - pack\_wds\_SLQSGetProfileSettings\_t, [266](#)
- profileId
  - wds\_DHCPPProfileIdTlv, [473](#)
  - wds\_DHCPv4ProfileId, [475](#)
  - wdsDhcpv4ProfileId, [487](#)
- ProfileId3GPP2
  - unpack\_qos\_swiQosFlow\_t, [421](#)
- ProfileIdTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, [443](#)
- ProfileIndex
  - PackCreateProfileOut, [277](#)
- profileIndex
  - pack\_wds\_SLQSDeleteProfile\_t, [264](#)
  - wds\_ProfileIdentifier, [481](#)
- profileList
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [456](#)
- ProfileName
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [462](#)
- ProfileType
  - pack\_wds\_SLQSGetProfileSettings\_t, [266](#)
  - PackCreateProfileOut, [277](#)
  - unpack\_wds\_SLQSGetProfileSettings\_t, [460](#)
- profileType
  - pack\_wds\_SLQSDeleteProfile\_t, [264](#)
  - pack\_wds\_SetDefaultProfile\_t, [259](#)
  - wds\_DHCPPProfileIdTlv, [473](#)
  - wds\_DHCPv4ProfileId, [475](#)
  - wds\_ProfileIdentifier, [481](#)
  - wdsDhcpv4ProfileId, [487](#)
- profiletype
  - pack\_wds\_GetDefaultProfile\_t, [256](#)
- Protocol
  - unpack\_nas\_GetCDMANetworkParameters\_t, [365](#)
- psAttachState
  - NASServingSystemInfo, [187](#)
  - nas\_servSystem, [153](#)
- psBarStatus
  - nas\_CallBarringSysInfo, [89](#)
  - nas\_callBarStatus, [90](#)
- psState
  - nas\_CommInfo, [99](#)
- psc
  - nas\_UMTSInfo, [166](#)
  - nas\_WCDMASysInfo, [176](#)
  - nas\_wcdmaCellInfo, [170](#)
- pscValid
  - nas\_WCDMASysInfo, [176](#)
- pscsfIPv4Addr

- wds\_PCSCFIPv4ServerAddressList, 481
- psetting
  - unpack\_wds\_GetAutoconnect\_t, 444
- psid
  - pack\_wds\_SLQSSStopDataSession\_t, 274
  - unpack\_wds\_SLQSSStartDataSession\_t, 467
- puk1Retries
  - appStats, 28
  - uim\_appStatus, 295
- puk2Retries
  - appStats, 28
  - uim\_appStatus, 295
- pukLen
  - uim\_unblockUIMPIN, 307
- pukVal
  - uim\_unblockUIMPIN, 307
- pukValue
  - pack\_dms\_UIMUnblockPIN\_t, 200
- QCI
  - LibPackQosClassID, 68
- QFlowState
  - unpack\_qos\_QosFlowInfo\_t, 406
- QMI pack/unpack (pack), 19
- QMI\_NO\_LTE\_FW\_SUPPORT
  - SwiDataTypes.h, 676
- QMI\_TLV\_PLACEHOLDER
  - SwiDataTypes.h, 676
- qaGobiApiTableBandClasses.h, 612
- qaGobiApiTableCallControlReturnReasons.h, 615
- qaGobiApiTableCallEndReasons.h, 616
- qaGobiApiTableCarrierCodes.h, 631
- qaGobiApiTableCodingScheme.h, 633
- \_\_GOBI\_API\_CODING\_SCHEME\_H\_\_, 636
- qaGobiApiTableGpsCapabilityCodes.h, 636
- qaGobiApiTablePowerModes.h, 637
- qaGobiApiTableRadioInterfaces.h, 637
- qaGobiApiTableRegionCodes.h, 638
- qaGobiApiTableServiceOptions.h, 638
- qaGobiApiTableSupServiceInfoClasses.h, 641
- qaGobiApiTableSwiAudio.h, 641
- qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 642
- qaGobiApiTableVoiceCallEndReasons.h, 643
- qm\_wds\_ds\_profile\_extended\_err\_codes
  - qmerrno.h, 656
- qmerrno.h, 650
  - eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT, 653
  - eQCWWAN\_ERR\_BUFFER\_SZ, 652
  - eQCWWAN\_ERR\_CANCEL\_OP, 653
  - eQCWWAN\_ERR\_DRIVER, 653
  - eQCWWAN\_ERR\_ENUM\_BEGIN, 652
  - eQCWWAN\_ERR\_ENUM\_END, 653
  - eQCWWAN\_ERR\_FILE\_COPY, 653
  - eQCWWAN\_ERR\_FILE\_OPEN, 653
  - eQCWWAN\_ERR\_GENERAL, 652
  - eQCWWAN\_ERR\_INTERNAL, 652
  - eQCWWAN\_ERR\_INVALID\_ARG, 652
  - eQCWWAN\_ERR\_INVALID\_DEVID, 652
  - eQCWWAN\_ERR\_INVALID\_FILE, 652
  - eQCWWAN\_ERR\_INVALID\_QMI\_RSP, 652
  - eQCWWAN\_ERR\_INVALID\_XID, 653
  - eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP, 652
  - eQCWWAN\_ERR\_MEMORY, 652
  - eQCWWAN\_ERR\_MULTIPLE\_DEVICES, 653
  - eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPP↵ORTED, 653
  - eQCWWAN\_ERR\_NO\_CANCELABLE\_OP, 653
  - eQCWWAN\_ERR\_NO\_CONNECTION, 652
  - eQCWWAN\_ERR\_NO\_DEVICE, 652
  - eQCWWAN\_ERR\_NO\_SIGNAL, 653
  - eQCWWAN\_ERR\_NONE, 652
  - eQCWWAN\_ERR\_NULL\_TLV, 656
  - eQCWWAN\_ERR\_OFFLINE, 653
  - eQCWWAN\_ERR\_PDU\_GENERATION, 653
  - eQCWWAN\_ERR\_QMI\_ABORTED, 653
  - eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED, 655
  - eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT, 655
  - eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG, 653
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_F↵AILED, 654
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_L↵OCK, 654
  - eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUP↵ORTED, 655
  - eQCWWAN\_ERR\_QMI\_CALL\_FAILED, 653
  - eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP, 656
  - eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTR↵OL\_FAILED, 655
  - eQCWWAN\_ERR\_QMI\_CAT\_END, 656
  - eQCWWAN\_ERR\_QMI\_CAT\_START, 656
  - eQCWWAN\_ERR\_QMI\_CAUSE\_CODE, 654
  - eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUS↵TED, 653
  - eQCWWAN\_ERR\_QMI\_CONNECT, 652
  - eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE, 653
  - eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY, 654
  - eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_F↵ULL, 654
  - eQCWWAN\_ERR\_QMI\_DISABLED, 655
  - eQCWWAN\_ERR\_QMI\_ENCODING, 654
  - eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAI↵LURE, 656
  - eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED, 656
  - eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL, 655
  - eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT, 655
  - eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED, 654
  - eQCWWAN\_ERR\_QMI\_GENERAL, 654
  - eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRI↵CTED, 655
  - eQCWWAN\_ERR\_QMI\_IFACE, 652
  - eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE, 655



- eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER, [654](#)
- eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN, [653](#)
- eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE, [655](#)
- eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT, [655](#)
- eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCES, [655](#)
- eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND, [654](#)
- eQCWWAN\_ERR\_QMI\_INTERNAL, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ARG, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD, [656](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_INDEX, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ID, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION, [655](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTION, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD, [655](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTION, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF, [653](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP, [656](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION, [654](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID, [653](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG, [653](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_IN\_USE, [654](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_USE, [654](#)
- eQCWWAN\_ERR\_QMI\_MAX, [655](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAILURE, [654](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT, [654](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG, [653](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED, [655](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED, [655](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY, [654](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE, [654](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT, [653](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY, [654](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE, [653](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY, [653](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOULND, [653](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO, [655](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION, [655](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS, [653](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE, [654](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED, [653](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED, [655](#)
- eQCWWAN\_ERR\_QMI\_OFFSET, [653](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTED, [653](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED, [653](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE, [655](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL, [653](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED, [654](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED, [654](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH, [655](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH, [652](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO, [652](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNNSUPPORTED, [654](#)
- eQCWWAN\_ERR\_QMI\_REQ, [652](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO, [652](#)
- eQCWWAN\_ERR\_QMI\_RSP, [652](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER, [655](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG, [655](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE, [654](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID, [654](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP, [654](#)
- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND, [655](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED, [654](#)

- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR, [655](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE, [655](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE, [655](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION, [653](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN, [654](#)
- eQCWWAN\_ERR\_QMI\_WIDTH, [656](#)
- eQCWWAN\_ERR\_RESET, [653](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR, [655](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS, [655](#)
- eQCWWAN\_ERR\_SWICM\_END, [656](#)
- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS, [655](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID, [655](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID, [656](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID, [656](#)
- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED, [655](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED, [655](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED, [655](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS, [656](#)
- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE, [655](#)
- eQCWWAN\_ERR\_SWICM\_START, [655](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT, [655](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN, [655](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP, [655](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN, [655](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP, [655](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED, [656](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND, [656](#)
- eQCWWAN\_ERR\_SWIDCS\_END, [656](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR, [656](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR, [656](#)
- eQCWWAN\_ERR\_SWIDCS\_START, [656](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE, [656](#)
- eQCWWAN\_ERR\_SWIIM\_END, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_ENTER\_DOWNLOAD\_MODE, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETION, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_INVALID\_SLOT\_INDEX, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRENT\_ACTIVE\_IMAGE, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS, [656](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT, [656](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE, [656](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH, [656](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR, [656](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE, [656](#)
- eQCWWAN\_ERR\_SWIIM\_START, [656](#)
- eQCWWAN\_ERR\_SWISM\_END, [656](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND, [656](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED, [656](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG, [656](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED, [656](#)
- eQCWWAN\_ERR\_SWISMS\_START, [656](#)
- eQCWWANError, [652](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENT\_FOR\_PROFILE, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFILE\_FAMILY, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_END, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENT, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_SUBS\_ID, [657](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_INDEX, [657](#)

- NVAL, 657
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_L↔  
EN\_INVALID, 657
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_L↔  
IB\_NOT\_INITED, 657
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL, 657
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_E↔  
ND, 657
- qm\_wds\_ds\_profile\_extended\_err\_codes, 656
- qmiSmsMessageList, 278
  - messageIndex, 278
  - messageTag, 278
- qmiWDSDataBearerTechnology, 278
  - currentNetwork, 279
  - ratMask, 279
  - soMask, 279
- qos.h, 657
  - LIBPACK\_MAX\_QOS\_FILTERS, 659
  - LIBPACK\_MAX\_QOS\_FLOW\_PER\_APN\_STATS, 659
  - LIBPACK\_MAX\_QOS\_FLOWS, 659
  - pack\_qos\_SLQSQosGetNetworkStatus, 659
  - pack\_qos\_SLQSQosSwiReadApnExtraParams, 659
  - pack\_qos\_SLQSQosSwiReadDataStats, 660
  - pack\_qos\_SLQSSetQosEventCallback, 661
  - unpack\_qos\_SLQSQosGetNetworkStatus, 661
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams, 662
  - unpack\_qos\_SLQSQosSwiReadDataStats, 662
  - unpack\_qos\_SLQSSetQosEventCallback, 663
  - unpack\_qos\_SLQSSetQosEventCallback\_ind, 663
  - unpack\_qos\_SLQSSetQosNWStatusCallback\_ind, 664
  - unpack\_qos\_SLQSSetQosPriEventCallback\_ind, 665
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind, 665
- qosDeliveryOrder
  - LibPackUMTSQoS, 72
  - wds\_UMTSMinQoS, 485
- qosFlow
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- QosFlowInfo
  - unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, 411
- RATMask
  - currNetworkInfo, 32
  - wds\_currNetworkInfo, 471
- RAN
  - unpack\_nas\_GetServingNetwork\_t, 369
- RAT
  - nas\_QmiNas3GppNetworkRAT, 145
- RFBandInfoElements, 279
  - activeBandClass, 279
  - activeChannel, 279
  - radioInterface, 279
  - unpack\_nas\_GetRFInfo\_t, 367
- RFTlv
  - unpack\_nas\_SetEventReportInd\_t, 371
- RMAutoConnect
  - pack\_dms\_SetCustFeature\_t, 191
  - unpack\_dms\_GetCustFeature\_t, 310
- RPTlv
  - NASQmiCbkNasSystemSelPrefInd, 185
- RRTlv
  - unpack\_nas\_SetEventReportInd\_t, 371
- RXChan
  - nas\_LTEInfo, 122
- rXDroppedCount
  - unpack\_wds\_GetPacketStatus\_t, 452
- rXOKBytesLastCall
  - unpack\_wds\_GetPacketStatus\_t, 452
- rXOkBytesCount
  - unpack\_wds\_GetPacketStatus\_t, 452
- rXPacketErrors
  - unpack\_wds\_GetPacketStatus\_t, 452
- rXPacketOverflows
  - unpack\_wds\_GetPacketStatus\_t, 452
- rXPacketSuccesses
  - unpack\_wds\_GetPacketStatus\_t, 452
- radio
  - unpack\_nas\_GetSignalStrengths\_t, 370
- radioIf
  - nas\_ecioListElement, 104
  - nas\_errorRateListElement, 105
  - nas\_rsrqInformation, 149
  - nas\_rxSignalStrengthListElement, 150
- Radiolfaces
  - unpack\_dms\_GetDeviceCap\_t, 311
  - unpack\_dms\_GetDeviceCapabilities\_t, 312
  - unpack\_nas\_GetServingNetwork\_t, 369
- RadiolfacesSize
  - unpack\_dms\_GetDeviceCap\_t, 311
  - unpack\_nas\_GetServingNetwork\_t, 369
- radiolfacesSize
  - unpack\_dms\_GetDeviceCapabilities\_t, 312
- radioInterface
  - nas\_RFInfoTlv, 147
  - nas\_SignalStrengthTlv, 154
  - nas\_roamIndList, 148
  - nas\_servSystem, 153
  - nas\_timeInfo, 164
  - RFBandInfoElements, 279
- radioInterfaceList
  - NASServingSystemInfo, 187
- radioInterfaceNo
  - NASServingSystemInfo, 187
- radioInterfaceSize
  - nas\_RFInfoTlv, 147
- range
  - unpack\_qos\_Port\_t, 404
- rat
  - nas\_CSGID, 100
  - nas\_MNRInfo, 135
- ratMask

- qmiWSDDataBearerTechnology, 279
- unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- rawHorConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- rawHorUncCircular
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- readTransparent
  - pack\_uim\_ReadTransparent\_t, 249
- reason
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 413
- reconfigReqd
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_ind\_t, 464
- refData
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 334
- refString
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 334
- refpn
  - nas\_CDMAInfo, 92
- regAction
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, 221
- RegForeignNID
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- RegForeignSID
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- RegHomeSID
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
- regPrd
  - nas\_AddCDMASysInfo, 88
- regRejectInfoValid
  - nas\_GSMSysInfo, 113
  - nas\_LTESysInfo, 133
  - nas\_WCDMASysInfo, 176
- regState
  - nas\_servSystem, 153
- RegistrationState
  - unpack\_nas\_GetServingNetwork\_t, 369
- registrationState
  - NASServingSystemInfo, 187
- rejCause
  - nas\_GSMSysInfo, 113
  - nas\_LTESysInfo, 133
  - nas\_WCDMASysInfo, 176
- rejectCause
  - nas\_RejectReasonTlv, 146
- rejectSrvDomain
  - nas\_GSMSysInfo, 113
  - nas\_LTESysInfo, 134
  - nas\_WCDMASysInfo, 176
- reliabilityClass
  - LibPackGPRSRequestedQoS, 44
  - wds\_GPRSQoS, 477
- resBerRatio
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- ResCode
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 334
- resetInfoInd
  - pack\_dms\_SLQSDmsSwiIndicationRegister\_t, 193
- RetryCount
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_ind\_t, 433
- revTunneling
  - unpack\_wds\_GetMobileIPProfile\_t, 449
- RmTrasnferStaticsReq
  - pack\_wds\_RMSetTransferStatistics\_t, 258
- rmTrasnferStaticsReq, 279
  - bResetStatistics, 280
  - ulMask, 280
- roamIndicator
  - nas\_roamIndList, 148
- RoamIndicatorVal
  - unpack\_nas\_SLQSGetServingSystem\_t, 379
- RoamPref
  - NASRoamPreferenceTlv, 185
- roamStatus
  - nas\_sysInfoCommon, 159
- roamStatusValid
  - nas\_sysInfoCommon, 159
- Roaming
  - nas\_QmiNas3GppNetworkInfo, 144
  - unpack\_nas\_GetCDMANetworkParameters\_t, 365
  - unpack\_nas\_GetServingNetwork\_t, 369
- roaming
  - unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, 374
- RoamingIndicatorList
  - unpack\_nas\_SLQSGetServingSystem\_t, 379
- rptRate
  - nas\_LTESigRptConfig, 130
- rscp
  - nas\_UMTSInfo, 166
  - tdscdmaSigInfoExt, 290
- rsrp
  - lteSSInfo, 86
  - nas\_RxSigInfo, 150
  - nas\_cellParams, 98
  - nas\_umtsLTENbrCell, 168
- rsrplevel
  - nas\_lteRsrpinformation, 127
- rsrq
  - lteSSInfo, 86
  - nas\_SccRxInfo, 151
  - nas\_cellParams, 98
  - nas\_rsrqInformation, 149
  - nas\_umtsLTENbrCell, 168
- rsrqDelta
  - nas\_SLQSSignalStrengthsIndReq, 155
- rsrqInfo
  - nas\_SLQSSignalStrengthsInformation, 156
  - unpack\_nas\_SLQSGetSignalStrength\_t, 380
- rssI
  - cdmaSSInfo, 30
  - hdrSSInfo, 42

