# LinuxQMISDK-Lite

SLQS04.00.17

Generated by Doxygen 1.8.6

Fri Oct 26 2018 06:15:28

# Contents


**1 Welcome to the Sierra Wireless Linux QMI SDK API Reference Guide** **1**

1.1 Important Notice . . . . . 1

1.2 Limitation of Liability . . . . . 1

1.3 Patents . . . . . 1

1.4 Copyright . . . . . 2

1.5 Trademarks . . . . . 2

1.6 Contact Information . . . . . 2

**2 Module Index** **3**

2.1 Modules . . . . . 3

**3 Namespace Index** **5**

3.1 Namespace List . . . . . 5

**4 Data Structure Index** **7**

4.1 Data Structures . . . . . 7

**5 File Index** **27**

5.1 File List . . . . . 27

**6 Module Documentation** **29**

6.1 QMI pack/unpack (lite-qmi) . . . . . 29

6.1.1 Detailed Description . . . . . 29

6.2 Streaming Download Protocol (lite-fw) . . . . . 30

6.2.1 Detailed Description . . . . . 30

**7 Namespace Documentation** **31**

7.1 Tables Namespace Reference . . . . . 31

7.1.1 Detailed Description . . . . . 31

**8 Data Structure Documentation** **33**

8.1 _litefw_FirmwareInfo_ Struct Reference . . . . . 33

8.1.1 Detailed Description . . . . . 33

8.1.2 Field Documentation . . . . . 33

8.1.2.1 szCarrier_str . . . 34
8.1.2.2 szCarrierPriversion_str . . . 34
8.1.2.3 szFwversion_str . . . 34
8.1.2.4 szModelid_str . . . 34
8.1.2.5 szPackageid_str . . . 34
8.1.2.6 szSku_str . . . 34
8.2 altSrcInfo_t Struct Reference . . . 34
8.2.1 Detailed Description . . . 34
8.2.2 Field Documentation . . . 35
8.2.2.1 coverage . . . 35
8.2.2.2 linkage . . . 35
8.2.2.3 source . . . 35
8.3 appStats Struct Reference . . . 35
8.3.1 Detailed Description . . . 35
8.3.2 Field Documentation . . . 37
8.3.2.1 aidLength . . . 37
8.3.2.2 aidVal . . . 37
8.3.2.3 appState . . . 37
8.3.2.4 appType . . . 37
8.3.2.5 persoFeature . . . 37
8.3.2.6 persoRetries . . . 37
8.3.2.7 persoState . . . 37
8.3.2.8 persoUnblockRetries . . . 38
8.3.2.9 pin1Retries . . . 38
8.3.2.10 pin1State . . . 38
8.3.2.11 pin2Retries . . . 38
8.3.2.12 pin2State . . . 38
8.3.2.13 puk1Retries . . . 38
8.3.2.14 puk2Retries . . . 38
8.3.2.15 univPin . . . 38
8.4 audio_RXAGCList Struct Reference . . . 38
8.4.1 Detailed Description . . . 38
8.4.2 Field Documentation . . . 38
8.4.2.1 pRXAIG . . . 39
8.4.2.2 pRXComprSlope . . . 39
8.4.2.3 pRXComprThres . . . 39
8.4.2.4 pRXExpSlope . . . 39
8.4.2.5 pRXExpThres . . . 39
8.4.2.6 pRXStaticGain . . . 39
8.5 audio_RXAVCList Struct Reference . . . 39

8.5.1 Detailed Description . . . 39
8.5.2 Field Documentation . . . 39
8.5.2.1 pAVRXAVCHeadroom . . . 39
8.5.2.2 pAVRXAVCSens . . . 39
8.6 audio_RXPCMIIRFltr Struct Reference . . . 39
8.6.1 Detailed Description . . . 39
8.6.2 Field Documentation . . . 41
8.6.2.1 pFlag . . . 41
8.6.2.2 pStage0Val . . . 41
8.6.2.3 pStage1Val . . . 41
8.6.2.4 pStage2Val . . . 41
8.6.2.5 pStage3Val . . . 41
8.6.2.6 pStage4Val . . . 41
8.6.2.7 pStageCnt . . . 41
8.7 audio_TXAGCList Struct Reference . . . 41
8.7.1 Detailed Description . . . 41
8.7.2 Field Documentation . . . 41
8.7.2.1 pTXAIG . . . 42
8.7.2.2 pTXComprSlope . . . 42
8.7.2.3 pTXComprThres . . . 42
8.7.2.4 pTXExpSlope . . . 42
8.7.2.5 pTXExpThres . . . 42
8.7.2.6 pTXStaticGain . . . 42
8.8 audio_TXPCMIIRFltr Struct Reference . . . 42
8.8.1 Detailed Description . . . 42
8.8.2 Field Documentation . . . 43
8.8.2.1 pFlag . . . 43
8.8.2.2 pStage0Val . . . 43
8.8.2.3 pStage1Val . . . 43
8.8.2.4 pStage2Val . . . 43
8.8.2.5 pStage3Val . . . 43
8.8.2.6 pStage4Val . . . 43
8.8.2.7 pStageCnt . . . 43
8.9 CarrierImage_t Struct Reference . . . 43
8.9.1 Detailed Description . . . 44
8.9.2 Field Documentation . . . 44
8.9.2.1 m_FwBuildId . . . 44
8.9.2.2 m_FwImageId . . . 44
8.9.2.3 m_nCarrierId . . . 44
8.9.2.4 m_nFolderId . . . 44

8.9.2.5 m_nStorage . . . 44
8.9.2.6 m_PriBuildId . . . 45
8.9.2.7 m_PriImageId . . . 45
8.10 cat_AlPhaIdentifierTlv Struct Reference . . . 45
8.10.1 Detailed Description . . . 45
8.10.2 Field Documentation . . . 45
8.10.2.1 AlphaID . . . 45
8.10.2.2 AlphaIDLength . . . 45
8.10.2.3 ReferenceID . . . 45
8.11 cat_commonEventTlv Struct Reference . . . 45
8.11.1 Detailed Description . . . 45
8.11.2 Field Documentation . . . 46
8.11.2.1 CatEvent . . . 46
8.11.2.2 EventID . . . 46
8.11.2.3 EventLength . . . 46
8.11.2.4 TlvPresent . . . 46
8.12 cat_currentCatEvent Union Reference . . . 46
8.12.1 Detailed Description . . . 46
8.12.2 Field Documentation . . . 46
8.12.2.1 CatAlphaIdtfr . . . 46
8.12.2.2 CatEndPS . . . 46
8.12.2.3 CatEventLst . . . 46
8.12.2.4 CatEvIDData . . . 46
8.12.2.5 CatRefresh . . . 46
8.13 cat_EndProactiveSessionTlv Struct Reference . . . 47
8.13.1 Detailed Description . . . 47
8.13.2 Field Documentation . . . 47
8.13.2.1 EndProactiveSession . . . 47
8.14 cat_EventIDDataTlv Struct Reference . . . 47
8.14.1 Detailed Description . . . 47
8.14.2 Field Documentation . . . 47
8.14.2.1 Data . . . 47
8.14.2.2 DataLength . . . 47
8.14.2.3 ReferenceID . . . 47
8.15 cat_EventListTlv Struct Reference . . . 47
8.15.1 Detailed Description . . . 48
8.15.2 Field Documentation . . . 48
8.15.2.1 SetupEventList . . . 48
8.16 cat_RefreshTlv Struct Reference . . . 48
8.16.1 Detailed Description . . . 48

8.16.2 Field Documentation . . . 48
8.16.2.1 RefreshMode . . . 48
8.16.2.2 RefreshStage . . . 48
8.17 cdmaSSInfo Struct Reference . . . 48
8.17.1 Detailed Description . . . 49
8.17.2 Field Documentation . . . 49
8.17.2.1 ecio . . . 49
8.17.2.2 rssi . . . 49
8.18 connectionStatus Struct Reference . . . 49
8.18.1 Detailed Description . . . 49
8.18.2 Field Documentation . . . 49
8.18.2.1 MDMCallDuration . . . 49
8.18.2.2 MDMConnStatus . . . 50
8.19 crashInfoParams Struct Reference . . . 50
8.19.1 Detailed Description . . . 50
8.19.2 Field Documentation . . . 50
8.19.2.1 crashInfo . . . 50
8.19.2.2 crashStatus . . . 50
8.20 crashInformation Struct Reference . . . 50
8.20.1 Detailed Description . . . 50
8.20.2 Field Documentation . . . 51
8.20.2.1 crashData . . . 51
8.20.2.2 crashId . . . 51
8.20.2.3 crashString . . . 51
8.20.2.4 crashStrlen . . . 51
8.20.2.5 gcdumpString . . . 51
8.20.2.6 gcdumpStrlen . . . 51
8.20.2.7 numCrashes . . . 51
8.21 currNetworkInfo Struct Reference . . . 51
8.21.1 Detailed Description . . . 51
8.21.2 Field Documentation . . . 52
8.21.2.1 NetworkType . . . 52
8.21.2.2 RATMask . . . 52
8.21.2.3 SOMask . . . 52
8.22 dms_ActivationStatusTlv Struct Reference . . . 52
8.22.1 Detailed Description . . . 52
8.22.2 Field Documentation . . . 52
8.22.2.1 activationStatus . . . 52
8.22.2.2 TlvPresent . . . 52
8.23 dms_OperatingModeTlv Struct Reference . . . 52

8.23.1 Detailed Description . . . 53
8.23.2 Field Documentation . . . 53
8.23.2.1 operatingMode . . . 53
8.23.2.2 TlvPresent . . . 53
8.24 dms_PSMActiveTimerIndTlv Struct Reference . . . 53
8.24.1 Detailed Description . . . 53
8.24.2 Field Documentation . . . 54
8.24.2.1 ActiveTimerInd . . . 54
8.24.2.2 TlvPresent . . . 54
8.25 dms_PSMActiveTimerTlv Struct Reference . . . 54
8.25.1 Detailed Description . . . 54
8.25.2 Field Documentation . . . 54
8.25.2.1 activeTimer . . . 54
8.25.2.2 TlvPresent . . . 54
8.26 dms_PSMDurationDueToOOSTlv Struct Reference . . . 54
8.26.1 Detailed Description . . . 54
8.26.2 Field Documentation . . . 55
8.26.2.1 durationDueToOOS . . . 55
8.26.2.2 TlvPresent . . . 55
8.27 dms_PSMDurationThresholdTlv Struct Reference . . . 55
8.27.1 Detailed Description . . . 55
8.27.2 Field Documentation . . . 55
8.27.2.1 durationThreshold . . . 55
8.27.2.2 TlvPresent . . . 55
8.28 dms_PSMEarlyWakeupTimeTlv Struct Reference . . . 55
8.28.1 Detailed Description . . . 55
8.28.2 Field Documentation . . . 56
8.28.2.1 earlyWakeupTime . . . 56
8.28.2.2 TlvPresent . . . 56
8.29 dms_PSMEnableStateIndTlv Struct Reference . . . 56
8.29.1 Detailed Description . . . 56
8.29.2 Field Documentation . . . 56
8.29.2.1 EnableStateInd . . . 56
8.29.2.2 TlvPresent . . . 56
8.30 dms_PSMEnableStateTlv Struct Reference . . . 56
8.30.1 Detailed Description . . . 56
8.30.2 Field Documentation . . . 57
8.30.2.1 enableState . . . 57
8.30.2.2 TlvPresent . . . 57
8.31 dms_PSMPeriodicUpdateTimerIndTlv Struct Reference . . . 57

8.31.1 Detailed Description . . . 57
8.31.2 Field Documentation . . . 57
8.31.2.1 PeriodicUpdateTimerInd . . . 57
8.31.2.2 TlvPresent . . . 57
8.32 dms_PSMPeriodicUpdateTimerTlv Struct Reference . . . 57
8.32.1 Detailed Description . . . 57
8.32.2 Field Documentation . . . 58
8.32.2.1 periodicUpdateTimer . . . 58
8.32.2.2 TlvPresent . . . 58
8.33 dms_PSMRandomizationWindowTlv Struct Reference . . . 58
8.33.1 Detailed Description . . . 58
8.33.2 Field Documentation . . . 58
8.33.2.1 randomizationWindow . . . 58
8.33.2.2 TlvPresent . . . 58
8.34 dms_TemperatureTlv Struct Reference . . . 58
8.34.1 Detailed Description . . . 58
8.34.2 Field Documentation . . . 59
8.34.2.1 Temperature . . . 59
8.34.2.2 TempStat . . . 59
8.34.2.3 TlvPresent . . . 59
8.35 dms_VoltageTlv Struct Reference . . . 59
8.35.1 Detailed Description . . . 59
8.35.2 Field Documentation . . . 60
8.35.2.1 TlvPresent . . . 60
8.35.2.2 Voltage . . . 60
8.35.2.3 VoltStat . . . 60
8.36 DMScustSettingInfo Struct Reference . . . 60
8.36.1 Detailed Description . . . 60
8.36.2 Field Documentation . . . 60
8.36.2.1 cust_attr . . . 60
8.36.2.2 cust_id . . . 60
8.36.2.3 cust_value . . . 60
8.36.2.4 id_length . . . 61
8.36.2.5 value_length . . . 61
8.37 DMScustSettingList Struct Reference . . . 61
8.37.1 Detailed Description . . . 61
8.37.2 Field Documentation . . . 61
8.37.2.1 custSetting . . . 61
8.37.2.2 list_type . . . 61
8.37.2.3 num_instances . . . 61

8.38 DMSgetCustomFeatureV2 Struct Reference . . . 61
8.38.1 Detailed Description . . . 61
8.38.2 Field Documentation . . . 62
8.38.2.1 pCustSettingInfo . . . 62
8.38.2.2 pCustSettingList . . . 62
8.38.2.3 pGetCustomInput . . . 62
8.39 DMSgetCustomInput Struct Reference . . . 62
8.39.1 Detailed Description . . . 62
8.39.2 Field Documentation . . . 62
8.39.2.1 cust_id . . . 62
8.39.2.2 list_type . . . 62
8.40 dunchannelRate Struct Reference . . . 62
8.40.1 Detailed Description . . . 63
8.40.2 Field Documentation . . . 63
8.40.2.1 CurrChanRxRate . . . 63
8.40.2.2 CurrChanTxRate . . . 63
8.40.2.3 MaxChanRxRate . . . 63
8.40.2.4 MaxChanTxRate . . . 63
8.41 eriDataparams Struct Reference . . . 63
8.41.1 Detailed Description . . . 63
8.41.2 Field Documentation . . . 64
8.41.2.1 eriData . . . 64
8.41.2.2 eriDataLen . . . 64
8.42 eTWSPLMNInfoTlv Struct Reference . . . 64
8.42.1 Detailed Description . . . 64
8.42.2 Field Documentation . . . 64
8.42.2.1 ETWSPLMNInfo . . . 64
8.42.2.2 TlvPresent . . . 64
8.43 FMSImageElement Struct Reference . . . 64
8.43.1 Detailed Description . . . 64
8.43.2 Field Documentation . . . 65
8.43.2.1 buildId . . . 65
8.43.2.2 buildIdLength . . . 65
8.43.2.3 imageId . . . 65
8.43.2.4 imageType . . . 65
8.44 FMSImageIdElement Struct Reference . . . 65
8.44.1 Detailed Description . . . 65
8.44.2 Field Documentation . . . 66
8.44.2.1 buildID . . . 66
8.44.2.2 buildIDLength . . . 66