- lteSSInfo, [86](#)
- nas\_cellParams, [98](#)
- nas\_gsmCellInfo, [109](#)
- tdscdmaSigInfoExt, [290](#)
- unpack\_nas\_GetSignalStrengths\_t, [370](#)
- rx\_bytes
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [465](#)
- rx\_pkts
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [465](#)
- rxChainIndex
  - nas\_RxSigInfo, [150](#)
- RxDropConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, [424](#)
- rxLev
  - nas\_GERANInfo, [107](#)
- rxOKBytesCount
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [459](#)
- RxOkByteCountTlv
  - unpack\_RMTransferStatistics\_ind\_t, [424](#)
- RxOkConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, [424](#)
- rxPower
  - nas\_RxSigInfo, [150](#)
- RxQFilter
  - unpack\_qos\_QosFlowInfo\_t, [406](#)
- RxQFlowGranted
  - unpack\_qos\_QosFlowInfo\_t, [406](#)
- rxSignalStrength
  - nas\_rxSignalStrengthListElement, [150](#)
- rxSignalStrengthDelta
  - nas\_SLQSSignalStrengthsIndReq, [155](#)
- rxSignalStrengthInfo
  - nas\_SLQSSignalStrengthsInformation, [156](#)
- rxSignalStrengthList
  - unpack\_nas\_SLQSGetSignalStrength\_t, [380](#)
- rxSignalStrengthListLen
  - unpack\_nas\_SLQSGetSignalStrength\_t, [380](#)
- SCI
  - unpack\_nas\_GetCDMANetworkParameters\_t, [365](#)
- SCM
  - unpack\_nas\_GetCDMANetworkParameters\_t, [365](#)
- SDK\_VALIDATE\_INPUT\_PACK\_PARAM\_AND\_FILL↔\_XID
  - common.h, [491](#)
- SDK\_VALIDATE\_INPUT\_PACK\_PARAM
  - common.h, [491](#)
- SDPTlv
  - NASQmiCbkNasSystemSelPrefInd, [185](#)
- SDU\_HDR\_LEN
  - common.h, [491](#)
- SHORT
  - SwiDataTypes.h, [677](#)
- sIntraSearch
  - nas\_LTEInfoIntrafreq, [125](#)
- SLQS\_MAX\_DYING\_GASP\_CFG\_SMS\_CONTENT↔\_LENGTH
  - dms.h, [502](#)
- SLQS\_MAX\_DYING\_GASP\_CFG\_SMS\_NUMBER↔\_LENGTH
  - dms.h, [502](#)
- SLQSSSTlv
  - unpack\_nas\_SetEventReportInd\_t, [372](#)
- sMSCAddress, [284](#)
  - data, [285](#)
  - length, [285](#)
- sMSCAddressInfo
  - sms.h, [668](#)
- sMSCAddressTlv, [285](#)
  - SMSCInfo, [285](#)
  - TlvPresent, [285](#)
- SMSCInfo
  - sMSCAddressTlv, [285](#)
- SMSCSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [426](#)
- sMSEtwMessage, [285](#)
  - data, [286](#)
  - length, [286](#)
  - notificationType, [286](#)
- sMSEtwMessageInfo
  - sms.h, [668](#)
- sMSEtwMessageTlv, [286](#)
  - EtwsMessageInfo, [286](#)
  - TlvPresent, [286](#)
- sMSEtwPlmn, [286](#)
  - mobileCountryCode, [287](#)
  - mobileNetworkCode, [287](#)
- sMSEtwPlmnInfo
  - sms.h, [669](#)
- sMSMTMessage, [287](#)
  - messageIndex, [288](#)
  - storageType, [288](#)
- sMSMTMessageInfo
  - sms.h, [669](#)
- sMSMessageMode, [287](#)
  - messageMode, [287](#)
- sMSMessageModelInfo
  - sms.h, [669](#)
- sMSOnIMSInfo
  - sms.h, [669](#)
- sMSOnIMSTlv, [288](#)
  - IMSInfo, [289](#)
  - TlvPresent, [289](#)
- sMSOnIMS, [288](#)
  - smsOnIMS, [288](#)
- SMSSupport
  - pack\_dms\_SetCustFeature\_t, [191](#)
  - unpack\_dms\_GetCustFeature\_t, [310](#)
- sMSTransferRouteMTMessage, [289](#)
  - ackIndicator, [289](#)
  - data, [289](#)
  - format, [289](#)
  - length, [289](#)
  - transactionID, [289](#)
- sMSTransferRouteMTMessageInfo

- sms.h, 669
- sNonIntraSearch
  - nas\_LTEInfoIntrafreq, 125
- SOMask
  - currNetworkInfo, 32
  - wds\_currNetworkInfo, 471
- SPC\_SIZE
  - dms.h, 502
- sPhyCaAggPcellInfo
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_↔  
ind\_t, 372
- sPhyCaAggScellIDBw
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_↔  
ind\_t, 372
- sPhyCaAggScellIndType
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_↔  
ind\_t, 372
- sPhyCaAggScellIndex
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_↔  
ind\_t, 372
- sPhyCaAggScellInfo
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_↔  
ind\_t, 372
- SSInfo
  - unpack\_nas\_SetServingSystemCallback\_ind\_t,  
374
- sLQSSignalStrengthsInfo
  - nas\_SLQSSignalStrengthsTlv, 156
- SSTlv
  - unpack\_nas\_SetEventReportInd\_t, 372
- SWI\_API
  - SwiDataTypes.h, 676
- SWIWWANCMAPI.h, 689
- scell\_idx
  - NASPhyCaAggScellIndex, 181
  - nas\_PhyCaAggScellIndex, 140
- scell\_state
  - NASPhyCaAggScellIndType, 182
  - NASPhyCaAggScellInfo, 183
  - nas\_PhyCaAggScellIndType, 140
  - nas\_PhyCaAggScellInfo, 143
- sduErrorRatio
  - LibPackUMTSQoS, 72
  - wds\_UMTSMinQoS, 485
- seDNSIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_↔  
\_t, 468
- seDNSIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_↔  
\_t, 469
- sePCSCFIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_↔  
\_t, 469
- sePCSCFIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_↔  
\_t, 469
- secdns
  - unpack\_wds\_GetDefaultProfile\_t, 447
- secdnsv6
  - unpack\_wds\_GetDefaultProfile\_t, 447
- second
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- SecondaryDNSV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- SecondaryDNSV6
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- secondaryDNS
  - pack\_wds\_SetDefaultProfile\_t, 260
- secondaryHA
  - unpack\_wds\_GetMobileIPProfile\_t, 450
- selNetwork
  - nas\_servSystem, 153
- selectedNetwork
  - NASServingSystemInfo, 187
- selection
  - pack\_swima\_SLQSOMADMSendSelection\_↔  
t, 246
- sensorData\_t, 280
  - flags, 281
  - sensorDataLen, 281
  - timeOfFirstSample, 281
  - timeOffset, 281
  - xAxis, 281
  - yAxis, 281
  - zAxis, 281
- sensorDataLen
  - sensorData\_t, 281
- ServerAddrList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- serviceDomain
  - nas\_RejectReasonTlv, 146
- servingCellId
  - nas\_LTEInfoIntrafreq, 125
- ServingSystem
  - unpack\_nas\_SLQSGetServingSystem\_t, 379
- sessionEndReason
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_↔  
\_t, 464
- SessionId
  - pack\_loc\_Start\_t, 218
  - pack\_loc\_Stop\_t, 218
- sessionId
  - unpack\_loc\_PositionRpt\_Ind\_t, 360
- sessionInfo
  - pack\_uim\_ChangePin\_t, 248
  - pack\_uim\_ReadTransparent\_t, 249
  - pack\_uim\_SetPinProtection\_t, 250
  - pack\_uim\_UnblockPin\_t, 254
  - pack\_uim\_VerifyPin\_t, 255
- SessionInfoConfig
  - unpack\_swima\_SLQSOMADMAAlertCallback\_↔  
ind\_t, 431
- SessionInfoFota
  - unpack\_swima\_SLQSOMADMAAlertCallback\_↔  
ind\_t, 431



- SessionInfoNotification
  - unpack\_swima\_SLQSOMADMAAlertCallback\_↔  
ind\_t, [431](#)
- SessionState
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_↔  
\_t, [433](#)
- sessionStatus
  - unpack\_loc\_PositionRpt\_Ind\_t, [360](#)
  - unpack\_omaDmNotificationsTlv\_t, [401](#)
- SessionType
  - pack\_swima\_SLQSOMADMGetSessionInfo\_t,  
[245](#)
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_↔  
\_t, [433](#)
- sessionType
  - pack\_swima\_SLQSOMADMCancelSession\_↔  
t, [244](#)
  - pack\_swima\_SLQSOMADMStartSession\_t, [247](#)
  - uim\_UIMSessionInformation, [306](#)
  - uim\_sessionInformation, [303](#)
  - unpack\_omaDmFotaTlv\_t, [400](#)
- set\_fix\_rate
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [244](#)
- set\_fix\_type
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [244](#)
- set\_function
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [244](#)
- set\_max\_dist
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [244](#)
- set\_max\_time
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [244](#)
- Severity
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_↔  
\_t, [433](#)
- severity
  - unpack\_omaDmFotaTlv\_t, [400](#)
- shortName
  - unpack\_nas\_SLQSGetPLMNName\_t, [376](#)
- shortNameCI
  - unpack\_nas\_SLQSGetPLMNName\_t, [376](#)
- shortNameEn
  - unpack\_nas\_SLQSGetPLMNName\_t, [376](#)
- shortNameLen
  - unpack\_nas\_SLQSGetPLMNName\_t, [376](#)
- shortNameSB
  - unpack\_nas\_SLQSGetPLMNName\_t, [376](#)
- sid
  - nas\_CDMAInfo, [92](#)
  - unpack\_nas\_GetHomeNetwork\_t, [366](#)
- SigInd
  - LibPackUMTSReqQoSsigInd, [73](#)
- sigInfo
  - nas\_SccRxInfo, [151](#)
- signalStrength
  - nas\_SignalStrengthTlv, [154](#)
- signalStrengthReqMask
  - unpack\_nas\_SLQSGetSignalStrength\_t, [380](#)
- SimCapability
  - unpack\_dms\_GetDeviceCap\_t, [311](#)
- simCapability
  - unpack\_dms\_GetDeviceCapabilities\_t, [312](#)
- sinr
  - hdrSSInfo, [42](#)
  - nas\_SLQSSignalStrengthsInformation, [156](#)
  - tdscdmaSigInfoExt, [290](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [380](#)
- sinrDelta
  - nas\_SLQSSignalStrengthsIndReq, [155](#)
- sinrThresholdList
  - nas\_SLQSSignalStrengthsIndReq, [155](#)
- sinrThresholdListLen
  - nas\_SLQSSignalStrengthsIndReq, [155](#)
- sku\_str
  - unpack\_dms\_GetFirmwareInfo\_t, [314](#)
- slot
  - pack\_uim\_SLQSUIMPowerDown\_t, [251](#)
  - pack\_uim\_SLQSUIMPowerUp\_t, [252](#)
- slot\_t, [281](#)
  - bICCIDLength, [282](#)
  - bICCID, [282](#)
  - bLogicalSlot, [282](#)
  - uPhyCardStatus, [282](#)
  - uPhySlotStatus, [282](#)
- slotInf, [282](#)
  - AppStatus, [284](#)
  - cardState, [284](#)
  - errorState, [284](#)
  - numApp, [284](#)
  - upinRetries, [284](#)
  - upinState, [284](#)
  - upukRetries, [284](#)
- SlotInfo
  - uim\_cardStatus, [297](#)
- slots\_t, [284](#)
  - uimSlotStatus, [284](#)
- slotsstatusChange
  - unpack\_uim\_SetUimSlotStatusChangeCallback\_↔  
\_ind\_t, [440](#)
- SlqsProfile3GPP2
  - unpackWdsProfileParam, [469](#)
  - wds\_profileInfo, [482](#)
- SlqsProfile3GPP
  - unpackWdsProfileParam, [469](#)
  - wds\_profileInfo, [482](#)
- sms.h, [666](#)
  - eqmiCbKsetStatus, [670](#)
  - LIBPACK\_QMI\_CBK\_PARAM\_NOCHANGE, [670](#)
  - LIBPACK\_QMI\_CBK\_PARAM\_RESET, [670](#)
  - LIBPACK\_QMI\_CBK\_PARAM\_SET, [670](#)
  - MAX\_CDMA\_ENC\_MO\_TXT\_MSG\_SIZE, [668](#)
  - MAX\_MS\_TRANSFER\_ROUTE\_MSG, [668](#)
  - MAX\_MSC\_ADDRESS\_SIZE, [668](#)
  - MAX\_MSE\_TWS\_MSG, [668](#)
  - MAX\_SMS\_LIST\_SIZE, [668](#)
  - MAX\_SMS\_MESSAGE\_SIZE, [668](#)
  - pack\_sms\_SLQSDeleteSMS, [671](#)

- pack\_sms\_SLQSGetSMSList, 672
- pack\_sms\_SLQSGetSMS, 671
- pack\_sms\_SLQSModifySMSStatus, 672
- pack\_sms\_SendSMS, 670
- pack\_sms\_SetNewSMSCallback, 670
- sMSCAddressInfo, 668
- sMSEtwsMessageInfo, 668
- sMSEtwsPlmnInfo, 669
- sMSMTMessageInfo, 669
- sMSMessageModelInfo, 669
- sMSOnIMSInfo, 669
- sMSTransferRouteMTMessageInfo, 669
- unpack\_sms\_SLQSDeleteSMS, 674
- unpack\_sms\_SLQSGetSMSList, 674
- unpack\_sms\_SLQSGetSMS, 674
- unpack\_sms\_SLQSModifySMSStatus, 675
- unpack\_sms\_SLQSWmsMemoryFullCallback\_ind, 675
- unpack\_sms\_SendSMS, 672
- unpack\_sms\_SetNewSMSCallback, 673
- unpack\_sms\_SetNewSMSCallback\_ind, 673
- smsOnIMS
  - sMSOnIMS, 288
- snr
  - loc\_satelliteInfo, 82
  - lteSSInfo, 86
  - nas\_SccRxInfo, 151
- snrlevel
  - nas\_lteSnrinformation, 131
- soMask
  - qmiWSDDataBearerTechnology, 279
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- Source
  - unpack\_swima\_SLQSOMADMGetSessionInfo←\_t, 433
- source
  - altSrcInfo\_t, 25
  - unpack\_dms\_GetNetworkTime\_t, 318
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, 325
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 326
- sourceIPMask
  - LibPackTFTIDParams, 70
- SourceLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo←\_t, 433
- spc
  - pack\_dms\_ResetToFactoryDefaults\_t, 190
  - pack\_nas\_SetACCOLC\_t, 219
  - pack\_wds\_SetMobileIPProfile\_t, 263
- spn
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- spnEncoding
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- spnLength
  - unpack\_nas\_SLQSGetPLMNName\_t, 376
- srcPortRangeEnd
  - LibPackTFTIDParams, 70
- srcPortRangeStart
  - LibPackTFTIDParams, 70
- srvCapability
  - nas\_detailSvcInfo, 103
  - nas\_sysInfoCommon, 159
- srvCapabilityValid
  - nas\_sysInfoCommon, 159
- srvDomain
  - nas\_sysInfoCommon, 159
- SrvDomainPref
  - NASServDomainPrefTlv, 186
- srvDomainValid
  - nas\_sysInfoCommon, 160
- srvStatus
  - nas\_GSMSrvStatusInfo, 110
  - nas\_SrvStatusInfo, 157
  - nas\_detailSvcInfo, 103
- srxlev
  - nas\_cellParams, 98
  - nas\_gsmCellInfo, 109
  - nas\_umtsLTENbrCell, 168
  - nas\_wcdmaCellInfo, 170
- state
  - pack\_loc\_SLQSLOCSetCradleMountConfig\_t, 216
  - unpack\_dms\_GetActivationState\_t, 309
  - unpack\_dms\_SLQSUIMGetState\_t, 338
  - unpack\_omaDmConfigTlv\_t, 398
  - unpack\_omaDmFotaTlv\_t, 400
  - unpack\_qos\_QosFlowInfoState\_t, 407
- statmask
  - pack\_wds\_GetPacketStatus\_t, 257
- StatsMask
  - transferStatInd, 292
- statsMask
  - wds\_TrStatInd, 482
- StatsPeriod
  - transferStatInd, 292
- statsPeriod
  - wds\_TrStatInd, 482
- Status
  - unpack\_swima\_SLQSOMADMGetSessionInfo←\_t, 434
- status
  - pack\_sms\_SetNewSMSCallback\_t, 239
  - unpack\_loc\_BestAvailPos\_Ind\_t, 351
  - unpack\_loc\_DeleteAssistData\_Ind\_t, 352
  - unpack\_loc\_SetExtPowerConfig\_Ind\_t, 361
  - unpack\_loc\_SetOperationMode\_Ind\_t, 362
  - unpack\_qos\_SLQSSetQosNWStatusCallback←\_ind\_t, 411
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 413
- storageIndex
  - FMSImageldElement, 40
- storageType
  - pack\_sms\_SLQSDeleteSMS\_t, 240
  - pack\_sms\_SLQSGetSMS\_t, 241



- pack\_sms\_SLQSGetSMSList\_t, 241
- pack\_sms\_SLQSModifySMSStatus\_t, 242
- sMSMTMessage, 288
- unpack\_sms\_SLQSWmsMemoryFullCallback\_ind\_t, 428
- Streaming Download Protocol (sdp), 20
- String
  - unpack\_dms\_GetDeviceHardwareRev\_t, 312
  - unpack\_dms\_GetDeviceMfr\_t, 313
  - unpack\_dms\_GetFSN\_t, 316
  - unpack\_dms\_UIMGetICCID\_t, 339
- stringSize
  - unpack\_dms\_GetDeviceHardwareRev\_t, 312
  - unpack\_dms\_GetDeviceMfr\_t, 313
  - unpack\_dms\_UIMGetICCID\_t, 339
- subnetMask
  - unpack\_qos\_IPv4Addr\_t, 402
- SubnetMaskV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- SupUSBComps
  - unpack\_dms\_GetUSBComp\_t, 321
- svInfoMask
  - loc\_satelliteInfo, 82
- svListLen
  - loc\_satelliteInfo, 82
- svStatus
  - loc\_satelliteInfo, 82
- svc
  - pack\_qmi\_t, 236
- sw1
  - uim\_cardResult, 296
- sw2
  - uim\_cardResult, 296
- swVerString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, 195
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 335
- SwiDataTypes.h, 676
  - BOOL, 677
  - BYTE, 677
  - CHAR, 677
  - FLOAT, 677
  - INT32, 677
  - INT8, 677
  - LPCSTR, 677
  - QMI\_NO\_LTE\_FW\_SUPPORT, 676
  - QMI\_TLV\_PLACEHOLDER, 676
  - SHORT, 677
  - SWI\_API, 676
  - ULONGLONG, 677
  - ULONG, 677
  - UNUSEDPARAM, 676
  - USHORT, 677
  - WORD, 677
- swiloc.h, 677
  - pack\_swiloc\_SwiLocGetAutoStart, 677
  - pack\_swiloc\_SwiLocSetAutoStart, 678
  - unpack\_swiloc\_SwiLocGetAutoStart, 678
  - unpack\_swiloc\_SwiLocSetAutoStart, 679
- swioma.h, 679
  - LIBPACK\_MAX\_SWIOMA\_STR\_LEN, 680
  - pack\_swioma\_SLQSOMADMAAlertCallback, 680
  - pack\_swioma\_SLQSOMADMCancelSession, 681
  - pack\_swioma\_SLQSOMADMGetSessionInfo, 681
  - pack\_swioma\_SLQSOMADMGetSettings, 682
  - pack\_swioma\_SLQSOMADMSendSelection, 683
  - pack\_swioma\_SLQSOMADMSetSettings, 683
  - pack\_swioma\_SLQSOMADMStartSession, 684
  - unpack\_swioma\_SLQSOMADMAAlertCallback, 685
  - unpack\_swioma\_SLQSOMADMAAlertCallback\_ind, 685
  - unpack\_swioma\_SLQSOMADMCancelSession, 686
  - unpack\_swioma\_SLQSOMADMGetSessionInfo, 686
  - unpack\_swioma\_SLQSOMADMGetSettings, 687
  - unpack\_swioma\_SLQSOMADMSendSelection, 687
  - unpack\_swioma\_SLQSOMADMSetSettings, 688
  - unpack\_swioma\_SLQSOMADMStartSession, 688
- sysInfoCDMA
  - nas\_CDMA SysInfo, 96
- sysInfoGSM
  - nas\_GSMSysInfo, 113
- sysInfoHDR
  - nas\_HDRSysInfo, 118
- sysInfoLTE
  - nas\_LTESysInfo, 134
- sysInfoWCDMA
  - nas\_WCDMA SysInfo, 176
- system
  - loc\_SV, 84
  - loc\_satelliteInfo, 82
- SystemID
  - unpack\_nas\_SLQSGetServingSystem\_t, 379
- systemID
  - nas\_CDMA SysInfo, 96
- systemMode
  - nas\_CommInfo, 99
- szCarrier\_str
  - \_libSDP\_FirmwareInfo\_, 24
- 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
- t3396\_apn
  - unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 392
- t3396\_plmn\_id
  - unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 392
- t3396\_val
  - unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 392

- TCPDstPort
  - unpack\_qos\_swiQosFilter\_t, 417
- TCPSrcPort
  - unpack\_qos\_swiQosFilter\_t, 417
- TDSCDMAECIOThreshListLen
  - nas\_TDSCDMAECIOThresh, 160
- TDSCDMARSCPThreshListLen
  - nas\_TDSCDMARSCPThresh, 161
- TDSCDMARSSIThreshListLen
  - nas\_TDSCDMARSSIThresh, 161
- TDSCDMASINRThreshListLen
  - nas\_TDSCDMASINRThresh, 162
- TRMessageTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 426
- TXChan
  - nas\_LTEInfo, 122
- tXDroppedCount
  - unpack\_wds\_GetPacketStatus\_t, 452
- tXOKBytesLastCall
  - unpack\_wds\_GetPacketStatus\_t, 453
- tXOkBytesCount
  - unpack\_wds\_GetPacketStatus\_t, 452
- tXPacketErrors
  - unpack\_wds\_GetPacketStatus\_t, 453
- tXPacketOverflows
  - unpack\_wds\_GetPacketStatus\_t, 453
- tXPacketSuccesses
  - unpack\_wds\_GetPacketStatus\_t, 453
- Tables, 21
- tac
  - nas\_LTEInfoIntraFreq, 125
  - nas\_LTESysInfo, 134
- tacValid
  - nas\_LTESysInfo, 134
- TdsBandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, 329
- tdscdmaSigInfoExt, 290
  - ecio, 290
  - rscp, 290
  - rsi, 290
  - sinr, 290
- techName
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback←\_t, 464
- Technology
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- TechnologyPref
  - pack\_nas\_SetNetworkPreference\_t, 220
- tempData\_t, 290
  - temperature, 291
  - temperatureDataLen, 291
  - timeOfFirstSample, 291
  - timeOffset, 291
  - timeSource, 291
- temperature
  - nas\_CommInfo, 99
  - tempData\_t, 291
- temperatureDataLen
  - tempData\_t, 291
- threshGsmHigh
  - nas\_lteGsmCellInfo, 120
- threshGsmLow
  - nas\_lteGsmCellInfo, 120
- threshServingLow
  - nas\_LTEInfoIntraFreq, 125
- threshXHigh
  - nas\_infoInterFreq, 119
- threshXLow
  - nas\_infoInterFreq, 119
- threshXhigh
  - nas\_lteWcdmaCellInfo, 135
- threshXlow
  - nas\_lteWcdmaCellInfo, 135
- Time
  - unpack\_swioma\_SLQSOMADMGetSessionInfo←\_t, 434
- time
  - NASTimeInfoTlv, 188
- TimeLength
  - unpack\_swioma\_SLQSOMADMGetSessionInfo←\_t, 434
- timeMsec
  - pack\_loc\_SLQSLOCInjectUTCTime\_t, 215
- timeOfFirstSample
  - sensorData\_t, 281
  - tempData\_t, 291
- timeOffset
  - sensorData\_t, 281
  - tempData\_t, 291
- timeSource
  - tempData\_t, 291
- timeTlv
  - NASQmiCbkNasSwiOTAMessageInd, 184
- timeUncMsec
  - pack\_loc\_SLQSLOCInjectUTCTime\_t, 215
- timeZone
  - nas\_timeInfo, 164
- timeout
  - pack\_qmi\_t, 237
- timestamp
  - unpack\_dms\_GetNetworkTime\_t, 318
- timestampAge
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- timestampUtc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- timingAdvance
  - nas\_GERANInfo, 107
- TlvPresent
  - dms\_ActivationStatusTlv, 33
  - dms\_OperatingModeTlv, 34
  - eTWSPLMNInfoTlv, 38
  - messageModeTlv, 86
  - NASBandPreferenceTlv, 176
  - NASEmergencyModeTlv, 177
  - NASGWAcqOrderPrefTlv, 177
  - NASLTEBandPreferenceTlv, 178

- NASLteNasReleaseInfoTlv, 178
- NASModePreferenceTlv, 178
- NASNetSelPreferenceTlv, 179
- NASOTAMessageTlv, 179
- NASPRLPreferenceTlv, 184
- NASPhyCaAggPcellInfo, 180
- NASPhyCaAggScellIDBw, 181
- NASPhyCaAggScellIndType, 182
- NASPhyCaAggScellIndex, 181
- NASPhyCaAggScellInfo, 183
- NASRoamPreferenceTlv, 185
- NASServDomainPrefTlv, 186
- NASTimeInfoTlv, 188
- nas\_PhyCaAggPcellInfo, 139
- nas\_PhyCaAggScellIDBw, 139
- nas\_PhyCaAggScellIndType, 140
- nas\_PhyCaAggScellIndex, 140
- nas\_PhyCaAggScellInfo, 143
- nas\_RFInfoTlv, 147
- nas\_RejectReasonTlv, 146
- nas\_SLQSSignalStrengthsTlv, 156
- nas\_SccRxInfo, 152
- nas\_SignalStrengthTlv, 154
- nas\_timeInfo, 164
- newMTMessageTlv, 188
- sMSCAddressTlv, 285
- sMSEtwsMessageTlv, 286
- sMSOnIMSTlv, 289
- transferRouteMessageTlv, 292
- wds\_DHCPLeaseOptTlv, 472
- wds\_DHCPLeaseStateTlv, 472
- wds\_DHCPProfileIdTlv, 473
- wds\_DataULongLongTlv, 471
- wds\_DataULongTlv, 472
- wds\_IPv4AdTlv, 478
- Tlvresult
  - pack\_dms\_GetCustFeaturesV2\_t, 189
  - pack\_dms\_SetCustFeaturesV2\_t, 192
  - pack\_dms\_SetPower\_t, 193
  - pack\_dms\_SetUSBComp\_t, 193
  - pack\_dms\_UIMGetICCID\_t, 197
  - pack\_fms\_GetImagesPreference\_t, 201
  - pack\_fms\_GetStoredImages\_t, 202
  - pack\_fms\_SetImagesPreference\_t, 203
  - pack\_loc\_Delete\_Assist\_Data\_t, 204
  - pack\_loc\_EventRegister\_t, 206
  - pack\_loc\_SLQSLOCGetBestAvailPos\_t, 208
  - pack\_loc\_SetExtPowerState\_t, 206
  - pack\_loc\_SetOperationMode\_t, 207
  - pack\_loc\_Start\_t, 218
  - pack\_loc\_Stop\_t, 218
  - pack\_nas\_SetNetworkPreference\_t, 220
  - pack\_uim\_ChangePin\_t, 248
  - pack\_uim\_ReadTransparent\_t, 249
  - pack\_uim\_SetPinProtection\_t, 250
  - pack\_uim\_UnblockPin\_t, 254
  - pack\_uim\_VerifyPin\_t, 255
  - unpack\_dms\_GetBandCapability\_t, 309
  - unpack\_dms\_GetCrashAction\_t, 309
  - unpack\_dms\_GetCustFeature\_t, 310
  - unpack\_dms\_GetCustFeaturesV2\_t, 310
  - unpack\_dms\_GetDeviceCap\_t, 311
  - unpack\_dms\_GetDeviceHardwareRev\_t, 312
  - unpack\_dms\_GetDeviceMfr\_t, 313
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 313
  - unpack\_dms\_GetFSN\_t, 316
  - unpack\_dms\_GetFirmwareInfo\_t, 314
  - unpack\_dms\_GetFirmwareRevision\_t, 315
  - unpack\_dms\_GetFirmwareRevisions\_t, 315
  - unpack\_dms\_GetIMSI\_t, 316
  - unpack\_dms\_GetManufacturer\_t, 317
  - unpack\_dms\_GetModelID\_t, 317
  - unpack\_dms\_GetNetworkTime\_t, 318
  - unpack\_dms\_GetOfflineReason\_t, 319
  - unpack\_dms\_GetPRLVersion\_t, 320
  - unpack\_dms\_GetPower\_t, 320
  - unpack\_dms\_GetUSBComp\_t, 321
  - unpack\_dms\_GetVoiceNumber\_t, 321
  - unpack\_dms\_ResetToFactoryDefaults\_t, 322
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, 325
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 326
  - unpack\_dms\_SLQSDmsSwiIndicationRegister\_t, 326
  - unpack\_dms\_SLQSGetERIFile\_t, 329
  - unpack\_dms\_SLQSSwiClearDyingGaspStatistics←\_t, 330
  - unpack\_dms\_SLQSSwiGetCrashInfo\_t, 330
  - unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, 331
  - unpack\_dms\_SLQSSwiGetDyingGaspStatistics←\_t, 331
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 334
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 335
  - unpack\_dms\_SLQSSwiGetOSInfo\_t, 335
  - unpack\_dms\_SLQSSwiGetSerialNoExt\_t, 336
  - unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t, 336
  - unpack\_dms\_SLQSSwiSetHostDevInfo\_t, 337
  - unpack\_dms\_SLQSSwiSetOSInfo\_t, 337
  - unpack\_dms\_SLQSUIMGetState\_t, 338
  - unpack\_dms\_SetActivationStatusCallback\_t, 322
  - unpack\_dms\_SetCustFeature\_t, 323
  - unpack\_dms\_SetCustFeaturesV2\_t, 323
  - unpack\_dms\_SetEventReport\_ind\_t, 324
  - unpack\_dms\_SetEventReport\_t, 324
  - unpack\_dms\_SetFirmwarePreference\_t, 324
  - unpack\_dms\_SetPower\_t, 325
  - unpack\_dms\_SetUSBComp\_t, 325
  - unpack\_dms\_UIMGetControlKeyStatus\_t, 339
  - unpack\_dms\_UIMGetICCID\_t, 339
  - unpack\_dms\_UIMGetPINStatus\_t, 341
  - unpack\_dms\_UIMSetControlKeyProtection\_t, 342
  - unpack\_dms\_UIMSetPINProtection\_t, 342
  - unpack\_dms\_UIMUnblockControlKey\_t, 343
  - unpack\_fms\_GetImagesPreference\_t, 344
  - unpack\_fms\_GetStoredImages\_t, 344
  - unpack\_fms\_SetImagesPreference\_t, 345

- unpack\_loc\_BestAvailPos\_Ind\_t, 351
- unpack\_loc\_Delete\_Assist\_Data\_t, 351
- unpack\_loc\_DeleteAssistData\_Ind\_t, 352
- unpack\_loc\_EngineState\_Ind\_t, 353
- unpack\_loc\_EventRegister\_t, 353
- unpack\_loc\_GnssSvInfo\_Ind\_t, 354
- unpack\_loc\_PositionRpt\_Ind\_t, 360
- unpack\_loc\_SLQSLOCGetBestAvailPos\_t, 363
- unpack\_loc\_SetExtPowerConfig\_Ind\_t, 361
- unpack\_loc\_SetExtPowerState\_t, 362
- unpack\_loc\_SetOperationMode\_Ind\_t, 362
- unpack\_loc\_SetOperationMode\_t, 363
- unpack\_loc\_Start\_t, 364
- unpack\_loc\_Stop\_t, 364
- unpack\_nas\_GetNetworkPreference\_t, 367
- unpack\_nas\_SLQSNasSwiOTAMessageCallback↔\_ind\_t, 391
- unpack\_nas\_SLQSSetSysSelectionPrefCall↔Back\_ind\_t, 393
- unpack\_nas\_SetNetworkPreference\_t, 373
- unpack\_nas\_SetServingSystemCallback\_ind\_t, 374
- unpack\_nas\_SlqsGetLTECphyCAInfo\_t, 375
- unpack\_uim\_ChangePin\_t, 437
- unpack\_uim\_GetCardStatus\_t, 437
- unpack\_uim\_ReadTransparent\_t, 438
- unpack\_uim\_SetPinProtection\_t, 439
- unpack\_uim\_UnblockPin\_t, 442
- unpack\_uim\_VerifyPin\_t, 443
- unpack\_wds\_SLQSCreateProfile\_t, 454
- unpack\_wds\_SLQSGetProfileSettings\_t, 460
- unpack\_wds\_SLQSSetIPFamilyPreference\_t, 463
- TokenBucket
  - unpack\_qos\_swiQosFlow\_t, 421
- tokenRate
  - unpack\_qos\_tokenBucket\_t, 422
- tosMask
  - LibPackTFTIDParams, 70
- total\_rx\_bytes
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- total\_rx\_pkt
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- total\_tx\_bytes
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- total\_tx\_bytes\_drp
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- total\_tx\_pkt
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- total\_tx\_pkt\_drp
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 410
- TrackAreaCode
  - unpack\_nas\_SLQSGetServingSystem\_t, 379
- TrafficClass
  - unpack\_qos\_swiQosFlow\_t, 421
- trafficClass
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- trafficPriority
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- TranDstPort
  - unpack\_qos\_swiQosFilter\_t, 417
- TranSrcPort
  - unpack\_qos\_swiQosFilter\_t, 417
- transactionID
  - sMSTransferRouteMTMessage, 289
- transferDelay
  - LibPackUMTSQoS, 72
  - wds\_UMTSMInQoS, 485
- TransferRouteMTMessageInfo
  - transferRouteMessageTlv, 292
- transferRouteMessageTlv, 291
  - TlvPresent, 292
  - TransferRouteMTMessageInfo, 292
- transferStatInd, 292
  - StatsMask, 292
  - StatsPeriod, 292
- transferStats
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 271
- trueSrvStatus
  - nas\_GSMSrvStatusInfo, 110
- tx\_bytes
  - unpack\_QosFlowStat\_t, 423
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- tx\_bytes\_drp
  - unpack\_QosFlowStat\_t, 423
- tx\_pkt
  - unpack\_QosFlowStat\_t, 423
- tx\_pkt\_drp
  - unpack\_QosFlowStat\_t, 423
- tx\_pkts
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- TxDropConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, 424
- txOKBytesCount
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 459
- TxOkByteCountTlv
  - unpack\_RMTransferStatistics\_ind\_t, 424
- TxOkConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, 424
- TxQFilter
  - unpack\_qos\_QosFlowInfo\_t, 406
- TxQFlowGranted
  - unpack\_qos\_QosFlowInfo\_t, 406
- type
  - pack\_wds\_GetDefaultProfileNum\_t, 256
  - pack\_wds\_SetDefaultProfileNum\_t, 260
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, 325
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 326
  - unpack\_qmi\_t, 401
- u16PRLVersion
  - unpack\_dms\_GetPRLVersion\_t, 320
- u8PRLPreference