8.44.2.3 failureCount . . . 66
8.44.2.4 imageID . . . 66
8.44.2.5 storageIndex . . . 66
8.45 FMSImageIDEntries Struct Reference . . . 66
8.45.1 Detailed Description . . . 66
8.45.2 Field Documentation . . . 66
8.45.2.1 executingImage . . . 66
8.45.2.2 imageIDElement . . . 66
8.45.2.3 imageIDSize . . . 67
8.45.2.4 imageType . . . 67
8.45.2.5 maxImages . . . 67
8.46 FMSImageList Struct Reference . . . 67
8.46.1 Detailed Description . . . 67
8.46.2 Field Documentation . . . 67
8.46.2.1 imageIDEntries . . . 67
8.46.2.2 listSize . . . 67
8.47 FMSPrefImageList Struct Reference . . . 67
8.47.1 Detailed Description . . . 67
8.47.2 Field Documentation . . . 68
8.47.2.1 listEntries . . . 68
8.47.2.2 listSize . . . 68
8.48 hdrSSInfo Struct Reference . . . 68
8.48.1 Detailed Description . . . 68
8.48.2 Field Documentation . . . 68
8.48.2.1 ecio . . . 68
8.48.2.2 io . . . 69
8.48.2.3 rssi . . . 69
8.48.2.4 sinr . . . 69
8.49 image_info_t Struct Reference . . . 69
8.49.1 Detailed Description . . . 69
8.49.2 Field Documentation . . . 69
8.49.2.1 buildID . . . 69
8.49.2.2 buildIDLen . . . 69
8.49.2.3 imageType . . . 69
8.49.2.4 uniqueID . . . 69
8.50 ims_AMRModeInfo Struct Reference . . . 69
8.50.1 Detailed Description . . . 70
8.50.2 Field Documentation . . . 70
8.50.2.1 amrMode . . . 70
8.50.2.2 TlvPresent . . . 70

8.51 ims_AMROctAlgnInfo Struct Reference . . . 70
8.51.1 Detailed Description . . . 70
8.51.2 Field Documentation . . . 70
8.51.2.1 amrOctAlgn . . . 70
8.51.2.2 TlvPresent . . . 71
8.52 ims_AMRWBModeInfo Struct Reference . . . 71
8.52.1 Detailed Description . . . 71
8.52.2 Field Documentation . . . 71
8.52.2.1 amrWBMode . . . 71
8.52.2.2 TlvPresent . . . 71
8.53 ims_AMRWBOctAlgnInfo Struct Reference . . . 71
8.53.1 Detailed Description . . . 71
8.53.2 Field Documentation . . . 72
8.53.2.1 amrWBOctAlgn . . . 72
8.53.2.2 TlvPresent . . . 72
8.54 ims_CSCFPortNameInfo Struct Reference . . . 72
8.54.1 Detailed Description . . . 72
8.54.2 Field Documentation . . . 72
8.54.2.1 cscfPortName . . . 72
8.54.2.2 TlvPresent . . . 72
8.55 ims_EnabAMRWBInfo Struct Reference . . . 72
8.55.1 Detailed Description . . . 72
8.55.2 Field Documentation . . . 72
8.55.2.1 amrWBEnable . . . 72
8.55.2.2 TlvPresent . . . 73
8.56 ims_EnabSCRAMRInfo Struct Reference . . . 73
8.56.1 Detailed Description . . . 73
8.56.2 Field Documentation . . . 73
8.56.2.1 scrAmrEnable . . . 73
8.56.2.2 TlvPresent . . . 73
8.57 ims_EnabSCRAMRWBInfo Struct Reference . . . 73
8.57.1 Detailed Description . . . 73
8.57.2 Field Documentation . . . 73
8.57.2.1 scrAmrWBEnable . . . 73
8.57.2.2 TlvPresent . . . 73
8.58 ims_IMSDomainInfo Struct Reference . . . 74
8.58.1 Detailed Description . . . 74
8.58.2 Field Documentation . . . 74
8.58.2.1 imsDomainName . . . 74
8.58.2.2 TlvPresent . . . 74

8.59 ims_IMSTestModeInfo Struct Reference . . . 74
8.59.1 Detailed Description . . . 74
8.59.2 Field Documentation . . . 74
8.59.2.1 imsTestMode . . . 74
8.59.2.2 TlvPresent . . . 74
8.60 ims_MinSessExpInfo Struct Reference . . . 74
8.60.1 Detailed Description . . . 75
8.60.2 Field Documentation . . . 75
8.60.2.1 minSessExp . . . 75
8.60.2.2 TlvPresent . . . 75
8.61 ims_PCSCFPortInfo Struct Reference . . . 75
8.61.1 Detailed Description . . . 75
8.61.2 Field Documentation . . . 75
8.61.2.1 priCSCFPort . . . 75
8.61.2.2 TlvPresent . . . 75
8.62 ims_PhCtxtURIInfo Struct Reference . . . 75
8.62.1 Detailed Description . . . 76
8.62.2 Field Documentation . . . 76
8.62.2.1 PhCtxtURI . . . 76
8.62.2.2 TlvPresent . . . 76
8.63 ims_RngBkTmrInfo Struct Reference . . . 76
8.63.1 Detailed Description . . . 76
8.63.2 Field Documentation . . . 76
8.63.2.1 RingBkTmr . . . 76
8.63.2.2 TlvPresent . . . 76
8.64 ims_RngTmrInfo Struct Reference . . . 76
8.64.1 Detailed Description . . . 77
8.64.2 Field Documentation . . . 77
8.64.2.1 RingTmr . . . 77
8.64.2.2 TlvPresent . . . 77
8.65 ims_RTPRTCPInactTmrDurInfo Struct Reference . . . 77
8.65.1 Detailed Description . . . 77
8.65.2 Field Documentation . . . 77
8.65.2.1 InactTmr . . . 77
8.65.2.2 TlvPresent . . . 77
8.66 ims_SessDurInfo Struct Reference . . . 77
8.66.1 Detailed Description . . . 78
8.66.2 Field Documentation . . . 78
8.66.2.1 sessExp . . . 78
8.66.2.2 TlvPresent . . . 78

8.67 ims_SigCompEnInfo Struct Reference . . . 78
8.67.1 Detailed Description . . . 78
8.67.2 Field Documentation . . . 78
8.67.2.1 SigCompEn . . . 78
8.67.2.2 TlvPresent . . . 78
8.68 ims_SIPPortInfo Struct Reference . . . 78
8.68.1 Detailed Description . . . 79
8.68.2 Field Documentation . . . 79
8.68.2.1 SIPLocalPort . . . 79
8.68.2.2 TlvPresent . . . 79
8.69 ims_SIPRegnTmrInfo Struct Reference . . . 79
8.69.1 Detailed Description . . . 79
8.69.2 Field Documentation . . . 79
8.69.2.1 TlvPresent . . . 79
8.69.2.2 tmrSIPRegn . . . 79
8.70 ims_SMSFmtInfo Struct Reference . . . 79
8.70.1 Detailed Description . . . 80
8.70.2 Field Documentation . . . 80
8.70.2.1 smsFormat . . . 80
8.70.2.2 TlvPresent . . . 80
8.71 ims_SMSoIPNwInfo Struct Reference . . . 80
8.71.1 Detailed Description . . . 80
8.71.2 Field Documentation . . . 80
8.71.2.1 smsoIPNW . . . 80
8.71.2.2 TlvPresent . . . 80
8.72 ims_SubscrTmrInfo Struct Reference . . . 80
8.72.1 Detailed Description . . . 81
8.72.2 Field Documentation . . . 81
8.72.2.1 subscrTmr . . . 81
8.72.2.2 TlvPresent . . . 81
8.73 ims_TmrT1Info Struct Reference . . . 81
8.73.1 Detailed Description . . . 81
8.73.2 Field Documentation . . . 81
8.73.2.1 TlvPresent . . . 81
8.73.2.2 tmrT1 . . . 81
8.74 ims_TmrT2Info Struct Reference . . . 81
8.74.1 Detailed Description . . . 81
8.74.2 Field Documentation . . . 82
8.74.2.1 TlvPresent . . . 82
8.74.2.2 tmrT2 . . . 82

8.75 ims_TmrTfInfo Struct Reference . . . 82
8.75.1 Detailed Description . . . 82
8.75.2 Field Documentation . . . 82
8.75.2.1 TlvPresent . . . 82
8.75.2.2 tmrTf . . . 82
8.76 imsa_IMSFailErrCodeTlv Struct Reference . . . 82
8.76.1 Detailed Description . . . 82
8.76.2 Field Documentation . . . 83
8.76.2.1 ImsFailErrCode . . . 83
8.76.2.2 TlvPresent . . . 83
8.77 imsa_IMSRegStatusErrorCodeInfo Struct Reference . . . 83
8.77.1 Detailed Description . . . 83
8.77.2 Field Documentation . . . 83
8.77.2.1 ErrorCode . . . 83
8.77.2.2 TlvPresent . . . 83
8.78 imsa_IMSRegStatusInfo Struct Reference . . . 83
8.78.1 Detailed Description . . . 84
8.78.2 Field Documentation . . . 84
8.78.2.1 ImsRegistered . . . 84
8.78.2.2 TlvPresent . . . 84
8.79 imsa_NewIMSRegStatusInfo Struct Reference . . . 84
8.79.1 Detailed Description . . . 84
8.79.2 Field Documentation . . . 84
8.79.2.1 ImsRegStatus . . . 84
8.79.2.2 TlvPresent . . . 84
8.80 imsa_RatHandoverStatusInfo Struct Reference . . . 84
8.80.1 Detailed Description . . . 85
8.80.2 Field Documentation . . . 85
8.80.2.1 ErrorCodeData . . . 85
8.80.2.2 ErrorCodeLen . . . 85
8.80.2.3 RatHandoverStatus . . . 85
8.80.2.4 SourceRAT . . . 85
8.80.2.5 TargetRAT . . . 85
8.80.2.6 TlvPresent . . . 85
8.81 imsa_SmsRatInfo Struct Reference . . . 85
8.81.1 Detailed Description . . . 86
8.81.2 Field Documentation . . . 86
8.81.2.1 SmsRatVal . . . 86
8.81.2.2 TlvPresent . . . 86
8.82 imsa_SmsSvcStatusInfo Struct Reference . . . 86

8.82.1 Detailed Description . . . 86
8.82.2 Field Documentation . . . 86
8.82.2.1 SmsSvcStatus . . . 86
8.82.2.2 TlvPresent . . . 86
8.83 imsa_UtRatInfo Struct Reference . . . 86
8.83.1 Detailed Description . . . 87
8.83.2 Field Documentation . . . 87
8.83.2.1 TlvPresent . . . 87
8.83.2.2 UtRatVal . . . 87
8.84 imsa_UtSvcStatusInfo Struct Reference . . . 87
8.84.1 Detailed Description . . . 87
8.84.2 Field Documentation . . . 87
8.84.2.1 TlvPresent . . . 87
8.84.2.2 UtSvcStatus . . . 87
8.85 imsa_VoipRatInfo Struct Reference . . . 88
8.85.1 Detailed Description . . . 88
8.85.2 Field Documentation . . . 88
8.85.2.1 TlvPresent . . . 88
8.85.2.2 VoipRatVal . . . 88
8.86 imsa_VoipSvcStatusInfo Struct Reference . . . 88
8.86.1 Detailed Description . . . 88
8.86.2 Field Documentation . . . 88
8.86.2.1 TlvPresent . . . 88
8.86.2.2 VoipSvcStatus . . . 89
8.87 imsa_VtRatInfo Struct Reference . . . 89
8.87.1 Detailed Description . . . 89
8.87.2 Field Documentation . . . 89
8.87.2.1 TlvPresent . . . 89
8.87.2.2 VtRatVal . . . 89
8.88 imsa_VtSvcStatusInfo Struct Reference . . . 89
8.88.1 Detailed Description . . . 89
8.88.2 Field Documentation . . . 90
8.88.2.1 TlvPresent . . . 90
8.88.2.2 VtSvcStatus . . . 90
8.89 ipv6AddressInfo Struct Reference . . . 90
8.89.1 Detailed Description . . . 90
8.89.2 Field Documentation . . . 90
8.89.2.1 IPAddressV6 . . . 90
8.89.2.2 IPV6PrefixLen . . . 90
8.90 LibPackGPRSRequestedQoS Struct Reference . . . 90

8.90.1 Detailed Description . . . 90
8.90.2 Field Documentation . . . 91
8.90.2.1 delayClass . . . 91
8.90.2.2 meanThroughputClass . . . 91
8.90.2.3 peakThroughputClass . . . 91
8.90.2.4 precedenceClass . . . 91
8.90.2.5 reliabilityClass . . . 91
8.91 LibpackProfile3GPP Struct Reference . . . 91
8.91.1 Detailed Description . . . 92
8.91.2 Field Documentation . . . 96
8.91.2.1 pAddrAllocPref . . . 96
8.91.2.2 pAPNClass . . . 96
8.91.2.3 pAPNDisabledFlag . . . 96
8.91.2.4 pAPNName . . . 96
8.91.2.5 pAPNnameSize . . . 96
8.91.2.6 pAuthenticationPref . . . 96
8.91.2.7 pGPRSMinimumQoS . . . 96
8.91.2.8 pGPRSRequestedQos . . . 97
8.91.2.9 pImCnFlag . . . 97
8.91.2.10 pIPv4AddrPref . . . 97
8.91.2.11 pIPv6AddPref . . . 97
8.91.2.12 pPassword . . . 97
8.91.2.13 pPasswordSize . . . 97
8.91.2.14 pPcscfAddrUsingDhcp . . . 97
8.91.2.15 pPcscfAddrUsingPCO . . . 97
8.91.2.16 pPDNInactivTimeout . . . 97
8.91.2.17 pPdpAccessConFlag . . . 97
8.91.2.18 pPdpContext . . . 97
8.91.2.19 pPdpDataCompType . . . 97
8.91.2.20 pPdpHdrCompType . . . 97
8.91.2.21 pPDPtype . . . 97
8.91.2.22 pPriDNSIPv4AddPref . . . 97
8.91.2.23 pPriDNSIPv6addpref . . . 97
8.91.2.24 pPrimaryID . . . 97
8.91.2.25 pProfilename . . . 97
8.91.2.26 pProfilenameSize . . . 97
8.91.2.27 pQosClassID . . . 97
8.91.2.28 pSecDNSIPv4AddPref . . . 97
8.91.2.29 pSecDNSIPv6addpref . . . 97
8.91.2.30 pSecondaryFlag . . . 97

8.91.2.31 pSupportEmergencyCalls . . . 97
8.91.2.32 pTFTID1Params . . . 97
8.91.2.33 pTFTID2Params . . . 97
8.91.2.34 pUMTSMinQoS . . . 97
8.91.2.35 pUMTSMinQosSigInd . . . 97
8.91.2.36 pUMTSReqQoS . . . 98
8.91.2.37 pUMTSReqQoSSigInd . . . 98
8.91.2.38 pUsername . . . 98
8.91.2.39 pUsernameSize . . . 98
8.92 LibpackProfile3GPP2 Struct Reference . . . 98
8.92.1 Detailed Description . . . 98
8.92.2 Field Documentation . . . 102
8.92.2.1 pAllowLinger . . . 102
8.92.2.2 pAPNClass3GPP2 . . . 102
8.92.2.3 pAPNEnabled3GPP2 . . . 102
8.92.2.4 pApnString . . . 102
8.92.2.5 pApnStringSize . . . 102
8.92.2.6 pAppPriority . . . 102
8.92.2.7 pAppType . . . 102
8.92.2.8 pAuthPassword . . . 102
8.92.2.9 pAuthPasswordSize . . . 102
8.92.2.10 pAuthProtocol . . . 102
8.92.2.11 pAuthRetryCount . . . 102
8.92.2.12 pAuthTimeout . . . 102
8.92.2.13 pDataMode . . . 102
8.92.2.14 pDataRate . . . 102
8.92.2.15 pIpcpAckTimeout . . . 102
8.92.2.16 pIpcpCreqRetryCount . . . 102
8.92.2.17 pIsPcscfAddressNedded . . . 102
8.92.2.18 pLcpAckTimeout . . . 102
8.92.2.19 pLcpCreqRetryCount . . . 103
8.92.2.20 pNegoDnsSrvrPref . . . 103
8.92.2.21 pPDNInactivTimeout3GPP2 . . . 103
8.92.2.22 pPdnType . . . 103
8.92.2.23 pPppSessCloseTimer1x . . . 103
8.92.2.24 pPppSessCloseTimerDO . . . 103
8.92.2.25 pPrimaryV4DnsAddress . . . 103
8.92.2.26 pPriV6DnsAddress . . . 103
8.92.2.27 pRATType . . . 103
8.92.2.28 pSecondaryV4DnsAddress . . . 103