- unpack\_dms\_GetPRLVersion\_t, 320
- UDPDstPort
  - unpack\_qos\_swiQosFilter\_t, 417
- UDPSrcPort
  - unpack\_qos\_swiQosFilter\_t, 417
- UIM\_MAX\_DESCRIPTION\_LENGTH
  - uim.h, 691
- UIM\_MAX\_NO\_OF\_APPLICATIONS
  - uim.h, 691
- UIM\_MAX\_NO\_OF\_SLOTS
  - uim.h, 691
- UIM\_UINT8\_MAX\_STRING\_SZ
  - uim.h, 691
- ULONGLONG
  - SwiDataTypes.h, 677
- ULONG
  - SwiDataTypes.h, 677
- UMTSGrantedQoS
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- UMTSInstInfo
  - nas\_UMTSInfo, 166
- UMTSLTENbrCell
  - nas\_WCDMAInfoLTENeighborCell, 172
- UMTSReqQoS
  - LibPackUMTSReqQoSsigInd, 73
- UNIQUE\_ID\_LEN
  - dms.h, 502
- UNUSEDPARAM
  - common.h, 491
  - SwiDataTypes.h, 676
- uPhyCardStatus
  - slot\_t, 282
- uPhySlotStatus
  - slot\_t, 282
- USBComp
  - pack\_dms\_SetUSBComp\_t, 193
  - unpack\_dms\_GetUSBComp\_t, 321
- USHORT
  - SwiDataTypes.h, 677
- uarfcn
  - nas\_UMTSInfo, 166
  - nas\_lteWcdmaCellInfo, 135
- ueInIdle
  - nas\_LTEInfoInterfreq, 123
  - nas\_LTEInfoIntrafreq, 125
  - nas\_LTEInfoNeighboringGSM, 126
  - nas\_LTEInfoNeighboringWCDMA, 127
- uim.h, 689
  - MAX\_DESCRIPTION\_LENGTH, 690
  - MAX\_ICCID\_LENGTH, 690
  - MAX\_NO\_OF\_APPLICATIONS, 690
  - MAX\_NO\_OF\_SLOTS, 690
  - MAX\_SLOTS\_STATUS, 690
  - pack\_uim\_ChangePin, 691
  - pack\_uim\_GetCardStatus, 691
  - pack\_uim\_ReadTransparent, 692
  - pack\_uim\_SLQSUIEventRegister, 692
  - pack\_uim\_SLQSUIGetSlotsStatus, 693
  - pack\_uim\_SLQSUIPowerDown, 693
  - pack\_uim\_SLQSUIPowerUp, 693
  - pack\_uim\_SLQSUISwitchSlot, 694
  - pack\_uim\_SetPinProtection, 692
  - pack\_uim\_UnblockPin, 694
  - pack\_uim\_VerifyPin, 695
  - UIM\_MAX\_DESCRIPTION\_LENGTH, 691
  - UIM\_MAX\_NO\_OF\_APPLICATIONS, 691
  - UIM\_MAX\_NO\_OF\_SLOTS, 691
  - UIM\_UINT8\_MAX\_STRING\_SZ, 691
  - unpack\_uim\_ChangePin, 695
  - unpack\_uim\_GetCardStatus, 695
  - unpack\_uim\_ReadTransparent, 696
  - unpack\_uim\_SLQSUIEventRegister, 697
  - unpack\_uim\_SLQSUIGetSlotsStatus, 697
  - unpack\_uim\_SLQSUIPowerDown, 698
  - unpack\_uim\_SLQSUIPowerUp, 698
  - unpack\_uim\_SLQSUISetStatusChangeCall↵
    - Back\_ind, 699
  - unpack\_uim\_SLQSUISwitchSlot, 699
  - unpack\_uim\_SetPinProtection, 696
  - unpack\_uim\_SetUimSlotStatusChangeCallback↵
    - \_ind, 697
  - unpack\_uim\_UnblockPin, 699
  - unpack\_uim\_VerifyPin, 700
- uim\_UIMSessionInformation, 305
  - aid, 306
  - aidLength, 306
  - sessionType, 306
- uim\_appStatus, 292
  - aidLength, 295
  - aidVal, 295
  - appState, 295
  - appType, 295
  - persoFeature, 295
  - persoRetries, 295
  - persoState, 295
  - persoUnblockRetries, 295
  - pin1Retries, 295
  - pin1State, 295
  - pin2Retries, 295
  - pin2State, 295
  - puk1Retries, 295
  - puk2Retries, 295
  - univPin, 295
- uim\_cardResult, 295
  - sw1, 296
  - sw2, 296
- uim\_cardStatus, 296
  - index1xPri, 297
  - index1xSec, 297
  - indexGwPri, 297
  - indexGwSec, 297
  - numSlot, 297
  - SlotInfo, 297
- uim\_changeUIMPIN, 297
  - oldPINLen, 298
  - oldPINVal, 298

- pinID, 298
- pinLen, 298
- pinVal, 298
- uim\_encryptedPIN1, 298
  - pin1Len, 299
  - pin1Val, 299
- uim\_fileInfo, 299
  - fileID, 299
  - path, 299
  - pathLen, 300
- uim\_hotSwapStatus, 300
  - hotSwap, 300
  - hotSwapLength, 300
- uim\_readResult, 300
  - content, 301
  - contentLen, 301
- uim\_readTransparentInfo, 301
  - length, 301
  - offset, 301
- uim\_remainingRetries, 301
  - unblockLeft, 302
  - verifyLeft, 302
- uim\_sessionInformation, 302
  - aid, 303
  - aidLength, 303
  - sessionType, 303
- uim\_setPINProtection, 303
  - pinID, 304
  - pinLength, 304
  - pinOperation, 304
  - pinValue, 304
- uim\_slotInfo, 304
  - AppStatus, 305
  - cardState, 305
  - errorState, 305
  - numApp, 305
  - upinRetries, 305
  - upinState, 305
  - upukRetries, 305
- uim\_unblockUIMPIN, 306
  - newPINLen, 307
  - newPINVal, 307
  - pinID, 307
  - pukLen, 307
  - pukVal, 307
- uim\_verifyUIMPIN, 307
  - pinID, 308
  - pinLen, 308
  - pinVal, 308
- uimSlotStatus
  - slots\_t, 284
- ulData
  - wds\_DataULongLongTlv, 472
- ulMask
  - rmTrasnferStaticsReq, 280
- ulPhysicalSlot
  - pack\_uim\_SLQSUIMSwitchSlot\_t, 253
- ulldata
  - wds\_DataULongLongTlv, 471
- umtsEcio
  - nas\_UMTSInstInfo, 167
- umtsInst
  - nas\_UMTSInfo, 166
- umtsLTENbrCellLen
  - nas\_WCDMAInfoLTENeighborCell, 172
- umtsPsc
  - nas\_UMTSInstInfo, 167
- umtsRscp
  - nas\_UMTSInstInfo, 167
- umtsUarfcn
  - nas\_UMTSInstInfo, 167
- UnPackGetProfileSettingOut, 469
  - curProfile, 469
  - pExtErrCode, 469
- unblockLeft
  - uim\_remainingRetries, 302
- unblockRetriesLeft
  - unpack\_dms\_UIMGetControlKeyStatus\_t, 339
  - unpack\_dms\_UIMSetPINProtection\_t, 342
  - unpack\_dms\_UIMUnblockControlKey\_t, 343
- uniqueID
  - image\_info\_t, 43
- univPin
  - appStats, 28
  - uim\_appStatus, 295
- universalTime
  - unpack\_nas\_SLQSNasNetworkTimeCallBack\_↔  
ind\_t, 389
- unpack\_QosFlowStat\_t, 422
  - bearerId, 423
  - tx\_bytes, 423
  - tx\_bytes\_drp, 423
  - tx\_pkt, 423
  - tx\_pkt\_drp, 423
- unpack\_RMTransferStatistics\_ind\_t, 423
  - RxDropConutTlv, 424
  - RxOkByteCountTlv, 424
  - RxOkConutTlv, 424
  - TxDropConutTlv, 424
  - TxOkByteCountTlv, 424
  - TxOkConutTlv, 424
- unpack\_dms\_ActivateAutomatic
  - dms.h, 527
- unpack\_dms\_GetActivationState
  - dms.h, 527
- unpack\_dms\_GetActivationState\_t, 308
  - state, 309
- unpack\_dms\_GetBandCapability
  - dms.h, 528
- unpack\_dms\_GetBandCapability\_t, 309
  - BandCapability, 309
  - Tlvresult, 309
- unpack\_dms\_GetCrashAction
  - dms.h, 528
- unpack\_dms\_GetCrashAction\_t, 309
  - DevCrashState, 309



- Tlvresult, [309](#)
- unpack\_dms\_GetCustFeature
  - dms.h, [528](#)
- unpack\_dms\_GetCustFeature\_t, [309](#)
  - DHCPRelayEnabled, [310](#)
  - DisableIMSI, [310](#)
  - GPSPMP, [310](#)
  - GPSSel, [310](#)
  - GpsEnable, [310](#)
  - IPFamSupport, [310](#)
  - IsVoiceEnabled, [310](#)
  - RMAutoConnect, [310](#)
  - SMSSupport, [310](#)
  - Tlvresult, [310](#)
- unpack\_dms\_GetCustFeaturesV2
  - dms.h, [529](#)
- unpack\_dms\_GetCustFeaturesV2\_t, [310](#)
  - GetCustomFeatureV2, [310](#)
  - Tlvresult, [310](#)
- unpack\_dms\_GetDeviceCap
  - dms.h, [529](#)
- unpack\_dms\_GetDeviceCap\_t, [311](#)
  - DataServiceCapability, [311](#)
  - MaxRXChannelRate, [311](#)
  - MaxTXChannelRate, [311](#)
  - Radiolfaces, [311](#)
  - RadiolfacesSize, [311](#)
  - SimCapability, [311](#)
  - Tlvresult, [311](#)
- unpack\_dms\_GetDeviceCapabilities
  - dms.h, [529](#)
- unpack\_dms\_GetDeviceCapabilities\_t, [311](#)
  - dataServiceCaCapability, [312](#)
  - maxRxChannelRate, [312](#)
  - maxTxChannelRate, [312](#)
  - Radiolfaces, [312](#)
  - radiolfacesSize, [312](#)
  - simCapability, [312](#)
- unpack\_dms\_GetDeviceHardwareRev
  - dms.h, [530](#)
- unpack\_dms\_GetDeviceHardwareRev\_t, [312](#)
  - String, [312](#)
  - stringSize, [312](#)
  - Tlvresult, [312](#)
- unpack\_dms\_GetDeviceMfr
  - dms.h, [530](#)
- unpack\_dms\_GetDeviceMfr\_t, [312](#)
  - String, [313](#)
  - stringSize, [313](#)
  - Tlvresult, [313](#)
- unpack\_dms\_GetDeviceSerialNumbers
  - dms.h, [531](#)
- unpack\_dms\_GetDeviceSerialNumbers\_t, [313](#)
  - ESNString, [313](#)
  - esnSize, [313](#)
  - IMEIString, [313](#)
  - imeiSize, [313](#)
  - imeiSvnSize, [313](#)
  - ImeiSvnString, [313](#)
  - MEIDString, [313](#)
  - meidSize, [313](#)
  - Tlvresult, [313](#)
- unpack\_dms\_GetFSN\_t, [316](#)
  - String, [316](#)
  - Tlvresult, [316](#)
- unpack\_dms\_GetFSN
  - dms.h, [532](#)
- unpack\_dms\_GetFirmwareInfo
  - dms.h, [531](#)
- unpack\_dms\_GetFirmwareInfo\_t, [313](#)
  - appversion\_str, [314](#)
  - bootversion\_str, [314](#)
  - carrier\_str, [314](#)
  - cur\_carr\_name, [314](#)
  - cur\_carr\_rev, [314](#)
  - modelid\_str, [314](#)
  - packageid\_str, [314](#)
  - priversion\_str, [314](#)
  - sku\_str, [314](#)
  - Tlvresult, [314](#)
- unpack\_dms\_GetFirmwareRevision
  - dms.h, [531](#)
- unpack\_dms\_GetFirmwareRevision\_t, [314](#)
  - AMSSString, [315](#)
  - amssSize, [315](#)
  - PRIString, [315](#)
  - Tlvresult, [315](#)
- unpack\_dms\_GetFirmwareRevisions
  - dms.h, [532](#)
- unpack\_dms\_GetFirmwareRevisions\_t, [315](#)
  - AMSSString, [315](#)
  - amssSize, [315](#)
  - bootSize, [315](#)
  - BootString, [315](#)
  - PRIString, [315](#)
  - priSize, [315](#)
  - Tlvresult, [315](#)
- unpack\_dms\_GetHardwareRevision
  - dms.h, [533](#)
- unpack\_dms\_GetHardwareRevision\_t, [316](#)
  - hwVer, [316](#)
- unpack\_dms\_GetIMSI\_t, [316](#)
  - imsi, [316](#)
  - Tlvresult, [316](#)
- unpack\_dms\_GetIMSI
  - dms.h, [533](#)
- unpack\_dms\_GetManufacturer
  - dms.h, [533](#)
- unpack\_dms\_GetManufacturer\_t, [316](#)
  - manufacturer, [317](#)
  - Tlvresult, [317](#)
- unpack\_dms\_GetModelID\_t, [317](#)
  - modelid, [317](#)
  - Tlvresult, [317](#)
- unpack\_dms\_GetModelID
  - dms.h, [534](#)