8.92.2.29 pSecV6DnsAddress . . . 103
8.92.2.30 pUserId . . . 103
8.92.2.31 pUserIdSize . . . 103
8.93 LibPackprofile_3GPP Struct Reference . . . 103
8.93.1 Detailed Description . . . 104
8.93.2 Field Documentation . . . 107
8.93.2.1 pAddrAllocPref . . . 108
8.93.2.2 pAPNClass . . . 108
8.93.2.3 pAPNDisabledFlag . . . 108
8.93.2.4 pAPNName . . . 108
8.93.2.5 pAPNnameSize . . . 108
8.93.2.6 pAuthenticationPref . . . 108
8.93.2.7 pGPRSMinimumQoS . . . 108
8.93.2.8 pGPRSRequestedQos . . . 108
8.93.2.9 pImCnFlag . . . 108
8.93.2.10 pIPv4AddrPref . . . 108
8.93.2.11 pIPv6AddPref . . . 108
8.93.2.12 pPassword . . . 108
8.93.2.13 pPasswordSize . . . 108
8.93.2.14 pPcscfAddrUsingDhcp . . . 108
8.93.2.15 pPcscfAddrUsingPCO . . . 108
8.93.2.16 pPDNInactivTimeout . . . 108
8.93.2.17 pPdpAccessConFlag . . . 108
8.93.2.18 pPdpContext . . . 108
8.93.2.19 pPdpDataCompType . . . 108
8.93.2.20 pPdpHdrCompType . . . 108
8.93.2.21 pPDPtype . . . 108
8.93.2.22 pPriDNSIPv4AddPref . . . 108
8.93.2.23 pPriDNSIPv6addpref . . . 108
8.93.2.24 pPrimaryID . . . 108
8.93.2.25 pProfilename . . . 108
8.93.2.26 pProfilenameSize . . . 108
8.93.2.27 pQosClassID . . . 108
8.93.2.28 pSecDNSIPv4AddPref . . . 108
8.93.2.29 pSecDNSIPv6addpref . . . 109
8.93.2.30 pSecondaryFlag . . . 109
8.93.2.31 pSupportEmergencyCalls . . . 109
8.93.2.32 pTFTID1Params . . . 109
8.93.2.33 pTFTID2Params . . . 109
8.93.2.34 pUMTSMinQoS . . . 109

8.93.2.35 pUMTSMinQosSigInd . . . 109
8.93.2.36 pUMTSReqQoS . . . 109
8.93.2.37 pUMTSReqQoSSigInd . . . 109
8.93.2.38 pUsername . . . 109
8.93.2.39 pUsernameSize . . . 109
8.94 LibPackprofile_3GPP2 Struct Reference . . . 109
8.94.1 Detailed Description . . . 110
8.94.2 Field Documentation . . . 113
8.94.2.1 pAllowLinger . . . 113
8.94.2.2 pAPNClass3GPP2 . . . 113
8.94.2.3 pAPNEnabled3GPP2 . . . 113
8.94.2.4 pApnString . . . 113
8.94.2.5 pApnStringSize . . . 113
8.94.2.6 pAppPriority . . . 113
8.94.2.7 pAppType . . . 113
8.94.2.8 pAuthPassword . . . 113
8.94.2.9 pAuthPassword_tSize . . . 113
8.94.2.10 pAuthProtocol . . . 113
8.94.2.11 pAuthRetryCount . . . 113
8.94.2.12 pAuthTimeout . . . 113
8.94.2.13 pDataMode . . . 113
8.94.2.14 pDataRate . . . 113
8.94.2.15 pIpcpAckTimeout . . . 114
8.94.2.16 pIpcpCreqRetryCount . . . 114
8.94.2.17 pIsPcscfAddressNedded . . . 114
8.94.2.18 pLcpAckTimeout . . . 114
8.94.2.19 pLcpCreqRetryCount . . . 114
8.94.2.20 pNegoDnsSrvrPref . . . 114
8.94.2.21 pPDNInactivTimeout3GPP2 . . . 114
8.94.2.22 pPdnType . . . 114
8.94.2.23 pPppSessCloseTimer1x . . . 114
8.94.2.24 pPppSessCloseTimerDO . . . 114
8.94.2.25 pPrimaryV4DnsAddress . . . 114
8.94.2.26 pPriV6DnsAddress . . . 114
8.94.2.27 pRATType . . . 114
8.94.2.28 pSecondaryV4DnsAddress . . . 114
8.94.2.29 pSecV6DnsAddress . . . 114
8.94.2.30 pUserId . . . 114
8.94.2.31 pUserIdSize . . . 114
8.95 LibPackQosClassID Struct Reference . . . 114

8.95.1 Detailed Description . . . 114
8.95.2 Field Documentation . . . 115
8.95.2.1 gDlBitRate . . . 115
8.95.2.2 gUlBitRate . . . 115
8.95.2.3 maxDlBitRate . . . 115
8.95.2.4 maxUlBitRate . . . 115
8.95.2.5 QCI . . . 115
8.96 LibPackTFTIDParams Struct Reference . . . 115
8.96.1 Detailed Description . . . 115
8.96.2 Field Documentation . . . 116
8.96.2.1 destPortRangeEnd . . . 116
8.96.2.2 destPortRangeStart . . . 116
8.96.2.3 eValid . . . 117
8.96.2.4 filterId . . . 117
8.96.2.5 flowLabel . . . 117
8.96.2.6 IPSECSPI . . . 117
8.96.2.7 ipVersion . . . 117
8.96.2.8 nextHeader . . . 117
8.96.2.9 pSourceIP . . . 117
8.96.2.10 sourceIPMask . . . 117
8.96.2.11 srcPortRangeEnd . . . 117
8.96.2.12 srcPortRangeStart . . . 117
8.96.2.13 tosMask . . . 117
8.97 LibPackUMTSQoS Struct Reference . . . 117
8.97.1 Detailed Description . . . 117
8.97.2 Field Documentation . . . 119
8.97.2.1 deliveryErrSDU . . . 119
8.97.2.2 grntDownlinkBitrate . . . 119
8.97.2.3 grntUplinkBitrate . . . 119
8.97.2.4 maxDownlinkBitrate . . . 119
8.97.2.5 maxSDUSize . . . 119
8.97.2.6 maxUplinkBitrate . . . 119
8.97.2.7 qosDeliveryOrder . . . 119
8.97.2.8 resBerRatio . . . 119
8.97.2.9 sduErrorRatio . . . 119
8.97.2.10 trafficClass . . . 119
8.97.2.11 trafficPriority . . . 119
8.97.2.12 transferDelay . . . 119
8.98 LibPackUMTSReqQoSSigInd Struct Reference . . . 120
8.98.1 Detailed Description . . . 120

8.98.2 Field Documentation . . . 120
8.98.2.1 SigInd . . . 120
8.98.2.2 UMTSReqQoS . . . 120
8.99 loc_accelAcceptReady Struct Reference . . . 120
8.99.1 Detailed Description . . . 120
8.99.2 Field Documentation . . . 121
8.99.2.1 batchPerSec . . . 121
8.99.2.2 injectEnable . . . 121
8.99.2.3 samplesPerBatch . . . 121
8.100 loc_accelTempAcceptReady Struct Reference . . . 121
8.100.1 Detailed Description . . . 121
8.100.2 Field Documentation . . . 122
8.100.2.1 batchPerSec . . . 122
8.100.2.2 injectEnable . . . 122
8.100.2.3 samplesPerBatch . . . 122
8.101 loc_BdsSV Struct Reference . . . 122
8.101.1 Detailed Description . . . 122
8.101.2 Field Documentation . . . 122
8.101.2.1 id . . . 122
8.101.2.2 mask . . . 122
8.102 loc_BdsSVInfo Struct Reference . . . 122
8.102.1 Detailed Description . . . 122
8.102.2 Field Documentation . . . 123
8.102.2.1 len . . . 123
8.102.2.2 pSV . . . 123
8.103 loc_CellDb Struct Reference . . . 123
8.103.1 Detailed Description . . . 123
8.103.2 Field Documentation . . . 123
8.103.2.1 mask . . . 123
8.104 loc_ClkInfo Struct Reference . . . 123
8.104.1 Detailed Description . . . 124
8.104.2 Field Documentation . . . 124
8.104.2.1 mask . . . 125
8.105 loc_GnssData Struct Reference . . . 125
8.105.1 Detailed Description . . . 125
8.105.2 Field Documentation . . . 126
8.105.2.1 mask . . . 126
8.106 loc_gpsTime Struct Reference . . . 126
8.106.1 Detailed Description . . . 127
8.106.2 Field Documentation . . . 127

8.106.2.1 gpsTimeOfWeekMs . . . 127
8.106.2.2 gpsWeek . . . 127
8.107loc_gyroAcceptReady Struct Reference . . . 127
8.107.1 Detailed Description . . . 127
8.107.2 Field Documentation . . . 128
8.107.2.1 batchPerSec . . . 128
8.107.2.2 injectEnable . . . 128
8.107.2.3 samplesPerBatch . . . 128
8.108loc_gyroTempAcceptReady Struct Reference . . . 128
8.108.1 Detailed Description . . . 128
8.108.2 Field Documentation . . . 128
8.108.2.1 batchPerSec . . . 128
8.108.2.2 injectEnable . . . 128
8.108.2.3 samplesPerBatch . . . 129
8.109loc_IPv4Config Struct Reference . . . 129
8.109.1 Detailed Description . . . 129
8.109.2 Field Documentation . . . 129
8.109.2.1 IPv4Addr . . . 129
8.109.2.2 IPv4Port . . . 129
8.110loc_IPv4Info Struct Reference . . . 129
8.110.1 Detailed Description . . . 129
8.110.2 Field Documentation . . . 130
8.110.2.1 address . . . 130
8.110.2.2 port . . . 130
8.110.2.3 TlvPresent . . . 130
8.111loc_IPv6Config Struct Reference . . . 130
8.111.1 Detailed Description . . . 130
8.111.2 Field Documentation . . . 130
8.111.2.1 IPv6Addr . . . 130
8.111.2.2 IPv6Port . . . 130
8.112loc_IPv6Info Struct Reference . . . 130
8.112.1 Detailed Description . . . 130
8.112.2 Field Documentation . . . 131
8.112.2.1 address . . . 131
8.112.2.2 port . . . 131
8.112.2.3 TlvPresent . . . 131
8.113loc_LocApplicationInfo Struct Reference . . . 131
8.113.1 Detailed Description . . . 131
8.113.2 Field Documentation . . . 132
8.113.2.1 appNameLength . . . 132

8.113.2.2 appProviderLength . . . 132
8.113.2.3 appVersionLength . . . 132
8.113.2.4 appVersionValid . . . 132
8.113.2.5 pAppName . . . 132
8.113.2.6 pAppProvider . . . 132
8.113.2.7 pAppVersion . . . 132
8.114loc_precisionDilution Struct Reference . . . 132
8.114.1 Detailed Description . . . 132
8.114.2 Field Documentation . . . 133
8.114.2.1 HDOP . . . 133
8.114.2.2 PDOP . . . 133
8.114.2.3 VDOP . . . 133
8.115loc_satelliteInfo Struct Reference . . . 133
8.115.1 Detailed Description . . . 133
8.115.2 Field Documentation . . . 135
8.115.2.1 azimuth . . . 135
8.115.2.2 elevation . . . 135
8.115.2.3 gnssSvId . . . 135
8.115.2.4 healthStatus . . . 135
8.115.2.5 snr . . . 135
8.115.2.6 svInfoMask . . . 135
8.115.2.7 svListLen . . . 135
8.115.2.8 svStatus . . . 135
8.115.2.9 system . . . 135
8.115.2.10validMask . . . 135
8.116loc_sensorDataUsage Struct Reference . . . 135
8.116.1 Detailed Description . . . 135
8.116.2 Field Documentation . . . 136
8.116.2.1 aidingIndicatorMask . . . 136
8.116.2.2 usageMask . . . 136
8.117loc_SV Struct Reference . . . 136
8.117.1 Detailed Description . . . 136
8.117.2 Field Documentation . . . 137
8.117.2.1 id . . . 137
8.117.2.2 mask . . . 137
8.117.2.3 system . . . 137
8.118loc_SVInfo Struct Reference . . . 137
8.118.1 Detailed Description . . . 137
8.118.2 Field Documentation . . . 137
8.118.2.1 len . . . 137

8.118.2.2 pSV . . . 137
8.119 loc_svUsedforFix Struct Reference . . . 137
8.119.1 Detailed Description . . . 137
8.119.2 Field Documentation . . . 138
8.119.2.1 gnssSvUsedList . . . 138
8.119.2.2 gnssSvUsedList_len . . . 138
8.120 loc_urlAddr Struct Reference . . . 138
8.120.1 Detailed Description . . . 138
8.120.2 Field Documentation . . . 138
8.120.2.1 address . . . 138
8.120.2.2 TlvPresent . . . 138
8.121 loc_URLAddrInfo Struct Reference . . . 138
8.121.1 Detailed Description . . . 139
8.121.2 Field Documentation . . . 139
8.121.2.1 urlAddr . . . 139
8.122 lteSSInfo Struct Reference . . . 139
8.122.1 Detailed Description . . . 139
8.122.2 Field Documentation . . . 139
8.122.2.1 rsrp . . . 140
8.122.2.2 rsrq . . . 140
8.122.2.3 rssi . . . 140
8.122.2.4 snr . . . 140
8.123 messageModeTlv Struct Reference . . . 140
8.123.1 Detailed Description . . . 140
8.123.2 Field Documentation . . . 140
8.123.2.1 MessageModeInfo . . . 140
8.123.2.2 TlvPresent . . . 140
8.124 nas_acqOrderPref Struct Reference . . . 140
8.124.1 Detailed Description . . . 140
8.124.2 Field Documentation . . . 141
8.124.2.1 acqOrdeLen . . . 141
8.124.2.2 pAcqOrder . . . 141
8.125 nas_AcqOrderPrefTlv Struct Reference . . . 141
8.125.1 Detailed Description . . . 141
8.125.2 Field Documentation . . . 141
8.125.2.1 acqOrdeLen . . . 141
8.125.2.2 pAcqOrder . . . 142
8.125.2.3 TlvPresent . . . 142
8.126 nas_ActPilotPNElement Struct Reference . . . 142
8.126.1 Detailed Description . . . 142

8.126.2 Field Documentation . . . 142
8.126.2.1 ActSetPilotPN . . . 142
8.126.2.2 ActSetPilotPNStrength . . . 142
8.127 nas_AddCDMASysInfo Struct Reference . . . 142
8.127.1 Detailed Description . . . 142
8.127.2 Field Documentation . . . 143
8.127.2.1 geoSysIdx . . . 143
8.127.2.2 regPrd . . . 143
8.128 nas_AddSysInfo Struct Reference . . . 143
8.128.1 Detailed Description . . . 143
8.128.2 Field Documentation . . . 143
8.128.2.1 cellBroadcastCap . . . 143
8.128.2.2 geoSysIdx . . . 143
8.129 nas_BandPrefTlv Struct Reference . . . 143
8.129.1 Detailed Description . . . 143
8.129.2 Field Documentation . . . 145
8.129.2.1 BandPref . . . 145
8.129.2.2 TlvPresent . . . 145
8.130 nas_CallBarringSysInfo Struct Reference . . . 145
8.130.1 Detailed Description . . . 145
8.130.2 Field Documentation . . . 145
8.130.2.1 csBarStatus . . . 145
8.130.2.2 psBarStatus . . . 145
8.131 nas_callBarStatus Struct Reference . . . 145
8.131.1 Detailed Description . . . 146
8.131.2 Field Documentation . . . 146
8.131.2.1 csBarStatus . . . 146
8.131.2.2 psBarStatus . . . 146
8.132 nas_CDMAChannel Struct Reference . . . 146
8.132.1 Detailed Description . . . 146
8.132.2 Field Documentation . . . 147
8.132.2.1 priChA . . . 147
8.132.2.2 priChB . . . 147
8.132.2.3 secChA . . . 147
8.132.2.4 secChB . . . 147
8.133 nas_CDMAECIOThresh Struct Reference . . . 147
8.133.1 Detailed Description . . . 147
8.133.2 Field Documentation . . . 147
8.133.2.1 CDMAECIOThreshListLen . . . 147
8.133.2.2 pCDMAECIOThreshList . . . 148

8.134 nas_CDMAInfo Struct Reference . . . 148
8.134.1 Detailed Description . . . 148
8.134.2 Field Documentation . . . 148
8.134.2.1 baseId . . . 148
8.134.2.2 baseLat . . . 148
8.134.2.3 baseLong . . . 148
8.134.2.4 nid . . . 149
8.134.2.5 refpn . . . 149
8.134.2.6 sid . . . 149
8.135 nas_CDMARSSIThresh Struct Reference . . . 149
8.135.1 Detailed Description . . . 149
8.135.2 Field Documentation . . . 149
8.135.2.1 CDMARSSIThreshListLen . . . 149
8.135.2.2 pCDMARSSIThreshList . . . 149
8.136 nas_CDMASysInfo Struct Reference . . . 149
8.136.1 Detailed Description . . . 150
8.136.2 Field Documentation . . . 152
8.136.2.1 baseId . . . 152
8.136.2.2 baseLat . . . 152
8.136.2.3 baseLong . . . 152
8.136.2.4 bsInfoValid . . . 152
8.136.2.5 bsPRev . . . 152
8.136.2.6 bsPRevValid . . . 152
8.136.2.7 ccsSupported . . . 152
8.136.2.8 ccsSupportedValid . . . 152
8.136.2.9 cdmaSysIdValid . . . 152
8.136.2.10 isSysPrlMatch . . . 152
8.136.2.11 isSysPrlMatchValid . . . 152
8.136.2.12 MCC . . . 152
8.136.2.13 MNC . . . 152
8.136.2.14 networkID . . . 152
8.136.2.15 networkIdValid . . . 153
8.136.2.16 packetZone . . . 153
8.136.2.17 packetZoneValid . . . 153
8.136.2.18 pRevInUse . . . 153
8.136.2.19 pRevInUseValid . . . 153
8.136.2.20 sysInfoCDMA . . . 153
8.136.2.21 systemID . . . 153
8.137 nas_CDMASysInfoExt Struct Reference . . . 153
8.137.1 Detailed Description . . . 153

8.137.2 Field Documentation . . . 153
8.137.2.1 imsi_11_12 . . . 153
8.137.2.2 MCC . . . 153
8.138 nas_cellParams Struct Reference . . . 153
8.138.1 Detailed Description . . . 154
8.138.2 Field Documentation . . . 154
8.138.2.1 pci . . . 154
8.138.2.2 rsrp . . . 154
8.138.2.3 rsrq . . . 154
8.138.2.4 rssi . . . 154
8.138.2.5 srxlev . . . 154
8.139 nas_ciotAcqOrderPref Struct Reference . . . 154
8.139.1 Detailed Description . . . 154
8.139.2 Field Documentation . . . 155
8.139.2.1 ciotAcqOrderLen . . . 155
8.139.2.2 pCiotAcqOrder . . . 155
8.140 nas_CiotAcqOrderPrefTlv Struct Reference . . . 155
8.140.1 Detailed Description . . . 155
8.140.2 Field Documentation . . . 156
8.140.2.1 ciotAcqOrderLen . . . 156
8.140.2.2 pCiotAcqOrder . . . 156
8.140.2.3 TlvPresent . . . 156
8.141 nas_CiotLteOpModePrefTlv Struct Reference . . . 156
8.141.1 Detailed Description . . . 156
8.141.2 Field Documentation . . . 156
8.141.2.1 ciotLteOpModePref . . . 156
8.141.2.2 TlvPresent . . . 156
8.142 nas_CommInfo Struct Reference . . . 156
8.142.1 Detailed Description . . . 157
8.142.2 Field Documentation . . . 158
8.142.2.1 imsRegState . . . 158
8.142.2.2 modemMode . . . 158
8.142.2.3 psState . . . 158
8.142.2.4 systemMode . . . 158
8.142.2.5 temperature . . . 158
8.143 nas_CSGID Struct Reference . . . 158
8.143.1 Detailed Description . . . 158
8.143.2 Field Documentation . . . 159
8.143.2.1 id . . . 159
8.143.2.2 mcc . . . 159

8.143.2.3 mnc . . . 159
8.143.2.4 mncPcsDigits . . . 159
8.143.2.5 rat . . . 159
8.144 nas_currentPLMN Struct Reference . . . 159
8.144.1 Detailed Description . . . 159
8.144.2 Field Documentation . . . 160
8.144.2.1 MCC . . . 160
8.144.2.2 MNC . . . 160
8.144.2.3 netDescr . . . 160
8.144.2.4 netDescrLength . . . 160
8.145 nas_dataSrvCapabilities Struct Reference . . . 160
8.145.1 Detailed Description . . . 160
8.145.2 Field Documentation . . . 161
8.145.2.1 dataCapabilities . . . 161
8.145.2.2 dataCapabilitiesLen . . . 161
8.146 nas_DataStatusDetail Struct Reference . . . 161
8.146.1 Detailed Description . . . 161
8.146.2 Field Documentation . . . 162
8.146.2.1 IPAddress . . . 162
8.146.2.2 LastErrCode . . . 162
8.147 nas_detailSvcInfo Struct Reference . . . 163
8.147.1 Detailed Description . . . 163
8.147.2 Field Documentation . . . 164
8.147.2.1 hdrHybrid . . . 164
8.147.2.2 hdrSrvStatus . . . 164
8.147.2.3 isSysForbidden . . . 164
8.147.2.4 srvCapability . . . 164
8.147.2.5 srvStatus . . . 164
8.148 nas_DeviceConfigDetail Struct Reference . . . 164
8.148.1 Detailed Description . . . 164
8.148.2 Field Documentation . . . 165
8.148.2.1 Chipset . . . 165
8.148.2.2 HWVersion . . . 165
8.148.2.3 QLIC . . . 165
8.148.2.4 Technology . . . 165
8.149 nas_dirNum Struct Reference . . . 165
8.149.1 Detailed Description . . . 165
8.149.2 Field Documentation . . . 165
8.149.2.1 dirNum . . . 165
8.149.2.2 dirNumLen . . . 165

8.150 nas_DRCParams Struct Reference . . . 165
8.150.1 Detailed Description . . . 166
8.150.2 Field Documentation . . . 166
8.150.2.1 DRCCover . . . 166
8.150.2.2 DRCValue . . . 166
8.151 nas_ecioListElement Struct Reference . . . 166
8.151.1 Detailed Description . . . 166
8.151.2 Field Documentation . . . 166
8.151.2.1 ecio . . . 167
8.151.2.2 radioIf . . . 167
8.152 nas_ECIOThresh Struct Reference . . . 167
8.152.1 Detailed Description . . . 167
8.152.2 Field Documentation . . . 167
8.152.2.1 ECIOThresListLen . . . 167
8.152.2.2 pECIOThresList . . . 167
8.153 nas_EmerModeTlv Struct Reference . . . 167
8.153.1 Detailed Description . . . 167
8.153.2 Field Documentation . . . 168
8.153.2.1 EmerMode . . . 168
8.153.2.2 TlvPresent . . . 168
8.154 nas_errorRateListElement Struct Reference . . . 168
8.154.1 Detailed Description . . . 168
8.154.2 Field Documentation . . . 169
8.154.2.1 errorRate . . . 169
8.154.2.2 radioIf . . . 169
8.155 nas_GERANInfo Struct Reference . . . 169
8.155.1 Detailed Description . . . 169
8.155.2 Field Documentation . . . 170
8.155.2.1 arfcn . . . 170
8.155.2.2 bsic . . . 170
8.155.2.3 cellID . . . 170
8.155.2.4 insNmrCellInfo . . . 170
8.155.2.5 lac . . . 170
8.155.2.6 nmrInst . . . 170
8.155.2.7 plmn . . . 170
8.155.2.8 rxLev . . . 170
8.155.2.9 timingAdvance . . . 170
8.156 nas_geranInstInfo Struct Reference . . . 171
8.156.1 Detailed Description . . . 171
8.156.2 Field Documentation . . . 171

8.156.2.1 geranArfcn . . . 171
8.156.2.2 geranBsicBcc . . . 171
8.156.2.3 geranBsicNcc . . . 171
8.156.2.4 geranRssi . . . 171
8.157 nas_gsmCellInfo Struct Reference . . . 171
8.157.1 Detailed Description . . . 172
8.157.2 Field Documentation . . . 172
8.157.2.1 arfcn . . . 172
8.157.2.2 band1900 . . . 172
8.157.2.3 bsicId . . . 172
8.157.2.4 cellIdValid . . . 172
8.157.2.5 rssi . . . 172
8.157.2.6 srxlev . . . 172
8.158 nas_GSMRSSIThresh Struct Reference . . . 172
8.158.1 Detailed Description . . . 173
8.158.2 Field Documentation . . . 173
8.158.2.1 GSMRSSIThreshListLen . . . 173
8.158.2.2 pGSMRSSIThreshList . . . 173
8.159 nas_GSMSrvStatusInfo Struct Reference . . . 173
8.159.1 Detailed Description . . . 173
8.159.2 Field Documentation . . . 174
8.159.2.1 isPrefDataPath . . . 174
8.159.2.2 srvStatus . . . 174
8.159.2.3 trueSrvStatus . . . 174
8.160 nas_GSMSysInfo Struct Reference . . . 174
8.160.1 Detailed Description . . . 174
8.160.2 Field Documentation . . . 176
8.160.2.1 cellId . . . 176
8.160.2.2 cellIdValid . . . 176
8.160.2.3 dtmSupp . . . 176
8.160.2.4 dtmSuppValid . . . 176
8.160.2.5 egprsSupp . . . 176
8.160.2.6 egprsSuppValid . . . 176
8.160.2.7 lac . . . 176
8.160.2.8 lacValid . . . 176
8.160.2.9 MCC . . . 176
8.160.2.10 MNC . . . 176
8.160.2.11 networkIdValid . . . 176
8.160.2.12 regRejectInfoValid . . . 176
8.160.2.13 rejCause . . . 177

8.160.2.14 rejectSrvDomain . . . 177
8.160.2.15 sysInfoGSM . . . 177
8.161 nas_GWAcqOrderPrefTlv Struct Reference . . . 177
8.161.1 Detailed Description . . . 177
8.161.2 Field Documentation . . . 177
8.161.2.1 GWAcqOrderPref . . . 177
8.161.2.2 TlvPresent . . . 177
8.162 nas_HDRECIOThresh Struct Reference . . . 177
8.162.1 Detailed Description . . . 177
8.162.2 Field Documentation . . . 178
8.162.2.1 HDRECIOThreshListLen . . . 178
8.162.2.2 pHDRECIOThreshList . . . 178
8.163 nas_HDRIOThresh Struct Reference . . . 178
8.163.1 Detailed Description . . . 178
8.163.2 Field Documentation . . . 178
8.163.2.1 HDRIOThreshListLen . . . 178
8.163.2.2 pHDRIOThreshList . . . 178
8.164 nas_HDRPersonality_Ind_Data Struct Reference . . . 178
8.164.1 Detailed Description . . . 179
8.164.2 Field Documentation . . . 179
8.164.2.1 pCurrentPersonality . . . 179
8.164.2.2 pPersonalityListLength . . . 179
8.164.2.3 pProtocolSubtypeElement . . . 179
8.165 nas_HDRRSSIThresh Struct Reference . . . 179
8.165.1 Detailed Description . . . 179
8.165.2 Field Documentation . . . 179
8.165.2.1 HDRRSSIThreshListLen . . . 179
8.165.2.2 pHDRRSSIThreshList . . . 180
8.166 nas_HDRSINRThresh Struct Reference . . . 180
8.166.1 Detailed Description . . . 180
8.166.2 Field Documentation . . . 180
8.166.2.1 HDRSINRThresListLen . . . 180
8.166.2.2 pHDRSINRThresList . . . 180
8.167 nas_HDRSINRThreshold Struct Reference . . . 180
8.167.1 Detailed Description . . . 180
8.167.2 Field Documentation . . . 181
8.167.2.1 HDRSINRThreshListLen . . . 181
8.167.2.2 pHDRSINRThreshList . . . 181
8.168 nas_HDRSysInfo Struct Reference . . . 181
8.168.1 Detailed Description . . . 181

8.168.2 Field Documentation . . . 183
8.168.2.1 hdrActiveProt . . . 183
8.168.2.2 hdrActiveProtValid . . . 183
8.168.2.3 hdrPersonality . . . 183
8.168.2.4 hdrPersonalityValid . . . 183
8.168.2.5 is856SysId . . . 183
8.168.2.6 is856SysIdValid . . . 183
8.168.2.7 isSysPrlMatch . . . 183
8.168.2.8 isSysPrlMatchValid . . . 183
8.168.2.9 sysInfoHDR . . . 183
8.169 nas_homeSIDNID Struct Reference . . . 183
8.169.1 Detailed Description . . . 183
8.169.2 Field Documentation . . . 183
8.169.2.1 numInstances . . . 183
8.169.2.2 SidNid . . . 183
8.170 nas_infoInterFreq Struct Reference . . . 183
8.170.1 Detailed Description . . . 184
8.170.2 Field Documentation . . . 184
8.170.2.1 cell_resel_priority . . . 184
8.170.2.2 cellInterFreqParams . . . 184
8.170.2.3 cells_len . . . 184
8.170.2.4 earfcn . . . 184
8.170.2.5 threshXHigh . . . 185
8.170.2.6 threshXLow . . . 185
8.171 nas_IOThresh Struct Reference . . . 185
8.171.1 Detailed Description . . . 185
8.171.2 Field Documentation . . . 185
8.171.2.1 IOThresListLen . . . 185
8.171.2.2 pIOThresList . . . 185
8.172 nas_lteBandPrefExt Struct Reference . . . 185
8.172.1 Detailed Description . . . 185
8.172.2 Field Documentation . . . 186
8.172.2.1 bits_129_192 . . . 186
8.172.2.2 bits_193_256 . . . 186
8.172.2.3 bits_1_64 . . . 186
8.172.2.4 bits_65_128 . . . 186
8.173 nas_LTEBandPrefTlv Struct Reference . . . 186
8.173.1 Detailed Description . . . 186
8.173.2 Field Documentation . . . 187
8.173.2.1 LTEBandPref . . . 188