- unpack\_dms\_GetNetworkTime
  - dms.h, [534](#)
- unpack\_dms\_GetNetworkTime\_t, [317](#)
  - source, [318](#)
  - timestamp, [318](#)
  - Tlvresult, [318](#)
- unpack\_dms\_GetOfflineReason
  - dms.h, [535](#)
- unpack\_dms\_GetOfflineReason\_t, [318](#)
  - pReasonMask, [319](#)
  - pbPlatform, [319](#)
  - Tlvresult, [319](#)
- unpack\_dms\_GetPRLVersion
  - dms.h, [535](#)
- unpack\_dms\_GetPRLVersion\_t, [320](#)
  - Tlvresult, [320](#)
  - u16PRLVersion, [320](#)
  - u8PRLPreference, [320](#)
- unpack\_dms\_GetPower
  - dms.h, [535](#)
- unpack\_dms\_GetPower\_t, [319](#)
  - HardwareControlledMode, [319](#)
  - OfflineReason, [320](#)
  - OperationMode, [320](#)
  - Tlvresult, [320](#)
- unpack\_dms\_GetSerialNumbers
  - dms.h, [536](#)
- unpack\_dms\_GetSerialNumbers\_t, [320](#)
  - esn, [320](#)
  - imei\_no, [320](#)
  - imeisv\_svn, [321](#)
  - meid, [321](#)
- unpack\_dms\_GetUSBComp
  - dms.h, [536](#)
- unpack\_dms\_GetUSBComp\_t, [321](#)
  - NumSupUSBComps, [321](#)
  - SupUSBComps, [321](#)
  - Tlvresult, [321](#)
  - USBComp, [321](#)
- unpack\_dms\_GetVoiceNumber
  - dms.h, [537](#)
- unpack\_dms\_GetVoiceNumber\_t, [321](#)
  - MIN, [321](#)
  - minSize, [321](#)
  - Tlvresult, [321](#)
  - VoiceNumber, [321](#)
  - voiceNumberSize, [321](#)
- unpack\_dms\_ResetToFactoryDefaults
  - dms.h, [537](#)
- unpack\_dms\_ResetToFactoryDefaults\_t, [321](#)
  - Tlvresult, [322](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo
  - dms.h, [541](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind
  - dms.h, [541](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, [325](#)
  - source, [325](#)
  - Tlvresult, [325](#)
  - type, [325](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, [325](#)
  - source, [326](#)
  - Tlvresult, [326](#)
  - type, [326](#)
- unpack\_dms\_SLQSDmsSwiIndicationRegister
  - dms.h, [542](#)
- unpack\_dms\_SLQSDmsSwiIndicationRegister\_t, [326](#)
  - Tlvresult, [326](#)
- unpack\_dms\_SLQSGetBandCapability
  - dms.h, [542](#)
- unpack\_dms\_SLQSGetBandCapability\_t, [326](#)
  - bandCapability, [329](#)
  - is\_LteBandCapability\_Available, [329](#)
  - is\_TdsBandCapability\_Available, [329](#)
  - LteBandCapability, [329](#)
  - TdsBandCapability, [329](#)
- unpack\_dms\_SLQSGetERIFile
  - dms.h, [543](#)
- unpack\_dms\_SLQSGetERIFile\_t, [329](#)
  - eriFile, [329](#)
  - Tlvresult, [329](#)
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics
  - dms.h, [543](#)
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_↔
  - t, [330](#)
  - Tlvresult, [330](#)
- unpack\_dms\_SLQSSwiGetCrashInfo
  - dms.h, [544](#)
- unpack\_dms\_SLQSSwiGetCrashInfo\_t, [330](#)
  - crashInfoParam, [330](#)
  - Tlvresult, [330](#)
- unpack\_dms\_SLQSSwiGetDyingGaspCfg
  - dms.h, [544](#)
- unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, [331](#)
  - pGetDyingGaspCfg, [331](#)
  - Tlvresult, [331](#)
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics
  - dms.h, [544](#)
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t, [331](#)
  - pGetDyingGaspStatistics, [331](#)
  - Tlvresult, [331](#)
- unpack\_dms\_SLQSSwiGetFirmwareCurr
  - dms.h, [545](#)
- unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [332](#)
  - carrier, [332](#)
  - fwvers, [332](#)
  - numEntries, [332](#)
  - pCurrImgInfo, [332](#)
  - pkgver, [332](#)
  - priver, [332](#)
- unpack\_dms\_SLQSSwiGetFwUpdateStatus
  - dms.h, [545](#)
- unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [333](#)
  - imgType, [334](#)
  - logString, [334](#)
  - refData, [334](#)
  - refString, [334](#)



- ResCode, [334](#)
- Tlvresult, [334](#)
- unpack\_dms\_SLQSSwiGetHostDevInfo
  - dms.h, [546](#)
- unpack\_dms\_SLQSSwiGetHostDevInfo\_t, [334](#)
  - manString, [335](#)
  - modelString, [335](#)
  - plasmaIDString, [335](#)
  - swVerString, [335](#)
  - Tlvresult, [335](#)
- unpack\_dms\_SLQSSwiGetOSInfo
  - dms.h, [546](#)
- unpack\_dms\_SLQSSwiGetOSInfo\_t, [335](#)
  - nameString, [335](#)
  - Tlvresult, [335](#)
  - versionString, [335](#)
- unpack\_dms\_SLQSSwiGetSerialNoExt
  - dms.h, [546](#)
- unpack\_dms\_SLQSSwiGetSerialNoExt\_t, [335](#)
  - meidString, [336](#)
  - Tlvresult, [336](#)
- unpack\_dms\_SLQSSwiSetDyingGaspCfg
  - dms.h, [547](#)
- unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [336](#)
  - Tlvresult, [336](#)
- unpack\_dms\_SLQSSwiSetHostDevInfo
  - dms.h, [547](#)
- unpack\_dms\_SLQSSwiSetHostDevInfo\_t, [336](#)
  - Tlvresult, [337](#)
- unpack\_dms\_SLQSSwiSetOSInfo
  - dms.h, [548](#)
- unpack\_dms\_SLQSSwiSetOSInfo\_t, [337](#)
  - Tlvresult, [337](#)
- unpack\_dms\_SLQSUIMGetState
  - dms.h, [548](#)
- unpack\_dms\_SLQSUIMGetState\_t, [337](#)
  - state, [338](#)
  - Tlvresult, [338](#)
- unpack\_dms\_SetActivationStatusCallback
  - dms.h, [537](#)
- unpack\_dms\_SetActivationStatusCallback\_t, [322](#)
  - Tlvresult, [322](#)
- unpack\_dms\_SetCrashAction
  - dms.h, [538](#)
- unpack\_dms\_SetCrashAction\_t, [322](#)
  - notused, [323](#)
- unpack\_dms\_SetCustFeature
  - dms.h, [538](#)
- unpack\_dms\_SetCustFeature\_t, [323](#)
  - Tlvresult, [323](#)
- unpack\_dms\_SetCustFeaturesV2
  - dms.h, [539](#)
- unpack\_dms\_SetCustFeaturesV2\_t, [323](#)
  - Tlvresult, [323](#)
- unpack\_dms\_SetEventReport
  - dms.h, [539](#)
- unpack\_dms\_SetEventReport\_ind
  - dms.h, [539](#)
- unpack\_dms\_SetEventReport\_ind\_t, [323](#)
  - ActivationStatusTlv, [324](#)
  - OperatingModeTlv, [324](#)
  - Tlvresult, [324](#)
- unpack\_dms\_SetEventReport\_t, [324](#)
  - Tlvresult, [324](#)
- unpack\_dms\_SetFirmwarePreference
  - dms.h, [540](#)
- unpack\_dms\_SetFirmwarePreference\_t, [324](#)
  - Tlvresult, [324](#)
- unpack\_dms\_SetPower
  - dms.h, [540](#)
- unpack\_dms\_SetPower\_t, [324](#)
  - Tlvresult, [325](#)
- unpack\_dms\_SetUSBComp
  - dms.h, [541](#)
- unpack\_dms\_SetUSBComp\_t, [325](#)
  - Tlvresult, [325](#)
- unpack\_dms\_UIMChangePIN
  - dms.h, [548](#)
- unpack\_dms\_UIMGetControlKeyStatus
  - dms.h, [549](#)
- unpack\_dms\_UIMGetControlKeyStatus\_t, [338](#)
  - facilityState, [339](#)
  - Tlvresult, [339](#)
  - unblockRetriesLeft, [339](#)
  - verifyRetriesLeft, [339](#)
- unpack\_dms\_UIMGetICCID\_t, [339](#)
  - String, [339](#)
  - stringSize, [339](#)
  - Tlvresult, [339](#)
- unpack\_dms\_UIMGetICCID
  - dms.h, [549](#)
- unpack\_dms\_UIMGetPINStatus
  - dms.h, [550](#)
- unpack\_dms\_UIMGetPINStatus\_t, [339](#)
  - p1Status, [341](#)
  - p1UnblockRetriesLeft, [341](#)
  - p1VerifyRetriesLeft, [341](#)
  - p2Status, [341](#)
  - p2UnblockRetriesLeft, [341](#)
  - p2VerifyRetriesLeft, [341](#)
  - Tlvresult, [341](#)
- unpack\_dms\_UIMSetControlKeyProtection
  - dms.h, [550](#)
- unpack\_dms\_UIMSetControlKeyProtection\_t, [341](#)
  - Tlvresult, [342](#)
  - verifyRetriesLeft, [342](#)
- unpack\_dms\_UIMSetPINProtection
  - dms.h, [550](#)
- unpack\_dms\_UIMSetPINProtection\_t, [342](#)
  - Tlvresult, [342](#)
  - unblockRetriesLeft, [342](#)
  - verifyRetriesLeft, [342](#)
- unpack\_dms\_UIMUnblockControlKey
  - dms.h, [551](#)
- unpack\_dms\_UIMUnblockControlKey\_t, [342](#)
  - Tlvresult, [343](#)

- unlockRetriesLeft, 343
- unpack\_dms\_UIMUnblockPIN
  - dms.h, 551
- unpack\_dms\_UIMVerifyPIN
  - dms.h, 552
- unpack\_dms\_ValidateSPC
  - dms.h, 552
- unpack\_fms\_GetImagesPreference
  - fms.h, 555
- unpack\_fms\_GetImagesPreference\_t, 343
  - ImageListSize, 343
  - pImageList, 344
  - Tlvresult, 344
- unpack\_fms\_GetStoredImages
  - fms.h, 555
- unpack\_fms\_GetStoredImages\_t, 344
  - imageList, 344
  - imagelistSize, 344
  - Tlvresult, 344
- unpack\_fms\_SetImagesPreference
  - fms.h, 556
- unpack\_fms\_SetImagesPreference\_t, 344
  - ImageTypes, 345
  - ImageTypesSize, 345
  - Tlvresult, 345
- unpack\_loc\_BestAvailPos\_Ind
  - loc.h, 573
- unpack\_loc\_BestAvailPos\_Ind\_t, 345
  - pAltitudeWrtEllipsoid, 349
  - pAltitudeWrtMeanSeaLevel, 349
  - pGpsTime, 350
  - pHeading, 350
  - pHeadingUnc, 350
  - pHorCirConf, 350
  - pHorEllpConf, 350
  - pHorReliability, 350
  - pHorUncCircular, 350
  - pHorUncEllipseOrientAzimuth, 350
  - pHorUncEllipseSemiMajor, 350
  - pHorUncEllipseSemiMinor, 350
  - pLatitude, 350
  - pLongitude, 350
  - pMagneticDeviation, 350
  - pPrecisionDilution, 350
  - pSensorDataUsage, 350
  - pSpeedHorizontal, 350
  - pSpeedUnc, 350
  - pSpeedVertical, 350
  - pSpeedVerticalUnc, 350
  - pSvUsedforFix, 350
  - pTechnologyMask, 350
  - pTimeSrc, 350
  - pTimeUnc, 351
  - pTimestampUtc, 350
  - pVertConfidence, 351
  - pVertReliability, 351
  - pVertUnc, 351
  - pXid, 351
- status, 351
- Tlvresult, 351
- unpack\_loc\_Delete\_Assist\_Data\_t, 351
  - Tlvresult, 351
- unpack\_loc\_DeleteAssistData
  - loc.h, 574
- unpack\_loc\_DeleteAssistData\_Ind
  - loc.h, 574
- unpack\_loc\_DeleteAssistData\_Ind\_t, 351
  - status, 352
  - Tlvresult, 352
- unpack\_loc\_EngineState\_Ind
  - loc.h, 574
- unpack\_loc\_EngineState\_Ind\_t, 352
  - engineState, 353
  - Tlvresult, 353
- unpack\_loc\_EventRegister
  - loc.h, 575
- unpack\_loc\_EventRegister\_t, 353
  - Tlvresult, 353
- unpack\_loc\_GnssSvInfo\_Ind
  - loc.h, 575
- unpack\_loc\_GnssSvInfo\_Ind\_t, 353
  - altitudeAssumed, 354
  - pSatelliteInfo, 354
  - Tlvresult, 354
- unpack\_loc\_PositionRpt\_Ind
  - loc.h, 576
- unpack\_loc\_PositionRpt\_Ind\_t, 354
  - pAltitudeAssumed, 359
  - pAltitudeWrtEllipsoid, 359
  - pAltitudeWrtMeanSeaLevel, 359
  - pFixId, 359
  - pGpsTime, 359
  - pHeading, 359
  - pHeadingUnc, 359
  - pHorConfidence, 359
  - pHorReliability, 359
  - pHorUncCircular, 359
  - pHorUncEllipseOrientAzimuth, 359
  - pHorUncEllipseSemiMajor, 359
  - pHorUncEllipseSemiMinor, 359
  - pLatitude, 359
  - pLeapSeconds, 359
  - pLongitude, 359
  - pMagneticDeviation, 359
  - pPrecisionDilution, 359
  - pSensorDataUsage, 359
  - pSpeedHorizontal, 359
  - pSpeedUnc, 360
  - pSpeedVertical, 360
  - pSvUsedforFix, 360
  - pTechnologyMask, 360
  - pTimeSrc, 360
  - pTimeUnc, 360
  - pTimestampUtc, 360
  - pVertConfidence, 360
  - pVertReliability, 360

- pVertUnc, [360](#)
- sessionId, [360](#)
- sessionStatus, [360](#)
- Tlvresult, [360](#)
- unpack\_loc\_SLQSLOCGetBestAvailPos
  - loc.h, [578](#)
- unpack\_loc\_SLQSLOCGetBestAvailPos\_t, [363](#)
  - Tlvresult, [363](#)
- unpack\_loc\_SLQSLOCInjectPosition
  - loc.h, [578](#)
- unpack\_loc\_SLQSLOCInjectSensorData
  - loc.h, [578](#)
- unpack\_loc\_SLQSLOCInjectUTCTime
  - loc.h, [579](#)
- unpack\_loc\_SLQSLOCSetCradleMountConfig
  - loc.h, [579](#)
- unpack\_loc\_SetExtPowerConfig\_Ind
  - loc.h, [576](#)
- unpack\_loc\_SetExtPowerConfig\_Ind\_t, [360](#)
  - status, [361](#)
  - Tlvresult, [361](#)
- unpack\_loc\_SetExtPowerState
  - loc.h, [576](#)
- unpack\_loc\_SetExtPowerState\_t, [361](#)
  - Tlvresult, [362](#)
- unpack\_loc\_SetOperationMode
  - loc.h, [577](#)
- unpack\_loc\_SetOperationMode\_Ind
  - loc.h, [577](#)
- unpack\_loc\_SetOperationMode\_Ind\_t, [362](#)
  - status, [362](#)
  - Tlvresult, [362](#)
- unpack\_loc\_SetOperationMode\_t, [362](#)
  - Tlvresult, [363](#)
- unpack\_loc\_Start
  - loc.h, [579](#)
- unpack\_loc\_Start\_t, [363](#)
  - Tlvresult, [364](#)
- unpack\_loc\_Stop
  - loc.h, [580](#)
- unpack\_loc\_Stop\_t, [364](#)
  - Tlvresult, [364](#)
- unpack\_nas\_GetACCOLC
  - nas.h, [598](#)
- unpack\_nas\_GetANAAAAAuthenticationStatus
  - nas.h, [598](#)
- unpack\_nas\_GetCDMANetworkParameters
  - nas.h, [599](#)
- unpack\_nas\_GetCDMANetworkParameters\_t, [364](#)
  - Application, [365](#)
  - Broadcast, [365](#)
  - CustomSCP, [365](#)
  - ForceRev0, [365](#)
  - Protocol, [365](#)
  - RegForeignNID, [365](#)
  - RegForeignSID, [365](#)
  - RegHomeSID, [365](#)
  - Roaming, [365](#)
  - SCI, [365](#)
  - SCM, [365](#)
- unpack\_nas\_GetHomeNetwork
  - nas.h, [599](#)
- unpack\_nas\_GetHomeNetwork\_t, [365](#)
  - mcc, [366](#)
  - mnc, [366](#)
  - name, [366](#)
  - nid, [366](#)
  - sid, [366](#)
- unpack\_nas\_GetNetworkPreference
  - nas.h, [599](#)
- unpack\_nas\_GetNetworkPreference\_t, [366](#)
  - ActiveTechPref, [367](#)
  - Duration, [367](#)
  - PersistentTechPref, [367](#)
  - Tlvresult, [367](#)
- unpack\_nas\_GetRFInfo
  - nas.h, [599](#)
- unpack\_nas\_GetRFInfo\_t, [367](#)
  - instancesSize, [367](#)
  - RFBandInfoElements, [367](#)
- unpack\_nas\_GetServingNetwork
  - nas.h, [600](#)
- unpack\_nas\_GetServingNetwork\_t, [368](#)
  - CSDomain, [368](#)
  - DataCaps, [368](#)
  - DataCapsLen, [368](#)
  - MCC, [368](#)
  - MNC, [368](#)
  - Name, [369](#)
  - nameSize, [369](#)
  - PSDomain, [369](#)
  - RAN, [369](#)
  - Radiolfaces, [369](#)
  - RadiolfacesSize, [369](#)
  - RegistrationState, [369](#)
  - Roaming, [369](#)
- unpack\_nas\_GetServingNetworkCapabilities
  - nas.h, [600](#)
- unpack\_nas\_GetServingNetworkCapabilities\_t, [369](#)
  - DataCaps, [369](#)
  - DataCapsLen, [369](#)
- unpack\_nas\_GetSignalStrengths
  - nas.h, [600](#)
- unpack\_nas\_GetSignalStrengths\_t, [369](#)
  - len, [370](#)
  - radio, [370](#)
  - rsi, [370](#)
- unpack\_nas\_PerformNetworkScan
  - nas.h, [601](#)
- unpack\_nas\_PerformNetworkScan\_t, [370](#)
  - p3GppNetworkInfoInstances, [370](#)
  - p3GppNetworkInstanceSize, [370](#)
  - pPcsInstance, [370](#)
  - pPcsInstanceSize, [370](#)
  - pRatInstance, [370](#)
  - pRatInstanceSize, [370](#)