8.173.2.2 TlvPresent . . . 188
8.174 nas_lteEARFCN Struct Reference . . . 188
8.174.1 Detailed Description . . . 188
8.174.2 Field Documentation . . . 188
8.174.2.1 earfcn0 . . . 188
8.174.2.2 earfcn1 . . . 188
8.174.2.3 status . . . 188
8.175 nas_lteGsmCellInfo Struct Reference . . . 188
8.175.1 Detailed Description . . . 188
8.175.2 Field Documentation . . . 189
8.175.2.1 cellReselPriority . . . 189
8.175.2.2 cells_len . . . 189
8.175.2.3 GsmCellInfo . . . 189
8.175.2.4 nccPermitted . . . 189
8.175.2.5 threshGsmHigh . . . 189
8.175.2.6 threshGsmLow . . . 189
8.176 nas_LTEInfo Struct Reference . . . 189
8.176.1 Detailed Description . . . 190
8.176.2 Field Documentation . . . 191
8.176.2.1 band . . . 191
8.176.2.2 bandwidth . . . 191
8.176.2.3 emmConnState . . . 191
8.176.2.4 emmState . . . 191
8.176.2.5 emmSubState . . . 191
8.176.2.6 RXChan . . . 191
8.176.2.7 TXChan . . . 191
8.177 nas_LTEInfoInterfreq Struct Reference . . . 192
8.177.1 Detailed Description . . . 192
8.177.2 Field Documentation . . . 192
8.177.2.1 freqsLen . . . 192
8.177.2.2 InfoInterfreq . . . 192
8.177.2.3 ueInIdle . . . 192
8.178 nas_LTEInfoIntrafreq Struct Reference . . . 192
8.178.1 Detailed Description . . . 193
8.178.2 Field Documentation . . . 194
8.178.2.1 CellParams . . . 194
8.178.2.2 cellReselPriority . . . 194
8.178.2.3 cellsLen . . . 194
8.178.2.4 earfcn . . . 194
8.178.2.5 globalCellId . . . 194

8.178.2.6 plmn . . . 194
8.178.2.7 servingCellId . . . 194
8.178.2.8 sIntraSearch . . . 194
8.178.2.9 sNonIntraSearch . . . 194
8.178.2.10 tac . . . 194
8.178.2.11 threshServingLow . . . 194
8.178.2.12 ueInIdle . . . 194
8.179 nas_LTEInfoNeighboringGSM Struct Reference . . . 194
8.179.1 Detailed Description . . . 195
8.179.2 Field Documentation . . . 195
8.179.2.1 freqsLen . . . 195
8.179.2.2 LteGsmCellInfo . . . 195
8.179.2.3 ueInIdle . . . 195
8.180 nas_LTEInfoNeighboringWCDMA Struct Reference . . . 195
8.180.1 Detailed Description . . . 195
8.180.2 Field Documentation . . . 196
8.180.2.1 freqsLen . . . 196
8.180.2.2 LTEWCDMACellInfo . . . 196
8.180.2.3 ueInIdle . . . 196
8.181 nas_LteM1BandPrefTlv Struct Reference . . . 196
8.181.1 Detailed Description . . . 196
8.181.2 Field Documentation . . . 197
8.181.2.1 lteM1BandPref . . . 198
8.181.2.2 TlvPresent . . . 198
8.182 nas_LteNb1BandPrefTlv Struct Reference . . . 198
8.182.1 Detailed Description . . . 198
8.182.2 Field Documentation . . . 199
8.182.2.1 lteNb1BandPref . . . 200
8.182.2.2 TlvPresent . . . 200
8.183 nas_LTEOperationMode Struct Reference . . . 200
8.183.1 Detailed Description . . . 200
8.183.2 Field Documentation . . . 200
8.183.2.1 pLTEOperationMode . . . 200
8.183.2.2 TlvPresent . . . 200
8.184 nas_ltePCI Struct Reference . . . 200
8.184.1 Detailed Description . . . 200
8.184.2 Field Documentation . . . 201
8.184.2.1 earfcn . . . 201
8.184.2.2 pci . . . 201
8.184.2.3 status . . . 201

8.185 nas_lteRsrpinformation Struct Reference . . . 201
8.185.1 Detailed Description . . . 201
8.185.2 Field Documentation . . . 201
8.185.2.1 rsrplevel . . . 201
8.186 nas_LTERSRPThresh Struct Reference . . . 201
8.186.1 Detailed Description . . . 201
8.186.2 Field Documentation . . . 201
8.186.2.1 LTERSRPThreshListLen . . . 202
8.186.2.2 pLTERSRPThreshList . . . 202
8.187 nas_LTERSRQThresh Struct Reference . . . 202
8.187.1 Detailed Description . . . 202
8.187.2 Field Documentation . . . 202
8.187.2.1 LTERSRQThreshListLen . . . 202
8.187.2.2 pLTERSRQThreshList . . . 202
8.188 nas_LTERSSIThresh Struct Reference . . . 202
8.188.1 Detailed Description . . . 202
8.188.2 Field Documentation . . . 203
8.188.2.1 LTERSSIThreshListLen . . . 203
8.188.2.2 pLTERSSIThreshList . . . 203
8.189 nas_LTESigRptCfg Struct Reference . . . 203
8.189.1 Detailed Description . . . 203
8.189.2 Field Documentation . . . 203
8.189.2.1 avgPeriod . . . 203
8.189.2.2 rptRate . . . 203
8.190 nas_LTESigRptConfig Struct Reference . . . 204
8.190.1 Detailed Description . . . 204
8.190.2 Field Documentation . . . 204
8.190.2.1 avgPeriod . . . 204
8.190.2.2 rptRate . . . 204
8.191 nas_lteSnrinformation Struct Reference . . . 204
8.191.1 Detailed Description . . . 205
8.191.2 Field Documentation . . . 205
8.191.2.1 snrlevel . . . 205
8.192 nas_LTESNRThresh Struct Reference . . . 205
8.192.1 Detailed Description . . . 205
8.192.2 Field Documentation . . . 205
8.192.2.1 LTESNRThresListLen . . . 205
8.192.2.2 pLTESNRThresList . . . 205
8.193 nas_LTESNRThreshold Struct Reference . . . 205
8.193.1 Detailed Description . . . 206

8.193.2 Field Documentation . . . 206
8.193.2.1 LTESNRThreshListLen . . . 206
8.193.2.2 pLTESNRThreshList . . . 206
8.194 nas_LTESysInfo Struct Reference . . . 206
8.194.1 Detailed Description . . . 206
8.194.2 Field Documentation . . . 208
8.194.2.1 cellId . . . 208
8.194.2.2 cellIdValid . . . 208
8.194.2.3 lac . . . 208
8.194.2.4 lacValid . . . 208
8.194.2.5 MCC . . . 208
8.194.2.6 MNC . . . 208
8.194.2.7 networkIdValid . . . 208
8.194.2.8 regRejectInfoValid . . . 208
8.194.2.9 rejCause . . . 208
8.194.2.10 rejectSrvDomain . . . 208
8.194.2.11 sysInfoLTE . . . 208
8.194.2.12 tac . . . 208
8.194.2.13 tacValid . . . 208
8.195 nas_lteWcdmaCellInfo Struct Reference . . . 208
8.195.1 Detailed Description . . . 209
8.195.2 Field Documentation . . . 209
8.195.2.1 cellReselPriority . . . 209
8.195.2.2 cellsLen . . . 209
8.195.2.3 threshXhigh . . . 209
8.195.2.4 threshXlow . . . 209
8.195.2.5 uarfcn . . . 209
8.195.2.6 WCDMACellInfo . . . 209
8.196 nas_minBasedIMSI Struct Reference . . . 209
8.196.1 Detailed Description . . . 210
8.196.2 Field Documentation . . . 210
8.196.2.1 imsiM1112 . . . 210
8.196.2.2 imsiMS1 . . . 210
8.196.2.3 imsiMS2 . . . 210
8.196.2.4 mccM . . . 210
8.197 nas_MNRInfo Struct Reference . . . 210
8.197.1 Detailed Description . . . 210
8.197.2 Field Documentation . . . 211
8.197.2.1 mcc . . . 211
8.197.2.2 mnc . . . 211

8.197.2.3 rat . . . 211
8.198 nas_ModePrefTlv Struct Reference . . . 211
8.198.1 Detailed Description . . . 211
8.198.2 Field Documentation . . . 211
8.198.2.1 ModePref . . . 211
8.198.2.2 TlvPresent . . . 212
8.199 nas_namName Struct Reference . . . 212
8.199.1 Detailed Description . . . 212
8.199.2 Field Documentation . . . 212
8.199.2.1 namName . . . 212
8.199.2.2 namNameLen . . . 212
8.200 nas_netSelectionPref Struct Reference . . . 212
8.200.1 Detailed Description . . . 212
8.200.2 Field Documentation . . . 213
8.200.2.1 mcc . . . 213
8.200.2.2 mnc . . . 213
8.200.2.3 netReg . . . 213
8.201 nas_NetSelPrefTlv Struct Reference . . . 213
8.201.1 Detailed Description . . . 213
8.201.2 Field Documentation . . . 213
8.201.2.1 NetSelPref . . . 213
8.201.2.2 TlvPresent . . . 213
8.202 nas_NetworkStat1x Struct Reference . . . 213
8.202.1 Detailed Description . . . 214
8.202.2 Field Documentation . . . 215
8.202.2.1 ActSetCnt . . . 215
8.202.2.2 NeighborSetCnt . . . 215
8.202.2.3 pActPilotPNElements . . . 215
8.202.2.4 pNeighborSetPilotPN . . . 215
8.202.2.5 RX_EC_IO . . . 215
8.202.2.6 RX_PWR . . . 215
8.202.2.7 SO . . . 215
8.202.2.8 State . . . 215
8.202.2.9 TX_PWR . . . 216
8.203 nas_NetworkStatEVDO Struct Reference . . . 216
8.203.1 Detailed Description . . . 216
8.203.2 Field Documentation . . . 217
8.203.2.1 MACIndex . . . 217
8.203.2.2 PER . . . 217
8.203.2.3 PilotEnergy . . . 217

8.203.2.4 pSectorID . . . 217
8.203.2.5 RX_PWR . . . 217
8.203.2.6 SectorIDLen . . . 217
8.203.2.7 SNR . . . 217
8.203.2.8 State . . . 217
8.204 nas_nmrCellInfo Struct Reference . . . 217
8.204.1 Detailed Description . . . 217
8.204.2 Field Documentation . . . 218
8.204.2.1 nmrArfcn . . . 218
8.204.2.2 nmrBsic . . . 218
8.204.2.3 nmrCellID . . . 218
8.204.2.4 nmrLac . . . 218
8.204.2.5 nmrPlmn . . . 218
8.204.2.6 nmrRxLev . . . 218
8.205 nas_nr5gBandPref Struct Reference . . . 218
8.205.1 Detailed Description . . . 219
8.205.2 Field Documentation . . . 219
8.205.2.1 bits_129_192 . . . 219
8.205.2.2 bits_193_256 . . . 219
8.205.2.3 bits_1_64 . . . 219
8.205.2.4 bits_65_128 . . . 219
8.206 nas_NR5gBandPrefTlv Struct Reference . . . 219
8.206.1 Detailed Description . . . 219
8.206.2 Field Documentation . . . 220
8.206.2.1 bits_129_192 . . . 220
8.206.2.2 bits_193_256 . . . 220
8.206.2.3 bits_1_64 . . . 220
8.206.2.4 bits_65_128 . . . 220
8.206.2.5 TlvPresent . . . 220
8.207 nas_operatorNameString Struct Reference . . . 220
8.207.1 Detailed Description . . . 220
8.207.2 Field Documentation . . . 220
8.207.2.1 PLMNName . . . 220
8.208 nas_OperatorPLMNData Struct Reference . . . 220
8.208.1 Detailed Description . . . 221
8.208.2 Field Documentation . . . 221
8.208.2.1 lac1 . . . 221
8.208.2.2 lac2 . . . 221
8.208.2.3 mcc . . . 221
8.208.2.4 mnc . . . 221

8.208.2.5 PLMNRecID . . . 221
8.209 nas_operatorPLMNList Struct Reference . . . 221
8.209.1 Detailed Description . . . 222
8.209.2 Field Documentation . . . 222
8.209.2.1 numInstance . . . 222
8.209.2.2 PLMNData . . . 222
8.210 nas_PhyCaAggPcellInfo Struct Reference . . . 222
8.210.1 Detailed Description . . . 222
8.210.2 Field Documentation . . . 223
8.210.2.1 dl_bw_value . . . 223
8.210.2.2 freq . . . 223
8.210.2.3 iLTEbandValue . . . 223
8.210.2.4 pci . . . 223
8.210.2.5 TlvPresent . . . 223
8.211 nas_PhyCaAggScellDlBw Struct Reference . . . 223
8.211.1 Detailed Description . . . 223
8.211.2 Field Documentation . . . 223
8.211.2.1 dl_bw_value . . . 223
8.211.2.2 TlvPresent . . . 223
8.212 nas_PhyCaAggScellIndex Struct Reference . . . 223
8.212.1 Detailed Description . . . 223
8.212.2 Field Documentation . . . 224
8.212.2.1 scell_idx . . . 224
8.212.2.2 TlvPresent . . . 224
8.213 nas_PhyCaAggScellIndType Struct Reference . . . 224
8.213.1 Detailed Description . . . 224
8.213.2 Field Documentation . . . 224
8.213.2.1 freq . . . 224
8.213.2.2 pci . . . 224
8.213.2.3 scell_state . . . 224
8.213.2.4 TlvPresent . . . 224
8.214 nas_PhyCaAggScellInfo Struct Reference . . . 225
8.214.1 Detailed Description . . . 225
8.214.2 Field Documentation . . . 227
8.214.2.1 dl_bw_value . . . 227
8.214.2.2 freq . . . 227
8.214.2.3 iLTEbandValue . . . 227
8.214.2.4 pci . . . 227
8.214.2.5 scell_state . . . 227
8.214.2.6 TlvPresent . . . 227

8.215 nas_PilotSetData Struct Reference . . . 227
8.215.1 Detailed Description . . . 227
8.215.2 Field Documentation . . . 227
8.215.2.1 NumPilots . . . 227
8.215.2.2 pPilotSetInfo . . . 227
8.216 nas_PilotSetParams Struct Reference . . . 228
8.216.1 Detailed Description . . . 228
8.216.2 Field Documentation . . . 228
8.216.2.1 PilotPN . . . 228
8.216.2.2 PilotStrength . . . 228
8.216.2.3 PilotType . . . 228
8.217 nas_PLMNNetworkName Struct Reference . . . 228
8.217.1 Detailed Description . . . 228
8.217.2 Field Documentation . . . 229
8.217.2.1 numInstance . . . 229
8.217.2.2 PLMNNetName . . . 229
8.218 nas_PLMNNetworkNameData Struct Reference . . . 229
8.218.1 Detailed Description . . . 229
8.218.2 Field Documentation . . . 230
8.218.2.1 codingScheme . . . 230
8.218.2.2 countryInitials . . . 230
8.218.2.3 longName . . . 230
8.218.2.4 longNameLen . . . 230
8.218.2.5 longNameSpareBits . . . 230
8.218.2.6 shortName . . . 230
8.218.2.7 shortNameLen . . . 230
8.218.2.8 shortNameSpareBits . . . 230
8.219 nas_PRLPrefTlv Struct Reference . . . 230
8.219.1 Detailed Description . . . 231
8.219.2 Field Documentation . . . 231
8.219.2.1 PRLPref . . . 231
8.219.2.2 TlvPresent . . . 231
8.220 nas_protocolSubtypeElement Struct Reference . . . 231
8.220.1 Detailed Description . . . 231
8.220.2 Field Documentation . . . 232
8.220.2.1 AccessMac . . . 232
8.220.2.2 AuthProt . . . 232
8.220.2.3 ControlMac . . . 232
8.220.2.4 EncryptProt . . . 232
8.220.2.5 ForwardMac . . . 232

8.220.2.6 IdleState . . . 232
8.220.2.7 KeyExchange . . . 232
8.220.2.8 MultDisc . . . 232
8.220.2.9 PhysicalLayer . . . 232
8.220.2.10 ReverseMac . . . 232
8.220.2.11 SecProt . . . 232
8.220.2.12 VirtStream . . . 232
8.221 nas_qaQmi3Gpp2TimeZone Struct Reference . . . 233
8.221.1 Detailed Description . . . 233
8.221.2 Field Documentation . . . 233
8.221.2.1 daylightSavings . . . 233
8.221.2.2 leapSeconds . . . 233
8.221.2.3 localTimeOffset . . . 233
8.222 nas_QmiNas3GppNetworkInfo Struct Reference . . . 233
8.222.1 Detailed Description . . . 233
8.222.2 Field Documentation . . . 234
8.222.2.1 Desription . . . 234
8.222.2.2 Forbidden . . . 234
8.222.2.3 InUse . . . 234
8.222.2.4 MCC . . . 234
8.222.2.5 MNC . . . 234
8.222.2.6 Preferred . . . 234
8.222.2.7 Roaming . . . 234
8.223 nas_QmiNas3GppNetworkRAT Struct Reference . . . 235
8.223.1 Detailed Description . . . 235
8.223.2 Field Documentation . . . 235
8.223.2.1 MCC . . . 235
8.223.2.2 MNC . . . 235
8.223.2.3 RAT . . . 235
8.224 nas_QmisNasPcsDigit Struct Reference . . . 235
8.224.1 Detailed Description . . . 235
8.224.2 Field Documentation . . . 236
8.224.2.1 includes_pcs_digit . . . 236
8.224.2.2 MCC . . . 236
8.224.2.3 MNC . . . 236
8.225 nas_RankIndicatorTlv Struct Reference . . . 236
8.225.1 Detailed Description . . . 236
8.225.2 Field Documentation . . . 236
8.225.2.1 count0 . . . 236
8.225.2.2 count1 . . . 236

8.225.2.3 TlvPresent . . . 236
8.226 nas_RatDisabledMaskTlv Struct Reference . . . 236
8.226.1 Detailed Description . . . 237
8.226.2 Field Documentation . . . 237
8.226.2.1 ratDisabledMask . . . 237
8.226.2.2 TlvPresent . . . 237
8.227 nas_RejectReasonTlv Struct Reference . . . 237
8.227.1 Detailed Description . . . 237
8.227.2 Field Documentation . . . 237
8.227.2.1 rejectCause . . . 238
8.227.2.2 serviceDomain . . . 238
8.227.2.3 TlvPresent . . . 238
8.228 nas_RFBandInfoElements Struct Reference . . . 238
8.228.1 Detailed Description . . . 238
8.228.2 Field Documentation . . . 238
8.228.2.1 activeBandClass . . . 238
8.228.2.2 activeChannel . . . 238
8.228.2.3 radioInterface . . . 238
8.229 nas_RfBandInfoExtFormat Struct Reference . . . 238
8.229.1 Detailed Description . . . 238
8.229.2 Field Documentation . . . 239
8.229.2.1 pInstancesSize . . . 239
8.229.2.2 pRfBandInfoExtFormatParam . . . 239
8.229.2.3 TlvPresent . . . 239
8.230 nas_RfBandInfoExtFormatElements Struct Reference . . . 239
8.230.1 Detailed Description . . . 239
8.230.2 Field Documentation . . . 239
8.230.2.1 activeBand . . . 239
8.230.2.2 activeChannel . . . 239
8.230.2.3 radioInterface . . . 240
8.231 nas_RfBandInfoList Struct Reference . . . 240
8.231.1 Detailed Description . . . 240
8.231.2 Field Documentation . . . 240
8.231.2.1 pInstanceSize . . . 240
8.231.2.2 pRFBandInfoParam . . . 240
8.232 nas_RfBandwidthInfo Struct Reference . . . 240
8.232.1 Detailed Description . . . 240
8.232.2 Field Documentation . . . 241
8.232.2.1 pInstancesSize . . . 241
8.232.2.2 pRfBandwidthInfoParam . . . 241

8.232.2.3 TlvPresent . . . 241
8.233 nas_RfBandwidthInfoElements Struct Reference . . . 241
8.233.1 Detailed Description . . . 241
8.233.2 Field Documentation . . . 241
8.233.2.1 bandwidth . . . 241
8.233.2.2 radioInterface . . . 242
8.234 nas_RfDedicatedBandInfo Struct Reference . . . 242
8.234.1 Detailed Description . . . 242
8.234.2 Field Documentation . . . 242
8.234.2.1 pInstancesSize . . . 242
8.234.2.2 pRfDedicatedBandInfoParam . . . 242
8.234.2.3 TlvPresent . . . 242
8.235 nas_RfDedicatedBandInfoElements Struct Reference . . . 242
8.235.1 Detailed Description . . . 242
8.235.2 Field Documentation . . . 243
8.235.2.1 dedicatedBand . . . 243
8.235.2.2 radioInterface . . . 243
8.236 nas_RFInfoTlv Struct Reference . . . 243
8.236.1 Detailed Description . . . 243
8.236.2 Field Documentation . . . 244
8.236.2.1 activeBandClass . . . 244
8.236.2.2 activeChannel . . . 244
8.236.2.3 radioInterface . . . 244
8.236.2.4 radioInterfaceSize . . . 244
8.236.2.5 TlvPresent . . . 244
8.237 nas_roamIndList Struct Reference . . . 244
8.237.1 Detailed Description . . . 244
8.237.2 Field Documentation . . . 244
8.237.2.1 numInstances . . . 245
8.237.2.2 radioInterface . . . 245
8.237.2.3 roamIndicator . . . 245
8.238 nas_RoamPrefTlv Struct Reference . . . 245
8.238.1 Detailed Description . . . 245
8.238.2 Field Documentation . . . 245
8.238.2.1 RoamPref . . . 245
8.238.2.2 TlvPresent . . . 245
8.239 nas_RSRPThresh Struct Reference . . . 245
8.239.1 Detailed Description . . . 245
8.239.2 Field Documentation . . . 246
8.239.2.1 pRSRPThresList . . . 246

8.239.2.2 RSRPThresListLen . . . 246
8.240 nas_rsrqInformation Struct Reference . . . 246
8.240.1 Detailed Description . . . 246
8.240.2 Field Documentation . . . 246
8.240.2.1 radioIf . . . 246
8.240.2.2 rsrq . . . 246
8.241 nas_RSRQThresh Struct Reference . . . 246
8.241.1 Detailed Description . . . 247
8.241.2 Field Documentation . . . 247
8.241.2.1 pRSRQThresList . . . 247
8.241.2.2 RSRQThresListLen . . . 247
8.242 nas_RSSIThresh Struct Reference . . . 247
8.242.1 Detailed Description . . . 247
8.242.2 Field Documentation . . . 248
8.242.2.1 pRSSIThresList . . . 248
8.242.2.2 RSSIThresListLen . . . 248
8.243 nas_rxInfo Struct Reference . . . 248
8.243.1 Detailed Description . . . 248
8.243.2 Field Documentation . . . 249
8.243.2.1 ecio . . . 249
8.243.2.2 isRadioTuned . . . 249
8.243.2.3 phase . . . 249
8.243.2.4 rscp . . . 249
8.243.2.5 rsrp . . . 249
8.243.2.6 rxPower . . . 249
8.244 nas_RxSigInfo Struct Reference . . . 249
8.244.1 Detailed Description . . . 249
8.244.2 Field Documentation . . . 250
8.244.2.1 isRadioTuned . . . 250
8.244.2.2 rsrp . . . 250
8.244.2.3 rxChainIndex . . . 250
8.244.2.4 rxPower . . . 250
8.245 nas_rxSignalStrengthListElement Struct Reference . . . 250
8.245.1 Detailed Description . . . 250
8.245.2 Field Documentation . . . 251
8.245.2.1 radioIf . . . 251
8.245.2.2 rxSignalStrength . . . 251
8.246 nas_SccRxInfo Struct Reference . . . 251
8.246.1 Detailed Description . . . 251
8.246.2 Field Documentation . . . 252

8.246.2.1 numInstances . . . 252
8.246.2.2 rsrq . . . 252
8.246.2.3 sigInfo . . . 252
8.246.2.4 snr . . . 252
8.246.2.5 TlvPresent . . . 252
8.247 nas_serviceProviderName Struct Reference . . . 252
8.247.1 Detailed Description . . . 252
8.247.2 Field Documentation . . . 252
8.247.2.1 displayCondition . . . 252
8.247.2.2 spn . . . 252
8.247.2.3 spnLength . . . 252
8.248 nas_servSystem Struct Reference . . . 253
8.248.1 Detailed Description . . . 253
8.248.2 Field Documentation . . . 254
8.248.2.1 csAttachState . . . 254
8.248.2.2 numRadioInterfaces . . . 254
8.248.2.3 psAttachState . . . 254
8.248.2.4 radioInterface . . . 254
8.248.2.5 regState . . . 254
8.248.2.6 selNetwork . . . 254
8.249 nas_sidNid Struct Reference . . . 254
8.249.1 Detailed Description . . . 254
8.249.2 Field Documentation . . . 254
8.249.2.1 nid . . . 254
8.249.2.2 sid . . . 254
8.250 nas_SignalStrengthTlv Struct Reference . . . 255
8.250.1 Detailed Description . . . 255
8.250.2 Field Documentation . . . 255
8.250.2.1 radioInterface . . . 255
8.250.2.2 signalStrength . . . 255
8.250.2.3 TlvPresent . . . 255
8.251 nas_SLQSSignalStrengthsIndReq Struct Reference . . . 255
8.251.1 Detailed Description . . . 255
8.251.2 Field Documentation . . . 256
8.251.2.1 ecioDelta . . . 256
8.251.2.2 ecioThresholdList . . . 256
8.251.2.3 ecioThresholdListLen . . . 257
8.251.2.4 ioDelta . . . 257
8.251.2.5 lteRsrpDelta . . . 257
8.251.2.6 lteSnrDelta . . . 257

8.251.2.7 rsrqDelta . . . 257
8.251.2.8 rxSignalStrengthDelta . . . 257
8.251.2.9 sinrDelta . . . 257
8.251.2.10 sinrThresholdList . . . 257
8.251.2.11 sinrThresholdListLen . . . 257
8.252 nas_SLQSSignalStrengthsInformation Struct Reference . . . 257
8.252.1 Detailed Description . . . 257
8.252.2 Field Documentation . . . 258
8.252.2.1 ecioInfo . . . 258
8.252.2.2 errorRateInfo . . . 258
8.252.2.3 io . . . 258
8.252.2.4 lteRsrpinfo . . . 258
8.252.2.5 lteSnrinfo . . . 258
8.252.2.6 rsrqInfo . . . 258
8.252.2.7 rxSignalStrengthInfo . . . 258
8.252.2.8 sinr . . . 258
8.253 nas_SLQSSignalStrengthsTlv Struct Reference . . . 258
8.253.1 Detailed Description . . . 258
8.253.2 Field Documentation . . . 259
8.253.2.1 sSLQSSignalStrengthsInfo . . . 259
8.253.2.2 TlvPresent . . . 259
8.254 nas_SrvDomainPrefTlv Struct Reference . . . 259
8.254.1 Detailed Description . . . 259
8.254.2 Field Documentation . . . 259
8.254.2.1 SrvDomainPref . . . 259
8.254.2.2 TlvPresent . . . 259
8.255 nas_SrvStatusInfo Struct Reference . . . 259
8.255.1 Detailed Description . . . 259
8.255.2 Field Documentation . . . 260
8.255.2.1 isPrefDataPath . . . 260
8.255.2.2 srvStatus . . . 260
8.256 nas_sysInfoCommon Struct Reference . . . 260
8.256.1 Detailed Description . . . 260
8.256.2 Field Documentation . . . 262
8.256.2.1 isSysForbidden . . . 262
8.256.2.2 isSysForbiddenValid . . . 262
8.256.2.3 roamStatus . . . 262
8.256.2.4 roamStatusValid . . . 262
8.256.2.5 srvCapability . . . 262
8.256.2.6 srvCapabilityValid . . . 262

8.256.2.7 srvDomain . . . 262
8.256.2.8 srvDomainValid . . . 262
8.257 nas_TDSCDMAECIOThresh Struct Reference . . . 263
8.257.1 Detailed Description . . . 263
8.257.2 Field Documentation . . . 263
8.257.2.1 pTDSCDMAECIOThreshList . . . 263
8.257.2.2 TDSCDMAECIOThreshListLen . . . 263
8.258 nas_TDSCDMARSCPThresh Struct Reference . . . 263
8.258.1 Detailed Description . . . 263
8.258.2 Field Documentation . . . 263
8.258.2.1 pTDSCDMARSCPThreshList . . . 263
8.258.2.2 TDSCDMARSCPThreshListLen . . . 264
8.259 nas_TDSCDMARSSIThresh Struct Reference . . . 264
8.259.1 Detailed Description . . . 264
8.259.2 Field Documentation . . . 264
8.259.2.1 pTDSCDMARSSIThreshList . . . 264
8.259.2.2 TDSCDMARSSIThreshListLen . . . 264
8.260 nas_TDSCDMASINRCONFThresh Struct Reference . . . 264
8.260.1 Detailed Description . . . 264
8.260.2 Field Documentation . . . 264
8.260.2.1 pTDSCDMASINRCONFThreshList . . . 265
8.260.2.2 TDSCDMASINRCONFThreshListLen . . . 265
8.261 nas_TDSCDMASINRThresh Struct Reference . . . 265
8.261.1 Detailed Description . . . 265
8.261.2 Field Documentation . . . 265
8.261.2.1 pTDSCDMASINRThreshList . . . 265
8.261.2.2 TDSCDMASINRThreshListLen . . . 265
8.262 nas_timeInfo Struct Reference . . . 265
8.262.1 Detailed Description . . . 265
8.262.2 Field Documentation . . . 267
8.262.2.1 day . . . 267
8.262.2.2 dayLtSavingAdj . . . 267
8.262.2.3 dayOfWeek . . . 267
8.262.2.4 hour . . . 267
8.262.2.5 minute . . . 267
8.262.2.6 month . . . 267
8.262.2.7 radioInterface . . . 267
8.262.2.8 second . . . 267
8.262.2.9 timeZone . . . 267
8.262.2.10 TlvPresent . . . 267