- pScanResult, 371
- unpack\_nas\_SLQSGetNetworkTime
  - nas.h, 604
- unpack\_nas\_SLQSGetNetworkTime\_t, 375
  - p3GPP2TimeInfo, 375
  - p3GPPTimeInfo, 375
- unpack\_nas\_SLQSGetPLMNName
  - nas.h, 604
- unpack\_nas\_SLQSGetPLMNName\_t, 376
  - longName, 376
  - longNameCI, 376
  - longNameEn, 376
  - longNameLen, 376
  - longNameSB, 376
  - shortName, 376
  - shortNameCI, 376
  - shortNameEn, 376
  - shortNameLen, 376
  - shortNameSB, 376
  - spn, 376
  - spnEncoding, 376
  - spnLength, 376
- unpack\_nas\_SLQSGetServingSystem
  - nas.h, 604
- unpack\_nas\_SLQSGetServingSystem\_t, 376
  - BasestationID, 378
  - BasestationLatitude, 378
  - BasestationLongitude, 378
  - CDMA\_P\_Rev, 378
  - CDMASystemInfoExt, 378
  - CallBarStatus, 378
  - CellID, 378
  - ConcSvcInfo, 378
  - CurrentPLMN, 378
  - DTMInd, 378
  - DataSrvCapabilities, 378
  - DefaultRoamInd, 378
  - DetailedSvcInfo, 378
  - Gpp2TimeZone, 378
  - GppNetworkDSTAdjustment, 378
  - GppTimeZone, 378
  - HdrPersonality, 378
  - Lac, 378
  - NetworkID, 378
  - PRLInd, 379
  - RoamIndicatorVal, 379
  - RoamingIndicatorList, 379
  - ServingSystem, 379
  - SystemID, 379
  - TrackAreaCode, 379
- unpack\_nas\_SLQSGetSignalStrength
  - nas.h, 605
- unpack\_nas\_SLQSGetSignalStrength\_t, 379
  - ecioList, 380
  - ecioListLen, 380
  - errorRateList, 380
  - errorRateListLen, 380
  - lo, 380
  - ltsr, 380
  - ltsnr, 380
  - rsrqInfo, 380
  - rxSignalStrengthList, 380
  - rxSignalStrengthListLen, 380
  - signalStrengthReqMask, 380
  - sinr, 380
- unpack\_nas\_SLQSGetSysInfo
  - nas.h, 605
- unpack\_nas\_SLQSGetSysInfo\_t, 380
  - pAddCDMASysInfo, 382
  - pAddGSMsSysInfo, 382
  - pAddHDRSysInfo, 382
  - pAddLTESysInfo, 382
  - pAddWCDMASysInfo, 382
  - pCDMASrvStatusInfo, 382
  - pCDMASysInfo, 382
  - pGSMCallBarringSysInfo, 382
  - pGSMCipherDomainSysInfo, 382
  - pGSMSrvStatusInfo, 382
  - pGSMsSysInfo, 382
  - pHDSrvStatusInfo, 382
  - pHDRSysInfo, 383
  - pLTESrvStatusInfo, 383
  - pLTESysInfo, 383
  - pLTEVoiceSupportSysInfo, 383
  - pWCDMACallBarringSysInfo, 383
  - pWCDMACipherDomainSysInfo, 383
  - pWCDMASrvStatusInfo, 383
  - pWCDMASysInfo, 383
- unpack\_nas\_SLQSGetSysSelectionPref
  - nas.h, 605
- unpack\_nas\_SLQSGetSysSelectionPref\_t, 383
  - pBandPref, 386
  - pEmerMode, 386
  - pGWAqOrderPref, 386
  - pLTEBandPref, 386
  - pModePref, 386
  - pNetSelPref, 386
  - pPRLPref, 386
  - pRoamPref, 386
  - pSrvDomainPref, 386
- unpack\_nas\_SLQSIInitiateNetworkRegistration
  - nas.h, 606
- unpack\_nas\_SLQSNasConfigSigInfo2
  - nas.h, 606
- unpack\_nas\_SLQSNasGetCellLocationInfo
  - nas.h, 606
- unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 386
  - pCDMAInfo, 387
  - pGERANInfo, 387
  - pLTEInfoInterfreq, 387
  - pLTEInfoIntrafreq, 388
  - pLTEInfoNeighboringGSM, 388
  - pLTEInfoNeighboringWCDMA, 388
  - pUMTSCellID, 388
  - pUMTSInfo, 388
  - pWCDMAInfoLTENeighborCell, 388

- unpack\_nas\_SLQSNasGetSigInfo
  - nas.h, [607](#)
- unpack\_nas\_SLQSNasGetSigInfo\_t, [388](#)
  - CDMASSInfo, [388](#)
  - GSMSSInfo, [388](#)
  - HDRSSInfo, [388](#)
  - LTSSInfo, [388](#)
  - WCDMASSInfo, [388](#)
- unpack\_nas\_SLQSNasIndicationRegisterExt
  - nas.h, [607](#)
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind
  - nas.h, [607](#)
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t, [389](#)
  - pDayltSavAdj, [389](#)
  - pRadiolInterface, [389](#)
  - pTimeZone, [389](#)
  - universalTime, [389](#)
- unpack\_nas\_SLQSNasSigInfoCallback\_ind
  - nas.h, [608](#)
- unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [390](#)
  - pCDMASigInfo, [390](#)
  - pGSMSigInfo, [390](#)
  - pHRSigInfo, [390](#)
  - pLTSigInfo, [390](#)
  - pRscp, [390](#)
  - pTDSCDMASigInfoExt, [390](#)
  - pWCDMASigInfo, [390](#)
- unpack\_nas\_SLQSNasSwiIndicationRegister
  - nas.h, [608](#)
- unpack\_nas\_SLQSNasSwiModemStatus
  - nas.h, [608](#)
- unpack\_nas\_SLQSNasSwiModemStatus\_t, [390](#)
  - commonInfo, [391](#)
  - pLTEInfo, [391](#)
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind
  - nas.h, [609](#)
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t, [391](#)
  - Info, [391](#)
  - Tlvresult, [391](#)
- unpack\_nas\_SLQSNasSysInfoCallback\_ind
  - nas.h, [609](#)
- unpack\_nas\_SLQSNasTimerCallback\_ind
  - nas.h, [609](#)
- unpack\_nas\_SLQSNasTimerCallback\_ind\_t, [391](#)
  - t3396\_apn, [392](#)
  - t3396\_plmn\_id, [392](#)
  - t3396\_val, [392](#)
- unpack\_nas\_SLQSSetBandPreference
  - nas.h, [610](#)
- unpack\_nas\_SLQSSetSignalStrengthsCallback
  - nas.h, [610](#)
- unpack\_nas\_SLQSSetSysSelectionPref
  - nas.h, [610](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind
  - nas.h, [611](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t, [392](#)
  - Info, [393](#)
  - Tlvresult, [393](#)
- unpack\_nas\_SLQSSwiGetLteCQI\_t, [393](#)
  - CQIValueCW0, [393](#)
  - CQIValueCW1, [393](#)
  - ValidityCW0, [393](#)
  - ValidityCW1, [393](#)
- unpack\_nas\_SLQSSwiGetLteCQI
  - nas.h, [611](#)
- unpack\_nas\_SLQSSwiGetLteSccRxInfo
  - nas.h, [611](#)
- unpack\_nas\_SLQSSwiGetLteSccRxInfo\_t, [394](#)
  - pSccRxInfo, [394](#)
- unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [394](#)
  - pAddCDMASysInfo, [396](#)
  - pAddGSMSSysInfo, [396](#)
  - pAddHDRSysInfo, [396](#)
  - pAddLTESysInfo, [396](#)
  - pAddWCDMASysInfo, [396](#)
  - pCDMASrvStatusInfo, [396](#)
  - pCDMASysInfo, [396](#)
  - pGSMCallBarringSysInfo, [396](#)
  - pGSMCipherDomainSysInfo, [396](#)
  - pGSMSSrvStatusInfo, [396](#)
  - pGSMSSysInfo, [397](#)
  - pHRSrvStatusInfo, [397](#)
  - pHRSysInfo, [397](#)
  - pLTESrvStatusInfo, [397](#)
  - pLTESysInfo, [397](#)
  - pLTEVoiceSupportSysInfo, [397](#)
  - pSysInfoNoChange, [397](#)
  - pWCDMACallBarringSysInfo, [397](#)
  - pWCDMACipherDomainSysInfo, [397](#)
  - pWCDMASrvStatusInfo, [397](#)
  - pWCDMASysInfo, [397](#)
- unpack\_nas\_SetACCOLC
  - nas.h, [601](#)
- unpack\_nas\_SetDataCapabilitiesCallback\_ind
  - nas.h, [602](#)
- unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, [371](#)
  - dataCaps, [371](#)
  - dataCapsSize, [371](#)
- unpack\_nas\_SetEventReportInd
  - nas.h, [602](#)
- unpack\_nas\_SetEventReportInd\_t, [371](#)
  - RFTlv, [371](#)
  - RRTlv, [371](#)
  - SLQSSSTlv, [372](#)
  - SSTlv, [372](#)
- unpack\_nas\_SetLUREjectCallback
  - nas.h, [602](#)
- unpack\_nas\_SetNasLTECphyCalIndCallback\_ind
  - nas.h, [602](#)
- unpack\_nas\_SetNasLTECphyCalIndCallback\_ind\_t, [372](#)
  - sPhyCaAggPcellInfo, [372](#)
  - sPhyCaAggScellDIBw, [372](#)

- sPhyCaAggScellIndType, 372
- sPhyCaAggScellIndex, 372
- sPhyCaAggScellInfo, 372
- unpack\_nas\_SetNetworkPreference
  - nas.h, 603
- unpack\_nas\_SetNetworkPreference\_t, 373
  - Tlvresult, 373
- unpack\_nas\_SetRFInfoCallback
  - nas.h, 603
- unpack\_nas\_SetRoamingIndicatorCallback\_ind
  - nas.h, 603
- unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, 373
  - roaming, 374
- unpack\_nas\_SetServingSystemCallback\_ind
  - nas.h, 603
- unpack\_nas\_SetServingSystemCallback\_ind\_t, 374
  - SSInfo, 374
  - Tlvresult, 374
- unpack\_nas\_SlqsGetLTECphyCAInfo
  - nas.h, 603
- unpack\_nas\_SlqsGetLTECphyCAInfo\_t, 374
  - LTECphyCAInfo, 375
  - Tlvresult, 375
- unpack\_omaDmConfigTlv\_t, 397
  - alertmsg, 398
  - alertmsglength, 398
  - state, 398
  - userInputReq, 398
  - userInputTimeout, 398
- unpack\_omaDmFotaTlv\_t, 398
  - description, 400
  - descriptionlength, 400
  - fwloadsize, 400
  - fwloadComplete, 400
  - namelength, 400
  - package\_name, 400
  - sessionType, 400
  - severity, 400
  - state, 400
  - updateCompleteStatus, 400
  - userInputReq, 400
  - userInputTimeout, 400
  - version, 400
  - versionlength, 400
- unpack\_omaDmNotificationsTlv\_t, 400
  - notification, 401
  - sessionStatus, 401
- unpack\_qmi\_t, 401
  - msgid, 401
  - type, 401
  - xid, 401
- unpack\_qos\_IPv4Addr\_t, 402
  - addr, 402
  - subnetMask, 402
- unpack\_qos\_IPv6Addr\_t, 402
  - addr, 403
  - prefixLen, 403
- unpack\_qos\_IPv6TrafCls\_t, 403
  - mask, 403
  - val, 403
- unpack\_qos\_Port\_t, 404
  - port, 404
  - range, 404
- unpack\_qos\_QosFlowInfo\_t, 405
  - BearerID, 406
  - is\_RxQFlowGranted\_Available, 406
  - is\_TxQFlowGranted\_Available, 406
  - NumRxFilters, 406
  - NumTxFilters, 406
  - QFlowState, 406
  - RxQFilter, 406
  - RxQFlowGranted, 406
  - TxQFilter, 406
  - TxQFlowGranted, 406
- unpack\_qos\_QosFlowInfoState\_t, 406
  - id, 407
  - isNewFlow, 407
  - state, 407
- unpack\_qos\_SLQSQosGetNetworkStatus
  - qos.h, 661
- unpack\_qos\_SLQSQosGetNetworkStatus\_t, 407
  - NWQoSStatus, 407
- unpack\_qos\_SLQSQosSwiReadApnExtraParams
  - qos.h, 662
- unpack\_qos\_SLQSQosSwiReadApnExtraParams\_←
  - t, 408
  - ambr\_dl, 409
  - ambr\_dl\_ext, 409
  - ambr\_dl\_ext2, 409
  - ambr\_ul, 409
  - ambr\_ul\_ext, 409
  - ambr\_ul\_ext2, 409
  - apnId, 409
- unpack\_qos\_SLQSQosSwiReadDataStats
  - qos.h, 662
- unpack\_qos\_SLQSQosSwiReadDataStats\_t, 409
  - apnId, 410
  - numQosFlow, 410
  - qosFlow, 410
  - total\_rx\_bytes, 410
  - total\_rx\_pkt, 410
  - total\_tx\_bytes, 410
  - total\_tx\_bytes\_drp, 410
  - total\_tx\_pkt, 410
  - total\_tx\_pkt\_drp, 410
- unpack\_qos\_SLQSSetQosEventCallback
  - qos.h, 663
- unpack\_qos\_SLQSSetQosEventCallback\_ind
  - qos.h, 663
- unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, 410
  - NumFlows, 411
  - QosFlowInfo, 411
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind
  - qos.h, 664
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t, 411
  - 411



- status, [411](#)
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind
  - qos.h, [665](#)
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t, [411](#)
  - event, [412](#)
- unpack\_qos\_SLQSSetQosStatusCallback\_ind
  - qos.h, [665](#)
- unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [412](#)
  - event, [413](#)
  - id, [413](#)
  - reason, [413](#)
  - status, [413](#)
- unpack\_qos\_Tos\_t, [422](#)
  - mask, [422](#)
  - val, [422](#)
- unpack\_qos\_dataRate\_t, [401](#)
  - dataRateMax, [402](#)
  - guaranteedRate, [402](#)
- unpack\_qos\_pktErrRate\_t, [404](#)
  - exponent, [404](#)
  - multiplier, [404](#)
- unpack\_qos\_swiQosFilter\_t, [413](#)
  - EspSpi, [415](#)
  - IPv4DstAddr, [416](#)
  - IPv4SrcAddr, [416](#)
  - IPv4Tos, [416](#)
  - IPv6DstAddr, [416](#)
  - IPv6Label, [416](#)
  - IPv6SrcAddr, [416](#)
  - IPv6TrafCls, [416](#)
  - Id, [415](#)
  - index, [416](#)
  - is\_EspSpi\_Available, [416](#)
  - is\_IPv4DstAddr\_Available, [416](#)
  - is\_IPv4SrcAddr\_Available, [416](#)
  - is\_IPv4Tos\_Available, [416](#)
  - is\_IPv6DstAddr\_Available, [416](#)
  - is\_IPv6Label\_Available, [416](#)
  - is\_IPv6SrcAddr\_Available, [416](#)
  - is\_IPv6TrafCls\_Available, [416](#)
  - is\_Id\_Available, [416](#)
  - is\_NxtHdrProto\_Available, [416](#)
  - is\_Precedence\_Available, [416](#)
  - is\_TCPDstPort\_Available, [416](#)
  - is\_TCPSrcPort\_Available, [416](#)
  - is\_TranDstPort\_Available, [416](#)
  - is\_TranSrcPort\_Available, [416](#)
  - is\_UDPDstPort\_Available, [417](#)
  - is\_UDPSrcPort\_Available, [417](#)
  - NxtHdrProto, [417](#)
  - Precedence, [417](#)
  - TCPDstPort, [417](#)
  - TCPSrcPort, [417](#)
  - TranDstPort, [417](#)
  - TranSrcPort, [417](#)
  - UDPDstPort, [417](#)
  - UDPSrcPort, [417](#)
  - version, [417](#)
- unpack\_qos\_swiQosFlow\_t, [417](#)
  - DataRate, [420](#)
  - index, [420](#)
  - is\_DataRate\_Available, [420](#)
  - is\_Jitter\_Available, [420](#)
  - is\_Latency\_Available, [420](#)
  - is\_LteQci\_Available, [420](#)
  - is\_MaxAllowedPktSz\_Available, [420](#)
  - is\_MinPolicedPktSz\_Available, [420](#)
  - is\_PktErrRate\_Available, [420](#)
  - is\_ProfileId3GPP2\_Available, [420](#)
  - is-TokenBucket\_Available, [420](#)
  - is\_TrafficClass\_Available, [420](#)
  - is\_val\_3GPP2Pri\_Available, [420](#)
  - is\_val\_3GPPImCn\_Available, [420](#)
  - is\_val\_3GPPResResidualBER\_Available, [420](#)
  - is\_val\_3GPPSigInd\_Available, [420](#)
  - is\_val\_3GPPTraHdlPri\_Available, [420](#)
  - Jitter, [420](#)
  - Latency, [421](#)
  - LteQci, [421](#)
  - MaxAllowedPktSz, [421](#)
  - MinPolicedPktSz, [421](#)
  - PktErrRate, [421](#)
  - ProfileId3GPP2, [421](#)
  - TokenBucket, [421](#)
  - TrafficClass, [421](#)
  - val\_3GPP2Pri, [421](#)
  - val\_3GPPImCn, [421](#)
  - val\_3GPPResResidualBER, [421](#)
  - val\_3GPPSigInd, [421](#)
  - val\_3GPPTraHdlPri, [421](#)
- unpack\_qos\_tokenBucket\_t, [421](#)
  - bucketSz, [422](#)
  - peakRate, [422](#)
  - tokenRate, [422](#)
- unpack\_result\_code\_only
  - common.h, [494](#)
- unpack\_sms\_SLQSDeleteSMS\_t, [426](#)
- unpack\_sms\_SLQSDeleteSMS
  - sms.h, [674](#)
- unpack\_sms\_SLQSGetSMS\_t, [426](#)
  - message, [427](#)
  - messageFormat, [427](#)
  - messageSize, [427](#)
  - messageTag, [427](#)
- unpack\_sms\_SLQSGetSMSList
  - sms.h, [674](#)
- unpack\_sms\_SLQSGetSMSList\_t, [427](#)
  - messageList, [427](#)
  - messageListSize, [427](#)
- unpack\_sms\_SLQSGetSMS
  - sms.h, [674](#)
- unpack\_sms\_SLQSModifySMSStatus
  - sms.h, [675](#)
- unpack\_sms\_SLQSModifySMSStatus\_t, [427](#)
- unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind
  - sms.h, [675](#)