8.262.2.11 year . . . 267
8.263 nas_trueIMSI Struct Reference . . . 267
8.263.1 Detailed Description . . . 267
8.263.2 Field Documentation . . . 268
8.263.2.1 imsiT1112 . . . 268
8.263.2.2 imsiTaddrNum . . . 268
8.263.2.3 imsiTS1 . . . 268
8.263.2.4 imsiTS2 . . . 268
8.263.2.5 mccT . . . 268
8.264 nas_txInfo Struct Reference . . . 268
8.264.1 Detailed Description . . . 268
8.264.2 Field Documentation . . . 268
8.264.2.1 isInTraffic . . . 268
8.264.2.2 txPower . . . 268
8.265 nas_UMTSInfo Struct Reference . . . 268
8.265.1 Detailed Description . . . 269
8.265.2 Field Documentation . . . 270
8.265.2.1 cellID . . . 270
8.265.2.2 ecio . . . 270
8.265.2.3 geranInst . . . 270
8.265.2.4 GeranInstInfo . . . 270
8.265.2.5 lac . . . 270
8.265.2.6 plmn . . . 270
8.265.2.7 psc . . . 270
8.265.2.8 rscp . . . 270
8.265.2.9 uarfcn . . . 270
8.265.2.10 umtsInst . . . 270
8.265.2.11 UMTSInstInfo . . . 270
8.266 nas_UMTSinstInfo Struct Reference . . . 270
8.266.1 Detailed Description . . . 270
8.266.2 Field Documentation . . . 271
8.266.2.1 umtsEcio . . . 271
8.266.2.2 umtsPsc . . . 271
8.266.2.3 umtsRscp . . . 271
8.266.2.4 umtsUarfcn . . . 271
8.267 nas_umtsLTENbrCell Struct Reference . . . 271
8.267.1 Detailed Description . . . 271
8.267.2 Field Documentation . . . 272
8.267.2.1 cellIsTDD . . . 272
8.267.2.2 earfcn . . . 272

8.267.2.3 pci . . . 272
8.267.2.4 rsrp . . . 272
8.267.2.5 rsrq . . . 272
8.267.2.6 srxlev . . . 272
8.268 nas_UniversalTime Struct Reference . . . 272
8.268.1 Detailed Description . . . 272
8.268.2 Field Documentation . . . 273
8.268.2.1 day . . . 273
8.268.2.2 dayOfWeek . . . 273
8.268.2.3 hour . . . 273
8.268.2.4 minute . . . 273
8.268.2.5 month . . . 273
8.268.2.6 second . . . 273
8.268.2.7 year . . . 273
8.269 nas_wcdmaCellInfo Struct Reference . . . 273
8.269.1 Detailed Description . . . 273
8.269.2 Field Documentation . . . 274
8.269.2.1 cpich_ecno . . . 274
8.269.2.2 cpich_rscp . . . 274
8.269.2.3 psc . . . 274
8.269.2.4 srxlev . . . 274
8.270 nas_WCDMAECIOThresh Struct Reference . . . 274
8.270.1 Detailed Description . . . 274
8.270.2 Field Documentation . . . 274
8.270.2.1 pWCDMAECIOThreshList . . . 274
8.270.2.2 WCDMAECIOThreshListLen . . . 274
8.271 nas_WCDMAInfoLTENeighborCell Struct Reference . . . 274
8.271.1 Detailed Description . . . 275
8.271.2 Field Documentation . . . 275
8.271.2.1 UMTSLTENbrCell . . . 275
8.271.2.2 umtsLTENbrCellLen . . . 275
8.271.2.3 wcdmaRRCState . . . 275
8.272 nas_WCDMARSSIThresh Struct Reference . . . 275
8.272.1 Detailed Description . . . 275
8.272.2 Field Documentation . . . 276
8.272.2.1 pWCDMARSSIThreshList . . . 276
8.272.2.2 WCDMARSSIThreshListLen . . . 276
8.273 nas_WCDMASysInfo Struct Reference . . . 276
8.273.1 Detailed Description . . . 276
8.273.2 Field Documentation . . . 278

8.273.2.1 cellId . . . 279
8.273.2.2 cellIdValid . . . 279
8.273.2.3 hsCallStatus . . . 279
8.273.2.4 hsCallStatusValid . . . 279
8.273.2.5 hsInd . . . 279
8.273.2.6 hsIndValid . . . 279
8.273.2.7 lac . . . 279
8.273.2.8 lacValid . . . 279
8.273.2.9 MCC . . . 279
8.273.2.10 MNC . . . 279
8.273.2.11 networkIdValid . . . 279
8.273.2.12 psc . . . 279
8.273.2.13 pscValid . . . 279
8.273.2.14 regRejectInfoValid . . . 279
8.273.2.15 rejCause . . . 279
8.273.2.16 rejectSrvDomain . . . 279
8.273.2.17 sysInfoWCDMA . . . 279
8.274 nas_wcdmaUARFCN Struct Reference . . . 279
8.274.1 Detailed Description . . . 279
8.274.2 Field Documentation . . . 280
8.274.2.1 status . . . 280
8.274.2.2 uarfcn . . . 280
8.275 NASBandPreferenceTlv Struct Reference . . . 280
8.275.1 Detailed Description . . . 280
8.275.2 Field Documentation . . . 280
8.275.2.1 band_pref . . . 280
8.275.2.2 TlvPresent . . . 280
8.276 NASEmergencyModeTlv Struct Reference . . . 280
8.276.1 Detailed Description . . . 280
8.276.2 Field Documentation . . . 280
8.276.2.1 EmerMode . . . 280
8.276.2.2 TlvPresent . . . 280
8.277 NasGetLTECphyCaInfo Struct Reference . . . 280
8.277.1 Detailed Description . . . 281
8.277.2 Field Documentation . . . 281
8.277.2.1 PhyCaAggPcellInfo . . . 281
8.277.2.2 PhyCaAggScellDlBw . . . 281
8.277.2.3 PhyCaAggScellIndex . . . 281
8.277.2.4 PhyCaAggScellIndType . . . 281
8.277.2.5 PhyCaAggScellInfo . . . 281

8.278 NASGWAcqOrderPrefTlv Struct Reference . . . 281
8.278.1 Detailed Description . . . 282
8.278.2 Field Documentation . . . 282
8.278.2.1 GWAcqOrderPref . . . 282
8.278.2.2 TlvPresent . . . 282
8.279 NASLTEBandPreferenceTlv Struct Reference . . . 282
8.279.1 Detailed Description . . . 282
8.279.2 Field Documentation . . . 282
8.279.2.1 LTEBandPref . . . 282
8.279.2.2 TlvPresent . . . 282
8.280 NASLteNasReleaseInfoTlv Struct Reference . . . 282
8.280.1 Detailed Description . . . 282
8.280.2 Field Documentation . . . 283
8.280.2.1 nas_major . . . 283
8.280.2.2 nas_minor . . . 283
8.280.2.3 nas_release . . . 283
8.280.2.4 TlvPresent . . . 283
8.281 NASModePreferenceTlv Struct Reference . . . 283
8.281.1 Detailed Description . . . 283
8.281.2 Field Documentation . . . 283
8.281.2.1 ModePref . . . 283
8.281.2.2 TlvPresent . . . 283
8.282 NASNetSelPreferenceTlv Struct Reference . . . 283
8.282.1 Detailed Description . . . 284
8.282.2 Field Documentation . . . 284
8.282.2.1 NetSelPref . . . 284
8.282.2.2 TlvPresent . . . 284
8.283 NASOTAMessageTlv Struct Reference . . . 284
8.283.1 Detailed Description . . . 284
8.283.2 Field Documentation . . . 284
8.283.2.1 data_buf . . . 284
8.283.2.2 data_len . . . 285
8.283.2.3 message_type . . . 285
8.283.2.4 TlvPresent . . . 285
8.284 NASPhyCaAggPcellInfo Struct Reference . . . 285
8.284.1 Detailed Description . . . 285
8.284.2 Field Documentation . . . 285
8.284.2.1 dl_bw_value . . . 285
8.284.2.2 freq . . . 285
8.284.2.3 iLTEbandValue . . . 285

8.284.2.4 pci . . . 285
8.284.2.5 TlvPresent . . . 285
8.285 NASPhyCaAggScellDlBw Struct Reference . . . 286
8.285.1 Detailed Description . . . 286
8.285.2 Field Documentation . . . 286
8.285.2.1 dl_bw_value . . . 286
8.285.2.2 TlvPresent . . . 286
8.286 NASPhyCaAggScellIndex Struct Reference . . . 286
8.286.1 Detailed Description . . . 286
8.286.2 Field Documentation . . . 286
8.286.2.1 scell_idx . . . 286
8.286.2.2 TlvPresent . . . 286
8.287 NASPhyCaAggScellIndType Struct Reference . . . 287
8.287.1 Detailed Description . . . 287
8.287.2 Field Documentation . . . 287
8.287.2.1 freq . . . 287
8.287.2.2 pci . . . 287
8.287.2.3 scell_state . . . 287
8.287.2.4 TlvPresent . . . 287
8.288 NASPhyCaAggScellInfo Struct Reference . . . 287
8.288.1 Detailed Description . . . 288
8.288.2 Field Documentation . . . 288
8.288.2.1 dl_bw_value . . . 288
8.288.2.2 freq . . . 288
8.288.2.3 iLTEbandValue . . . 288
8.288.2.4 pci . . . 288
8.288.2.5 scell_state . . . 288
8.288.2.6 TlvPresent . . . 288
8.289 NASPRLPreferenceTlv Struct Reference . . . 288
8.289.1 Detailed Description . . . 288
8.289.2 Field Documentation . . . 289
8.289.2.1 PRLPref . . . 289
8.289.2.2 TlvPresent . . . 289
8.290 NASQmiCbkNasSwiOTAMessageInd Struct Reference . . . 289
8.290.1 Detailed Description . . . 289
8.290.2 Field Documentation . . . 289
8.290.2.1 nasRelInfoTlv . . . 289
8.290.2.2 otaMsgTlv . . . 289
8.290.2.3 timeTlv . . . 289
8.291 NASQmiCbkNasSystemSelPrefInd Struct Reference . . . 289

8.291.1 Detailed Description . . . 290
8.291.2 Field Documentation . . . 290
8.291.2.1 BPTlv . . . 291
8.291.2.2 EMTlv . . . 291
8.291.2.3 GWAOPTlv . . . 291
8.291.2.4 LBPTlv . . . 291
8.291.2.5 MPTlv . . . 291
8.291.2.6 NSPTlv . . . 291
8.291.2.7 PRLPTlv . . . 291
8.291.2.8 RPTlv . . . 291
8.291.2.9 SDPTlv . . . 291
8.292 NASRoamPreferenceTlv Struct Reference . . . 291
8.292.1 Detailed Description . . . 291
8.292.2 Field Documentation . . . 291
8.292.2.1 RoamPref . . . 291
8.292.2.2 TlvPresent . . . 291
8.293 NASServDomainPrefTlv Struct Reference . . . 291
8.293.1 Detailed Description . . . 291
8.293.2 Field Documentation . . . 292
8.293.2.1 SrvDomainPref . . . 292
8.293.2.2 TlvPresent . . . 292
8.294 NASServingSystemInfo Struct Reference . . . 292
8.294.1 Detailed Description . . . 292
8.294.2 Field Documentation . . . 293
8.294.2.1 csAttachState . . . 293
8.294.2.2 hdrPersonality . . . 293
8.294.2.3 psAttachState . . . 293
8.294.2.4 radioInterfaceList . . . 293
8.294.2.5 radioInterfaceNo . . . 293
8.294.2.6 registrationState . . . 293
8.294.2.7 selectedNetwork . . . 293
8.295 NASTimeInfoTlv Struct Reference . . . 293
8.295.1 Detailed Description . . . 293
8.295.2 Field Documentation . . . 294
8.295.2.1 time . . . 294
8.295.2.2 TlvPresent . . . 294
8.296 newMTMessageTlv Struct Reference . . . 294
8.296.1 Detailed Description . . . 294
8.296.2 Field Documentation . . . 294
8.296.2.1 MTMessageInfo . . . 294

8.296.2.2 TlvPresent . . . 294
8.297 pack_audio_SLQSGetAudioPathConfig_t Struct Reference . . . 294
8.297.1 Detailed Description . . . 294
8.297.2 Field Documentation . . . 295
8.297.2.1 Item . . . 295
8.297.2.2 Profile . . . 295
8.298 pack_audio_SLQSGetAudioProfile_t Struct Reference . . . 295
8.298.1 Detailed Description . . . 295
8.298.2 Field Documentation . . . 295
8.298.2.1 Generator . . . 295
8.299 pack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference . . . 296
8.299.1 Detailed Description . . . 296
8.299.2 Field Documentation . . . 296
8.299.2.1 Generator . . . 296
8.299.2.2 Item . . . 296
8.299.2.3 Profile . . . 296
8.299.2.4 Volume . . . 296
8.300 pack_audio_SLQSSetAudioPathConfig_t Struct Reference . . . 296
8.300.1 Detailed Description . . . 297
8.300.2 Field Documentation . . . 298
8.300.2.1 pCodecSTGain . . . 298
8.300.2.2 pDTMFTXGain . . . 298
8.300.2.3 pECMode . . . 298
8.300.2.4 pNSEnable . . . 298
8.300.2.5 Profile . . . 298
8.300.2.6 pRXAGCList . . . 298
8.300.2.7 pRXAVCAGCSwitch . . . 298
8.300.2.8 pRXAVCList . . . 298
8.300.2.9 pRXPCMIIRFltr . . . 298
8.300.2.10 pTXAGCList . . . 298
8.300.2.11 pTXAVCSwitch . . . 298
8.300.2.12 pTXGain . . . 298
8.300.2.13 pTXPCMIIRFltr . . . 298
8.301 pack_audio_SLQSSetAudioProfile_t Struct Reference . . . 298
8.301.1 Detailed Description . . . 299
8.301.2 Field Documentation . . . 299
8.301.2.1 EarMute . . . 299
8.301.2.2 Generator . . . 299
8.301.2.3 MicMute . . . 299
8.301.2.4 Profile . . . 299

8.301.2.5 Volume . . . 299
8.302 pack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference . . . 300
8.302.1 Detailed Description . . . 300
8.302.2 Field Documentation . . . 300
8.302.2.1 Generator . . . 300
8.302.2.2 Item . . . 300
8.302.2.3 Profile . . . 300
8.302.2.4 Volume . . . 300
8.302.2.5 VolValue . . . 300
8.303 pack_cat_CATSendEnvelopeCommand_t Struct Reference . . . 300
8.303.1 Detailed Description . . . 301
8.303.2 Field Documentation . . . 301
8.303.2.1 cmdID . . . 301
8.303.2.2 dataLen . . . 301
8.303.2.3 pData . . . 301
8.304 pack_cat_CATSendTerminalResponse_t Struct Reference . . . 301
8.304.1 Detailed Description . . . 301
8.304.2 Field Documentation . . . 302
8.304.2.1 dataLen . . . 302
8.304.2.2 pData . . . 302
8.304.2.3 refID . . . 302
8.305 pack_cat_SetCATEventCallback_t Struct Reference . . . 302
8.305.1 Detailed Description . . . 302
8.305.2 Field Documentation . . . 302
8.305.2.1 eventMask . . . 302
8.306 pack_dms_ActivateAutomatic_t Struct Reference . . . 303
8.306.1 Detailed Description . . . 303
8.306.2 Field Documentation . . . 303
8.306.2.1 actCode . . . 303
8.307 pack_dms_GetCustFeaturesV2_t Struct Reference . . . 303
8.307.1 Detailed Description . . . 303
8.307.2 Field Documentation . . . 303
8.307.2.1 cust_id . . . 303
8.307.2.2 list_type . . . 303
8.307.2.3 Tlvresult . . . 304
8.308 pack_dms_ResetToFactoryDefaults_t Struct Reference . . . 304
8.308.1 Detailed Description . . . 304
8.308.2 Field Documentation . . . 304
8.308.2.1 spc . . . 304
8.309 pack_dms_SetActivationStatusCallback_t Struct Reference . . . 304