- unpack\_sms\_SLQSWmsMemoryFullCallback\_ind\_t, 428
  - messageMode, 428
  - storageType, 428
- unpack\_sms\_SendSMS\_t, 424
  - messageFailureCode, 425
  - messageID, 425
- unpack\_sms\_SendSMS
  - sms.h, 672
- unpack\_sms\_SetNewSMSCallback
  - sms.h, 673
- unpack\_sms\_SetNewSMSCallback\_ind
  - sms.h, 673
- unpack\_sms\_SetNewSMSCallback\_ind\_t, 425
  - ETWSPLMNTIv, 426
  - ETWSTIv, 426
  - IMSTIv, 426
  - MMTIv, 426
  - NewMMTIv, 426
  - SMSCTIv, 426
  - TRMessageTIv, 426
- unpack\_sms\_SetNewSMSCallback\_t, 426
- unpack\_swiloc\_SwiLocGetAutoStart
  - swiloc.h, 678
- unpack\_swiloc\_SwiLocGetAutoStart\_t, 428
  - fix\_rate, 429
  - fix\_rate\_reported, 429
  - fix\_type, 429
  - fix\_type\_reported, 430
  - function, 430
  - function\_reported, 430
  - max\_dist, 430
  - max\_dist\_reported, 430
  - max\_time, 430
  - max\_time\_reported, 430
- unpack\_swiloc\_SwiLocSetAutoStart
  - swiloc.h, 679
- unpack\_swioma\_SLQSOMADMAAlertCallback
  - swioma.h, 685
- unpack\_swioma\_SLQSOMADMAAlertCallback\_ind
  - swioma.h, 685
- unpack\_swioma\_SLQSOMADMAAlertCallback\_ind\_t, 430
  - eventType, 431
  - SessionInfoConfig, 431
  - SessionInfoFota, 431
  - SessionInfoNotification, 431
- unpack\_swioma\_SLQSOMADMCancelSession
  - swioma.h, 686
- unpack\_swioma\_SLQSOMADMGetSessionInfo
  - swioma.h, 686
- unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, 431
  - Date, 433
  - DateLength, 433
  - PkgDescLength, 433
  - PkgDescription, 433
  - PkgName, 433
  - PkgNameLength, 433
  - RetryCount, 433
  - SessionState, 433
  - SessionType, 433
  - Severity, 433
  - Source, 433
  - SourceLength, 433
  - Status, 434
  - Time, 434
  - TimeLength, 434
  - UpdateCompleteStatus, 434
- unpack\_swioma\_SLQSOMADMGetSettings
  - swioma.h, 687
- unpack\_swioma\_SLQSOMADMGetSettings\_t, 434
  - Autosdm, 435
  - FOTAUpdate, 435
  - FOTAdownload, 435
  - FwAutoCheck, 435
  - OMADMEEnabled, 435
- unpack\_swioma\_SLQSOMADMSendSelection
  - swioma.h, 687
- unpack\_swioma\_SLQSOMADMSetSettings
  - swioma.h, 688
- unpack\_swioma\_SLQSOMADMStartSession
  - swioma.h, 688
- unpack\_swioma\_SLQSOMADMStartSession\_t, 435
  - FwAvailability, 436
- unpack\_uim\_ChangePin
  - uim.h, 695
- unpack\_uim\_ChangePin\_t, 436
  - pEncryptedPIN1, 436
  - pIndicationToken, 436
  - pRemainingRetries, 437
  - Tlvresult, 437
- unpack\_uim\_GetCardStatus
  - uim.h, 695
- unpack\_uim\_GetCardStatus\_t, 437
  - pCardStatus, 437
  - pHotSwapStatus, 437
  - Tlvresult, 437
- unpack\_uim\_ReadTransparent
  - uim.h, 696
- unpack\_uim\_ReadTransparent\_t, 437
  - pCardResult, 438
  - pEncryptedData, 438
  - pIndicationToken, 438
  - pReadResult, 438
  - Tlvresult, 438
- unpack\_uim\_SLQSUIEventRegister
  - uim.h, 697
- unpack\_uim\_SLQSUIEventRegister\_t, 440
  - eventMask, 440
- unpack\_uim\_SLQSUIGetSlotsStatus
  - uim.h, 697
- unpack\_uim\_SLQSUIGetSlotsStatus\_t, 440
  - pNumberOfPhySlot, 441
  - pUimSlotsStatus, 441
- unpack\_uim\_SLQSUIPowerDown
  - uim.h, 698



- unpack\_uim\_SLQSUIMPowerUp
  - uim.h, [698](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind
  - uim.h, [699](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_↵
  - ind\_t, [441](#)
  - pCardStatus, [441](#)
- unpack\_uim\_SLQSUIMSwitchSlot
  - uim.h, [699](#)
- unpack\_uim\_SetPinProtection
  - uim.h, [696](#)
- unpack\_uim\_SetPinProtection\_t, [438](#)
  - pEncryptedPIN1, [439](#)
  - pIndicationToken, [439](#)
  - pRemainingRetries, [439](#)
  - Tlvresult, [439](#)
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind
  - uim.h, [697](#)
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, [439](#)
  - bNumberOfPhySlots, [440](#)
  - slotsstatusChange, [440](#)
- unpack\_uim\_UnblockPin
  - uim.h, [699](#)
- unpack\_uim\_UnblockPin\_t, [441](#)
  - pEncryptedPIN1, [442](#)
  - pIndicationToken, [442](#)
  - pRemainingRetries, [442](#)
  - Tlvresult, [442](#)
- unpack\_uim\_VerifyPin
  - uim.h, [700](#)
- unpack\_uim\_VerifyPin\_t, [442](#)
  - pEncryptedPIN1, [443](#)
  - pIndicationToken, [443](#)
  - pRemainingRetries, [443](#)
  - Tlvresult, [443](#)
- unpack\_wds\_DHCPv4ClientLease\_ind
  - wds.h, [727](#)
- unpack\_wds\_DHCPv4ClientLease\_ind\_t, [443](#)
  - DHCPv4LeaseOptTlv, [443](#)
  - DHCPv4LeaseStateTlv, [443](#)
  - IPv4AddrTlv, [443](#)
  - ProfileIdTlv, [443](#)
- unpack\_wds\_DHCPv4ClientLeaseChange
  - wds.h, [728](#)
- unpack\_wds\_GetAutoconnect
  - wds.h, [728](#)
- unpack\_wds\_GetAutoconnect\_t, [443](#)
  - psetting, [444](#)
- unpack\_wds\_GetByteTotals
  - wds.h, [728](#)
- unpack\_wds\_GetByteTotals\_t, [444](#)
  - pRXTotalBytes, [444](#)
  - pTXTotalBytes, [444](#)
- unpack\_wds\_GetConnectionRate
  - wds.h, [729](#)
- unpack\_wds\_GetConnectionRate\_t, [444](#)
  - currentChannelTXRate, [445](#)
  - currentChannelTXRate, [445](#)
  - maxChannelRXRate, [445](#)
  - maxChannelTXRate, [445](#)
- unpack\_wds\_GetDataBearerTechnology
  - wds.h, [729](#)
- unpack\_wds\_GetDataBearerTechnology\_t, [445](#)
  - pDataBearer, [446](#)
- unpack\_wds\_GetDefaultProfile
  - wds.h, [730](#)
- unpack\_wds\_GetDefaultProfile\_t, [446](#)
  - apnname, [446](#)
  - apnsize, [446](#)
  - auth, [446](#)
  - ipaddr, [446](#)
  - ipaddrv6, [447](#)
  - name, [447](#)
  - namesize, [447](#)
  - pdptype, [447](#)
  - pridns, [447](#)
  - pridnsv6, [447](#)
  - secdns, [447](#)
  - secdnsv6, [447](#)
  - username, [447](#)
  - usersize, [447](#)
- unpack\_wds\_GetDefaultProfileNum
  - wds.h, [730](#)
- unpack\_wds\_GetDefaultProfileNum\_t, [447](#)
  - index, [447](#)
- unpack\_wds\_GetDormancyState
  - wds.h, [730](#)
- unpack\_wds\_GetDormancyState\_t, [447](#)
  - dormancyState, [448](#)
- unpack\_wds\_GetLastMobileIPError
  - wds.h, [731](#)
- unpack\_wds\_GetLastMobileIPError\_t, [448](#)
  - error, [448](#)
- unpack\_wds\_GetMobileIP\_t, [448](#)
  - mipMode, [448](#)
- unpack\_wds\_GetMobileIPProfile
  - wds.h, [732](#)
- unpack\_wds\_GetMobileIPProfile\_t, [448](#)
  - AAASPI, [449](#)
  - AAASPI, [449](#)
  - address, [449](#)
  - enabled, [449](#)
  - HASPI, [449](#)
  - HASPI, [449](#)
  - NAI, [449](#)
  - naiSize, [449](#)
  - primaryHA, [449](#)
  - revTunneling, [449](#)
  - secondaryHA, [450](#)
- unpack\_wds\_GetMobileIP
  - wds.h, [731](#)
- unpack\_wds\_GetPacketStatistics
  - wds.h, [732](#)
- unpack\_wds\_GetPacketStatistics\_t, [450](#)
  - pRXDroppedCount, [451](#)

- pRXOKBytesLastCall, [451](#)
  - pRXOkBytesCount, [451](#)
  - pRXPacketErrors, [451](#)
  - pRXPacketOverflows, [451](#)
  - pRXPacketSuccesses, [451](#)
  - pTXDroppedCount, [451](#)
  - pTXOKBytesLastCall, [451](#)
  - pTXOkBytesCount, [451](#)
  - pTXPacketErrors, [451](#)
  - pTXPacketOverflows, [451](#)
  - pTXPacketSuccesses, [451](#)
- unpack\_wds\_GetPacketStatus
  - wds.h, [732](#)
- unpack\_wds\_GetPacketStatus\_t, [451](#)
  - rXDroppedCount, [452](#)
  - rXOKBytesLastCall, [452](#)
  - rXOkBytesCount, [452](#)
  - rXPacketErrors, [452](#)
  - rXPacketOverflows, [452](#)
  - rXPacketSuccesses, [452](#)
  - tXDroppedCount, [452](#)
  - tXOKBytesLastCall, [453](#)
  - tXOkBytesCount, [452](#)
  - tXPacketErrors, [453](#)
  - tXPacketOverflows, [453](#)
  - tXPacketSuccesses, [453](#)
- unpack\_wds\_GetSessionDuration
  - wds.h, [733](#)
- unpack\_wds\_GetSessionDuration\_t, [453](#)
  - callDuration, [453](#)
- unpack\_wds\_GetSessionState
  - wds.h, [733](#)
- unpack\_wds\_GetSessionState\_t, [453](#)
  - connectionStatus, [453](#)
- unpack\_wds\_RMSetTransferStatistics
  - wds.h, [734](#)
- unpack\_wds\_RMSetTransferStatistics\_t, [454](#)
- unpack\_wds\_RMTransferStatistics\_ind
  - wds.h, [734](#)
- unpack\_wds\_SLQSCreateProfile
  - wds.h, [737](#)
- unpack\_wds\_SLQSCreateProfile\_t, [454](#)
  - pCreateProfileOut, [454](#)
  - pProfileID, [454](#)
  - Tlvresult, [454](#)
- unpack\_wds\_SLQSDeleteProfile
  - wds.h, [737](#)
- unpack\_wds\_SLQSDeleteProfile\_t, [454](#)
  - extendedErrorCode, [454](#)
- unpack\_wds\_SLQSGet3GPPConfigItem
  - wds.h, [737](#)
- unpack\_wds\_SLQSGet3GPPConfigItem\_t, [455](#)
  - \_3gppRelease, [456](#)
  - defaultPDNEnabled, [456](#)
  - LTEAttachProfile, [456](#)
  - LTEAttachProfileList, [456](#)
  - LTEAttachProfileListLen, [456](#)
  - profileList, [456](#)
- unpack\_wds\_SLQSGetCurrDataSystemStat
  - wds.h, [738](#)
- unpack\_wds\_SLQSGetCurrDataSystemStat\_t, [456](#)
  - currNetworkInfo, [456](#)
  - networkInfoLen, [456](#)
  - prefNetwork, [456](#)
- unpack\_wds\_SLQSGetCurrentChannelRate
  - wds.h, [738](#)
- unpack\_wds\_SLQSGetCurrentChannelRate\_t, [457](#)
  - current\_channel\_rx\_rate, [457](#)
  - current\_channel\_tx\_rate, [457](#)
  - max\_channel\_rx\_rate, [457](#)
  - max\_channel\_tx\_rate, [457](#)
- unpack\_wds\_SLQSGetDUNCallInfo
  - wds.h, [739](#)
- unpack\_wds\_SLQSGetDUNCallInfo\_t, [458](#)
  - callEndReason, [459](#)
  - channelRate, [459](#)
  - connectionStatus, [459](#)
  - dataBearerTech, [459](#)
  - dormancyStatus, [459](#)
  - lastCallDataBearerTech, [459](#)
  - lastCallRXOKBytesCnt, [459](#)
  - lastCallTXOKBytesCnt, [459](#)
  - mdmCallDurationActive, [459](#)
  - rxOKBytesCount, [459](#)
  - txOKBytesCount, [459](#)
- unpack\_wds\_SLQSGetDataBearerTechnology
  - wds.h, [739](#)
- unpack\_wds\_SLQSGetDataBearerTechnology\_t, [458](#)
  - curDataBearerTechnology, [458](#)
  - dataBearerMask, [458](#)
  - lastCallDataBearerTechnology, [458](#)
- unpack\_wds\_SLQSGetProfileSettings
  - wds.h, [739](#)
- unpack\_wds\_SLQSGetProfileSettings\_t, [459](#)
  - pProfileSettings, [460](#)
  - ProfileType, [460](#)
  - Tlvresult, [460](#)
- unpack\_wds\_SLQSGetRuntimeSettings
  - wds.h, [740](#)
- unpack\_wds\_SLQSGetRuntimeSettings\_t, [460](#)
  - APNName, [461](#)
  - Authentication, [461](#)
  - DomainList, [461](#)
  - GPRSGrantedQoS, [461](#)
  - GWAddressV4, [461](#)
  - IMCNflag, [461](#)
  - IPFamilyPreference, [461](#)
  - IPv6AddrInfo, [461](#)
  - IPv6GWAddrInfo, [461](#)
  - IPv4, [461](#)
  - Mtu, [461](#)
  - PCSCFAddrPCO, [461](#)
  - PCSCFFQDNAddrList, [461](#)
  - PDPTType, [461](#)
  - PrimaryDNSV4, [462](#)
  - PrimaryDNSV6, [462](#)

- ProfileID, [462](#)
- ProfileName, [462](#)
- SecondaryDNSV4, [462](#)
- SecondaryDNSV6, [462](#)
- ServerAddrList, [462](#)
- SubnetMaskV4, [462](#)
- Technology, [462](#)
- UMTSGrantedQoS, [462](#)
- Username, [462](#)
- unpack\_wds\_SLQSMModifyProfile
  - wds.h, [740](#)
- unpack\_wds\_SLQSMModifyProfile\_t, [462](#)
  - pExtErrorCode, [462](#)
- unpack\_wds\_SLQSRResetPacketStatics
  - wds.h, [740](#)
- unpack\_wds\_SLQSSGetDHCPv4ClientConfig
  - wds.h, [742](#)
- unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t, [465](#)
  - pHwConfig, [466](#)
  - pRequestOptionList, [466](#)
- unpack\_wds\_SLQSSGetLoopback
  - wds.h, [743](#)
- unpack\_wds\_SLQSSGetLoopback\_t, [466](#)
  - ByteLoopbackMode, [466](#)
  - ByteLoopbackMultiplier, [466](#)
- unpack\_wds\_SLQSSSetDHCPv4ClientConfig
  - wds.h, [743](#)
- unpack\_wds\_SLQSSSetLoopback
  - wds.h, [744](#)
- unpack\_wds\_SLQSSSet3GPPConfigItem
  - wds.h, [741](#)
- unpack\_wds\_SLQSSSetIPFamilyPreference
  - wds.h, [741](#)
- unpack\_wds\_SLQSSSetIPFamilyPreference\_t, [462](#)
  - Tlvresult, [463](#)
- unpack\_wds\_SLQSSSetPacketSrvStatusCallback
  - wds.h, [741](#)
- unpack\_wds\_SLQSSSetPacketSrvStatusCallback\_t, [463](#)
  - bearerID, [463](#)
  - conn\_status, [463](#)
  - ipFamily, [463](#)
  - reconfigReqd, [464](#)
  - sessionEndReason, [464](#)
  - techName, [464](#)
  - verboseSessnEndReason, [464](#)
  - verboseSessnEndReasonType, [464](#)
- unpack\_wds\_SLQSSSetWdsEventCallback
  - wds.h, [742](#)
- unpack\_wds\_SLQSSSetWdsEventCallback\_ind
  - wds.h, [742](#)
- unpack\_wds\_SLQSSSetWdsEventCallback\_ind\_t, [464](#)
  - currDBTechAvail, [465](#)
  - currNWInfo, [465](#)
  - dBTechAvail, [465](#)
  - dBTechnology, [465](#)
  - dataSysStatAvail, [465](#)
  - dormancyStatAvail, [465](#)
  - dormancyStatus, [465](#)
  - mipStatus, [465](#)
  - mipstatAvail, [465](#)
  - netInfoLen, [465](#)
  - prefNetwork, [465](#)
  - ratMask, [465](#)
  - rx\_bytes, [465](#)
  - rx\_pkts, [465](#)
  - soMask, [465](#)
  - tx\_bytes, [465](#)
  - tx\_pkts, [465](#)
  - xferStatAvail, [465](#)
- unpack\_wds\_SLQSStartDataSession
  - wds.h, [744](#)
- unpack\_wds\_SLQSStartDataSession\_t, [466](#)
  - pFailureReason, [467](#)
  - pVerboseFailReasonType, [467](#)
  - pVerboseFailureReason, [467](#)
  - psid, [467](#)
- unpack\_wds\_SLQSStopDataSession
  - wds.h, [744](#)
- unpack\_wds\_SLQSWdsGoActive
  - wds.h, [745](#)
- unpack\_wds\_SLQSWdsGoDormant
  - wds.h, [745](#)
- unpack\_wds\_SLQSWdsSetEventReport
  - wds.h, [745](#)
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings
  - wds.h, [746](#)
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_↔
  - t, [467](#)
  - apnName, [468](#)
  - bearerId, [468](#)
  - contextId, [468](#)
  - ipv4Address, [468](#)
  - ipv4GWAddress, [468](#)
  - ipv6Address, [468](#)
  - ipv6GWAddress, [468](#)
  - prDNSIPv4Address, [468](#)
  - prDNSIPv6Address, [468](#)
  - prPCSCFIPv4Address, [468](#)
  - prPCSCFIPv6Address, [468](#)
  - seDNSIPv4Address, [468](#)
  - seDNSIPv6Address, [469](#)
  - sePCSCFIPv4Address, [469](#)
  - sePCSCFIPv6Address, [469](#)
- unpack\_wds\_SetAutoconnect
  - wds.h, [734](#)
- unpack\_wds\_SetDefaultProfile
  - wds.h, [735](#)
- unpack\_wds\_SetDefaultProfileNum
  - wds.h, [735](#)
- unpack\_wds\_SetMobileIPParameters
  - wds.h, [736](#)
- unpack\_wds\_SetMobileIPProfile
  - wds.h, [736](#)
- unpack\_wds\_SetMobileIPProfile\_t, [454](#)
- unpack\_wds\_SetMobileIP
  - wds.h, [735](#)

- UnpackQmiProfileInfo
  - wds.h, 706
- unpackWdsProfileParam, 469
  - SlqsProfile3GPP2, 469
  - SlqsProfile3GPP, 469
- UpdateCompleteStatus
  - unpack\_swioma\_SLQSOMADMGetSessionInfo↔\_t, 434
- updateCompleteStatus
  - unpack\_omaDmFotaTlv\_t, 400
- upinRetries
  - slotInf, 284
  - uim\_slotInfo, 305
- upinState
  - slotInf, 284
  - uim\_slotInfo, 305
- upukRetries
  - slotInf, 284
  - uim\_slotInfo, 305
- usageMask
  - loc\_sensorDataUsage, 83
- userInputReq
  - unpack\_omaDmConfigTlv\_t, 398
  - unpack\_omaDmFotaTlv\_t, 400
- userInputTimeout
  - unpack\_omaDmConfigTlv\_t, 398
  - unpack\_omaDmFotaTlv\_t, 400
- Username
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 462
- username
  - unpack\_wds\_GetDefaultProfile\_t, 447
- usersize
  - unpack\_wds\_GetDefaultProfile\_t, 447
- VDOP
  - loc\_precisionDilution, 80
- val
  - unpack\_qos\_IPv6TrafCls\_t, 403
  - unpack\_qos\_Tos\_t, 422
- val\_3GPP2Pri
  - unpack\_qos\_swiQosFlow\_t, 421
- val\_3GPPImCn
  - unpack\_qos\_swiQosFlow\_t, 421
- val\_3GPPResResidualBER
  - unpack\_qos\_swiQosFlow\_t, 421
- val\_3GPPSigInd
  - unpack\_qos\_swiQosFlow\_t, 421
- val\_3GPPTraHdlPri
  - unpack\_qos\_swiQosFlow\_t, 421
- validMask
  - loc\_satelliteInfo, 82
- ValidityCW0
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 393
- ValidityCW1
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 393
- value
  - pack\_dms\_UIMSetPINProtection\_t, 199
  - pack\_dms\_UIMVerifyPIN\_t, 201
- value\_length
  - DMScustSettingInfo, 35
  - pack\_dms\_SetCustFeaturesV2\_t, 192
- verboseSessnEndReason
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback↔\_t, 464
- verboseSessnEndReasonType
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback↔\_t, 464
- verifyLeft
  - uim\_remainingRetries, 302
- verifyPIN
  - pack\_uim\_VerifyPin\_t, 255
- verifyRetriesLeft
  - unpack\_dms\_UIMGetControlKeyStatus\_t, 339
  - unpack\_dms\_UIMSetControlKeyProtection\_t, 342
  - unpack\_dms\_UIMSetPINProtection\_t, 342
- version
  - unpack\_omaDmFotaTlv\_t, 400
  - unpack\_qos\_swiQosFilter\_t, 417
- versionString
  - pack\_dms\_SLQSSwiSetOSInfo\_t, 196
  - unpack\_dms\_SLQSSwiGetOSInfo\_t, 335
- versionlength
  - unpack\_omaDmFotaTlv\_t, 400
- vertConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- vertReliability
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- vertUnc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 213
- VoiceNumber
  - unpack\_dms\_GetVoiceNumber\_t, 321
- voiceNumberSize
  - unpack\_dms\_GetVoiceNumber\_t, 321
- WCDMACellInfo
  - nas\_lteWcdmaCellInfo, 135
- WCDMAECIOThreshListLen
  - nas\_WCDMAECIOThresh, 171
- WCDMARSSIOThreshListLen
  - nas\_WCDMARSSIOThresh, 172
- WCDMASSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 388
- WDS\_DHCP\_MAX\_NUM\_OPTIONS
  - wds.h, 706
- WDS\_DHCP\_OPTION\_DATA\_BUF\_SIZE
  - wds.h, 706
- WORD
  - SwiDataTypes.h, 677
- wcdmaRRCState
  - nas\_WCDMAInfoLTENeighborCell, 172
- wds.h, 700
  - BYT\_STAT\_STAT\_MASK, 706
  - IPV6\_ADDRESS\_ARRAY\_SIZE, 706
  - MAX\_WDS\_3GPP\_CONF\_LTE\_ATTACH\_PRO↔FILE\_LIST\_SIZE, 706
  - PACK\_WDS\_IPV4, 706
  - PACK\_WDS\_IPV6, 706
  - pack\_wds\_DHCPv4ClientLeaseChange, 706

- pack\_wds\_GetAutoconnect, 707
- pack\_wds\_GetByteTotals, 707
- pack\_wds\_GetConnectionRate, 708
- pack\_wds\_GetDataBearerTechnology, 708
- pack\_wds\_GetDefaultProfile, 709
- pack\_wds\_GetDefaultProfileNum, 709
- pack\_wds\_GetDormancyState, 709
- pack\_wds\_GetLastMobileIPError, 710
- pack\_wds\_GetMobileIPProfile, 711
- pack\_wds\_GetMobileIP, 710
- pack\_wds\_GetPacketStatistics, 711
- pack\_wds\_GetPacketStatus, 712
- pack\_wds\_GetSessionDuration, 712
- pack\_wds\_GetSessionState, 713
- pack\_wds\_RMSetTransferStatistics, 713
- pack\_wds\_SLQSCreateProfile, 716
- pack\_wds\_SLQSDeleteProfile, 717
- pack\_wds\_SLQSGet3GPPConfigItem, 717
- pack\_wds\_SLQSGetCurrDataSystemStat, 718
- pack\_wds\_SLQSGetCurrentChannelRate, 718
- pack\_wds\_SLQSGetDUNCallInfo, 719
- pack\_wds\_SLQSGetDataBearerTechnology, 719
- pack\_wds\_SLQSGetProfileSettings, 719
- pack\_wds\_SLQSGetRuntimeSettings, 720
- pack\_wds\_SLQSModifyProfile, 720
- pack\_wds\_SLQSResetPacketStatics, 721
- pack\_wds\_SLQSSetDHCPv4ClientConfig, 723
- pack\_wds\_SLQSSetLoopback, 723
- pack\_wds\_SLQSSetDHCPv4ClientConfig, 724
- pack\_wds\_SLQSSetLoopback, 724
- pack\_wds\_SLQSSet3GPPConfigItem, 721
- pack\_wds\_SLQSSetIPFamilyPreference, 722
- pack\_wds\_SLQSSetWdsEventCallback, 722
- pack\_wds\_SLQSStartDataSession, 724
- pack\_wds\_SLQSStopDataSession, 725
- pack\_wds\_SLQSWdsGoActive, 725
- pack\_wds\_SLQSWdsGoDormant, 726
- pack\_wds\_SLQSWdsSetEventReport, 726
- pack\_wds\_SLQSWdsSwiPDPRuntimeSettings, 727
- pack\_wds\_SetAutoconnect, 714
- pack\_wds\_SetDefaultProfile, 714
- pack\_wds\_SetDefaultProfileNum, 714
- pack\_wds\_SetMobileIPParameters, 715
- pack\_wds\_SetMobileIPProfile, 716
- pack\_wds\_SetMobileIP, 715
- unpack\_wds\_DHCPv4ClientLease\_ind, 727
- unpack\_wds\_DHCPv4ClientLeaseChange, 728
- unpack\_wds\_GetAutoconnect, 728
- unpack\_wds\_GetByteTotals, 728
- unpack\_wds\_GetConnectionRate, 729
- unpack\_wds\_GetDataBearerTechnology, 729
- unpack\_wds\_GetDefaultProfile, 730
- unpack\_wds\_GetDefaultProfileNum, 730
- unpack\_wds\_GetDormancyState, 730
- unpack\_wds\_GetLastMobileIPError, 731
- unpack\_wds\_GetMobileIPProfile, 732
- unpack\_wds\_GetMobileIP, 731
- unpack\_wds\_GetPacketStatistics, 732
- unpack\_wds\_GetPacketStatus, 732
- unpack\_wds\_GetSessionDuration, 733
- unpack\_wds\_GetSessionState, 733
- unpack\_wds\_RMSetTransferStatistics, 734
- unpack\_wds\_RMTransferStatistics\_ind, 734
- unpack\_wds\_SLQSCreateProfile, 737
- unpack\_wds\_SLQSDeleteProfile, 737
- unpack\_wds\_SLQSGet3GPPConfigItem, 737
- unpack\_wds\_SLQSGetCurrDataSystemStat, 738
- unpack\_wds\_SLQSGetCurrentChannelRate, 738
- unpack\_wds\_SLQSGetDUNCallInfo, 739
- unpack\_wds\_SLQSGetDataBearerTechnology, 739
- unpack\_wds\_SLQSGetProfileSettings, 739
- unpack\_wds\_SLQSGetRuntimeSettings, 740
- unpack\_wds\_SLQSModifyProfile, 740
- unpack\_wds\_SLQSResetPacketStatics, 740
- unpack\_wds\_SLQSSetDHCPv4ClientConfig, 742
- unpack\_wds\_SLQSSetLoopback, 743
- unpack\_wds\_SLQSSetDHCPv4ClientConfig, 743
- unpack\_wds\_SLQSSetLoopback, 744
- unpack\_wds\_SLQSSet3GPPConfigItem, 741
- unpack\_wds\_SLQSSetIPFamilyPreference, 741
- unpack\_wds\_SLQSSetPacketSrvStatusCallback, 741
- unpack\_wds\_SLQSSetWdsEventCallback, 742
- unpack\_wds\_SLQSSetWdsEventCallback\_ind, 742
- unpack\_wds\_SLQSStartDataSession, 744
- unpack\_wds\_SLQSStopDataSession, 744
- unpack\_wds\_SLQSWdsGoActive, 745
- unpack\_wds\_SLQSWdsGoDormant, 745
- unpack\_wds\_SLQSWdsSetEventReport, 745
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings, 746
- unpack\_wds\_SetAutoconnect, 734
- unpack\_wds\_SetDefaultProfile, 735
- unpack\_wds\_SetDefaultProfileNum, 735
- unpack\_wds\_SetMobileIPParameters, 736
- unpack\_wds\_SetMobileIPProfile, 736
- unpack\_wds\_SetMobileIP, 735
- UnpackQmiProfileInfo, 706
- WDS\_DHCP\_MAX\_NUM\_OPTIONS, 706
- WDS\_DHCP\_OPTION\_DATA\_BUF\_SIZE, 706
- wds\_DHCPLeaseOptTlv, 472
  - numOpt, 472
  - optList, 472
  - optListData, 472
  - TlvPresent, 472
- wds\_DHCPLeaseStateTlv, 472
  - leaseState, 472
  - TlvPresent, 472
- wds\_DHCPOpt, 472
  - optCode, 473
  - optValLen, 473
  - pOptVal, 473

- wds\_DHCPProfileIdTlv, 473
  - profileId, 473
  - profileType, 473
  - TlvPresent, 473
- wds\_DHCPv4HWConfig, 473
  - chaddr, 474
  - chaddrLen, 474
  - hwType, 474
- wds\_DHCPv4Option, 474
  - optCode, 474
  - optVal, 474
  - optValLen, 474
- wds\_DHCPv4OptionList, 474
  - numOpt, 475
  - pOptList, 475
- wds\_DHCPv4ProfileId, 475
  - profileId, 475
  - profileType, 475
- wds\_DataULongLongTlv, 471
  - TlvPresent, 471
  - ulldata, 471
- wds\_DataULongTlv, 471
  - TlvPresent, 472
  - ulldata, 472
- wds\_Domain, 476
  - domainLen, 476
  - domainName, 476
- wds\_DomainNameList, 476
  - domain, 477
  - numInstances, 477
- wds\_GPRSQoS, 477
  - delayClass, 477
  - meanThroughputClass, 477
  - peakThroughputClass, 477
  - precedenceClass, 477
  - reliabilityClass, 477
- wds\_IPV6AddressInfo, 478
  - IPAddressV6, 478
  - IPv6PrefixLen, 478
- wds\_IPV6GWAddressInfo, 478
  - gwAddressV6, 479
  - gwV6PrefixLen, 479
- wds\_IPv4AdTlv, 478
  - IPv4Addr, 478
  - TlvPresent, 478
- wds\_PCSCFFQDNAddress, 479
  - fqdnAddr, 479
  - fqdnLen, 480
- wds\_PCSCFFQDNAddressList, 480
  - numInstances, 480
  - pcsfFQDNAddress, 480
- wds\_PCSCFIPv4ServerAddressList, 480
  - numInstances, 481
  - pcsfIPv4Addr, 481
- wds\_ProfileIdentifier, 481
  - profileIndex, 481
  - profileType, 481
- wds\_TrStatInd, 482
  - statsMask, 482
  - statsPeriod, 482
- wds\_UMTSMInQoS, 482
  - deliveryErrSDU, 484
  - grntDownlinkBitrate, 484
  - grntUplinkBitrate, 485
  - maxDownlinkBitrate, 485
  - maxSDUSize, 485
  - maxUplinkBitrate, 485
  - qosDeliveryOrder, 485
  - resBerRatio, 485
  - sduErrorRatio, 485
  - trafficClass, 485
  - trafficPriority, 485
  - transferDelay, 485
- wds\_currNetworkInfo, 469
  - NetworkType, 471
  - RATMask, 471
  - SOMask, 471
- wds\_profileInfo, 481
  - SlqsProfile3GPP2, 482
  - SlqsProfile3GPP, 482
- wdsDhcpv4HwConfig, 485
  - chaddr, 485
  - chaddrLen, 485
  - hwType, 486
- wdsDhcpv4Option, 486
  - optCode, 486
  - optVal, 486
  - optValLen, 486
- wdsDhcpv4OptionList, 486
  - numOpt, 487
  - pOptList, 487
- wdsDhcpv4ProfileId, 487
  - profileId, 487
  - profileType, 487
- xAxis
  - sensorData\_t, 281
- xferStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 465
- xid
  - pack\_loc\_SLQSLOCGetBestAvailPos\_t, 208
  - pack\_qmi\_t, 237
  - unpack\_qmi\_t, 401
- yAxis
  - sensorData\_t, 281
- year
  - nas\_UniversalTime, 169
  - nas\_timeInfo, 164
- zAxis
  - sensorData\_t, 281