8.309.1 Detailed Description . . . 304
8.309.2 Field Documentation . . . 304
8.309.2.1 activationState . . . 304
8.310 pack_dms_SetCrashAction_t Struct Reference . . . 304
8.310.1 Detailed Description . . . 305
8.310.2 Field Documentation . . . 305
8.310.2.1 crashAction . . . 305
8.311 pack_dms_SetCustFeature_t Struct Reference . . . 305
8.311.1 Detailed Description . . . 305
8.311.2 Field Documentation . . . 307
8.311.2.1 DHCPRelayEnabled . . . 307
8.311.2.2 DisableIMSI . . . 307
8.311.2.3 GpsEnable . . . 307
8.311.2.4 GPSLPM . . . 307
8.311.2.5 GPSSel . . . 307
8.311.2.6 IPFamSupport . . . 307
8.311.2.7 IsVoiceEnabled . . . 307
8.311.2.8 RMAutoConnect . . . 307
8.311.2.9 SMSSupport . . . 307
8.312 pack_dms_SetCustFeaturesV2_t Struct Reference . . . 307
8.312.1 Detailed Description . . . 307
8.312.2 Field Documentation . . . 308
8.312.2.1 cust_id . . . 308
8.312.2.2 cust_value . . . 308
8.312.2.3 Tlvresult . . . 308
8.312.2.4 value_length . . . 308
8.313 pack_dms_SetEventReport_t Struct Reference . . . 308
8.313.1 Detailed Description . . . 308
8.313.2 Field Documentation . . . 308
8.313.2.1 mode . . . 308
8.314 pack_dms_SetIndicationRegister_t Struct Reference . . . 308
8.314.1 Detailed Description . . . 308
8.314.2 Field Documentation . . . 309
8.314.2.1 PSMCfgChangeInfo . . . 309
8.314.2.2 PSMStatus . . . 309
8.314.2.3 RptIMSCapability . . . 309
8.315 pack_dms_SetPower_t Struct Reference . . . 309
8.315.1 Detailed Description . . . 309
8.315.2 Field Documentation . . . 310
8.315.2.1 mode . . . 310

8.315.2.2 Tlvresult . . . 310
8.316 pack_dms_SetUSBComp_t Struct Reference . . . 310
8.316.1 Detailed Description . . . 310
8.316.2 Field Documentation . . . 310
8.316.2.1 Tlvresult . . . 310
8.316.2.2 USBComp . . . 311
8.317 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 311
8.317.1 Detailed Description . . . 311
8.317.2 Field Documentation . . . 311
8.317.2.1 resetInfoInd . . . 311
8.318 pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference . . . 311
8.318.1 Detailed Description . . . 311
8.318.2 Field Documentation . . . 312
8.318.2.1 pActiveTimer . . . 312
8.318.2.2 pPeriodicUpdateTimer . . . 312
8.318.2.3 pPsmEnableState . . . 312
8.319 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 312
8.319.1 Detailed Description . . . 312
8.319.2 Field Documentation . . . 312
8.319.2.1 clear . . . 312
8.320 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 312
8.320.1 Detailed Description . . . 312
8.320.2 Field Documentation . . . 313
8.320.2.1 pDestSMSContent . . . 313
8.320.2.2 pDestSMSNum . . . 313
8.321 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 313
8.321.1 Detailed Description . . . 313
8.321.2 Field Documentation . . . 313
8.321.2.1 manString . . . 313
8.321.2.2 modelString . . . 314
8.321.2.3 plasmaIDString . . . 314
8.321.2.4 swVerString . . . 314
8.322 pack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 314
8.322.1 Detailed Description . . . 314
8.322.2 Field Documentation . . . 314
8.322.2.1 nameString . . . 314
8.322.2.2 versionString . . . 314
8.323 pack_dms_SwiSetEventReport_t Struct Reference . . . 314
8.323.1 Detailed Description . . . 314
8.323.2 Field Documentation . . . 315

8.323.2.1 pTempReport . . . 315
8.323.2.2 pVoltReport . . . 315
8.324 pack_dms_UIMChangePIN_t Struct Reference . . . 315
8.324.1 Detailed Description . . . 315
8.324.2 Field Documentation . . . 315
8.324.2.1 id . . . 315
8.324.2.2 newValue . . . 315
8.324.2.3 oldValue . . . 315
8.325 pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 315
8.325.1 Detailed Description . . . 316
8.325.2 Field Documentation . . . 316
8.325.2.1 facility . . . 316
8.326 pack_dms_UIMGetICCID_t Struct Reference . . . 316
8.326.1 Detailed Description . . . 316
8.326.2 Field Documentation . . . 316
8.326.2.1 ParamPresenceMask . . . 316
8.326.2.2 Tlvresult . . . 316
8.327 pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 316
8.327.1 Detailed Description . . . 317
8.327.2 Field Documentation . . . 317
8.327.2.1 facility . . . 317
8.327.2.2 facilityCk . . . 317
8.327.2.3 facilityState . . . 317
8.328 pack_dms_UIMSetPINProtection_t Struct Reference . . . 317
8.328.1 Detailed Description . . . 317
8.328.2 Field Documentation . . . 318
8.328.2.1 bEnable . . . 318
8.328.2.2 id . . . 318
8.328.2.3 value . . . 318
8.329 pack_dms_UIMUnblockControlKey_t Struct Reference . . . 318
8.329.1 Detailed Description . . . 318
8.329.2 Field Documentation . . . 318
8.329.2.1 facility . . . 318
8.329.2.2 facilityCk . . . 318
8.330 pack_dms_UIMUnblockPIN_t Struct Reference . . . 318
8.330.1 Detailed Description . . . 319
8.330.2 Field Documentation . . . 319
8.330.2.1 id . . . 319
8.330.2.2 newPin . . . 319
8.330.2.3 pukValue . . . 319

8.331 pack_dms_UIMVerifyPIN_t Struct Reference . . . 319
8.331.1 Detailed Description . . . 319
8.331.2 Field Documentation . . . 319
8.331.2.1 id . . . 319
8.331.2.2 value . . . 319
8.332 pack_fms_GetImagesPreference_t Struct Reference . . . 320
8.332.1 Detailed Description . . . 320
8.332.2 Field Documentation . . . 320
8.332.2.1 Tlvresult . . . 320
8.333 pack_fms_GetStoredImages_t Struct Reference . . . 320
8.333.1 Detailed Description . . . 320
8.333.2 Field Documentation . . . 320
8.333.2.1 Tlvresult . . . 320
8.334 pack_fms_SetImagesPreference_t Struct Reference . . . 320
8.334.1 Detailed Description . . . 321
8.334.2 Field Documentation . . . 321
8.334.2.1 bForceDownload . . . 321
8.334.2.2 imageListSize . . . 321
8.334.2.3 modemindex . . . 321
8.334.2.4 pImageList . . . 321
8.334.2.5 Tlvresult . . . 321
8.335 pack_ims_SLQSImsConfigIndicationRegister_t Struct Reference . . . 321
8.335.1 Detailed Description . . . 321
8.335.2 Field Documentation . . . 322
8.335.2.1 pRegMgrConfigEvents . . . 322
8.335.2.2 pSIPConfigEvents . . . 322
8.335.2.3 pSMSConfigEvents . . . 322
8.335.2.4 pUserConfigEvents . . . 322
8.335.2.5 pVoIPConfigEvents . . . 322
8.336 pack_ims_SLQSSetIMSSMSConfig_t Struct Reference . . . 323
8.336.1 Detailed Description . . . 323
8.336.2 Field Documentation . . . 323
8.336.2.1 pPhoneCtxtURI . . . 323
8.336.2.2 pPhoneCtxtURILen . . . 323
8.336.2.3 pSMSFormat . . . 323
8.336.2.4 pSMSOverIPNwInd . . . 323
8.337 pack_ims_SLQSSetIMSUserConfig_t Struct Reference . . . 323
8.337.1 Detailed Description . . . 324
8.337.2 Field Documentation . . . 324
8.337.2.1 pIMSDomain . . . 324

8.337.2.2 pIMSDomainLen . . . 324
8.338 pack_ims_SLQSSetIMSVoIPConfig_t Struct Reference . . . 324
8.338.1 Detailed Description . . . 324
8.338.2 Field Documentation . . . 326
8.338.2.1 pAmrMode . . . 326
8.338.2.2 pAmrOctetAligned . . . 326
8.338.2.3 pAmrWbEnable . . . 326
8.338.2.4 pAmrWBMode . . . 326
8.338.2.5 pAmrWBOctetAligned . . . 326
8.338.2.6 pMinSessionExpiryTimer . . . 326
8.338.2.7 pRingBackTimer . . . 326
8.338.2.8 pRingingTimer . . . 326
8.338.2.9 pRTPRTCPInactTimer . . . 326
8.338.2.10 pScrAmrEnable . . . 326
8.338.2.11 pScrAmrWbEnable . . . 326
8.338.2.12 pSessionExpiryTimer . . . 326
8.339 pack_ims_SLQSSetRegMgrConfig_t Struct Reference . . . 326
8.339.1 Detailed Description . . . 326
8.339.2 Field Documentation . . . 327
8.339.2.1 pCSCFPortName . . . 327
8.339.2.2 pCSCFPortNameLen . . . 327
8.339.2.3 pIMSTestMode . . . 327
8.339.2.4 pPriCSCFPort . . . 327
8.340 pack_ims_SLQSSetSIPConfig_t Struct Reference . . . 327
8.340.1 Detailed Description . . . 327
8.340.2 Field Documentation . . . 328
8.340.2.1 pSigCompEnabled . . . 328
8.340.2.2 pSIPLocalPort . . . 328
8.340.2.3 pSubscribeTimer . . . 328
8.340.2.4 pTimerSIPReg . . . 328
8.340.2.5 pTimerT1 . . . 328
8.340.2.6 pTimerT2 . . . 328
8.340.2.7 pTimerTf . . . 328
8.341 pack_imsa_SLQSRegisterIMSAIndication_t Struct Reference . . . 328
8.341.1 Detailed Description . . . 328
8.341.2 Field Documentation . . . 329
8.341.2.1 has_PdpStatusConfig . . . 329
8.341.2.2 has_RatHandoverStatusConfig . . . 329
8.341.2.3 has_RegStatusConfig . . . 329
8.341.2.4 has_ServiceStatusConfig . . . 329

8.341.2.5 PdpStatusConfig . . . 329
8.341.2.6 RatHandoverStatusConfig . . . 329
8.341.2.7 RegStatusConfig . . . 329
8.341.2.8 ServiceStatusConfig . . . 329
8.342 pack_loc_Delete_Assist_Data_t Struct Reference . . . 329
8.342.1 Detailed Description . . . 330
8.342.2 Field Documentation . . . 330
8.342.2.1 pBdsSVInfo . . . 330
8.342.2.2 pCellDb . . . 330
8.342.2.3 pClkInfo . . . 330
8.342.2.4 pGnssData . . . 330
8.342.2.5 pSVInfo . . . 330
8.342.2.6 Tlvresult . . . 330
8.343 pack_loc_EventRegister_t Struct Reference . . . 330
8.343.1 Detailed Description . . . 331
8.343.2 Field Documentation . . . 333
8.343.2.1 eventRegister . . . 333
8.343.2.2 Tlvresult . . . 333
8.344 pack_loc_SetExtPowerState_t Struct Reference . . . 333
8.344.1 Detailed Description . . . 333
8.344.2 Field Documentation . . . 333
8.344.2.1 extPowerState . . . 333
8.344.2.2 Tlvresult . . . 333
8.345 pack_loc_SetOperationMode_t Struct Reference . . . 333
8.345.1 Detailed Description . . . 334
8.345.2 Field Documentation . . . 334
8.345.2.1 mode . . . 334
8.345.2.2 Tlvresult . . . 334
8.346 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 334
8.346.1 Detailed Description . . . 334
8.346.2 Field Documentation . . . 334
8.346.2.1 Tlvresult . . . 334
8.346.2.2 xid . . . 335
8.347 pack_loc_SLQSLOCGetServer_t Struct Reference . . . 335
8.347.1 Detailed Description . . . 335
8.347.2 Field Documentation . . . 335
8.347.2.1 has_serverAddrTypeMask . . . 335
8.347.2.2 serverAddrTypeMask . . . 335
8.347.2.3 serverType . . . 335
8.348 pack_loc_SLQSLOCInjectPosition_t Struct Reference . . . 335

8.348.1 Detailed Description . . . 336
8.348.2 Field Documentation . . . 339
8.348.2.1 altitudeSrcInfo . . . 339
8.348.2.2 altitudeWrtEllipsoid . . . 339
8.348.2.3 altitudeWrtMeanSeaLevel . . . 339
8.348.2.4 has_altitudeSrcInfo . . . 339
8.348.2.5 has_altitudeWrtEllipsoid . . . 339
8.348.2.6 has_altitudeWrtMeanSeaLevel . . . 339
8.348.2.7 has_horConfidence . . . 339
8.348.2.8 has_horReliability . . . 339
8.348.2.9 has_horUncCircular . . . 339
8.348.2.10 has_latitude . . . 339
8.348.2.11 has_longitude . . . 339
8.348.2.12 has_positionSrc . . . 339
8.348.2.13 has_rawHorConfidence . . . 339
8.348.2.14 has_rawHorUncCircular . . . 339
8.348.2.15 has_timestampAge . . . 339
8.348.2.16 has_timestampUtc . . . 339
8.348.2.17 has_vertConfidence . . . 339
8.348.2.18 has_vertRelicability . . . 340
8.348.2.19 has_vertUnc . . . 340
8.348.2.20 horConfidence . . . 340
8.348.2.21 horReliability . . . 340
8.348.2.22 horUncCircular . . . 340
8.348.2.23 latitude . . . 340
8.348.2.24 longitude . . . 340
8.348.2.25 positionSrc . . . 340
8.348.2.26 rawHorConfidence . . . 340
8.348.2.27 rawHorUncCircular . . . 340
8.348.2.28 timestampAge . . . 340
8.348.2.29 timestampUtc . . . 340
8.348.2.30 vertConfidence . . . 340
8.348.2.31 vertReliability . . . 340
8.348.2.32 vertUnc . . . 340
8.349 pack_loc_SLQSLOCInjectSensorData_t Struct Reference . . . 340
8.349.1 Detailed Description . . . 340
8.349.2 Field Documentation . . . 341
8.349.2.1 acceleroData . . . 341
8.349.2.2 acceleroTimeSrc . . . 341
8.349.2.3 accelTemp . . . 341

8.349.2.4 gyroData . . . 341
8.349.2.5 gyroTemp . . . 342
8.349.2.6 gyroTimeSrc . . . 342
8.349.2.7 has_acceleroTimeSrc . . . 342
8.349.2.8 has_accelTemp . . . 342
8.349.2.9 has_accleroData . . . 342
8.349.2.10 has_gyroData . . . 342
8.349.2.11 has_gyroTemp . . . 342
8.349.2.12 has_gyroTimeSrc . . . 342
8.349.2.13 has_opaqueId . . . 342
8.349.2.14 opaqueId . . . 342
8.350 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . 342
8.350.1 Detailed Description . . . 342
8.350.2 Field Documentation . . . 342
8.350.2.1 timeMsec . . . 342
8.350.2.2 timeUncMsec . . . 342
8.351 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . 342
8.351.1 Detailed Description . . . 343
8.351.2 Field Documentation . . . 343
8.351.2.1 confidence . . . 343
8.351.2.2 has_confidence . . . 343
8.351.2.3 state . . . 343
8.352 pack_loc_SLQSLOCSetServer_t Struct Reference . . . 343
8.352.1 Detailed Description . . . 343
8.352.2 Field Documentation . . . 344
8.352.2.1 pIPv4Config . . . 344
8.352.2.2 pIPv6Config . . . 344
8.352.2.3 pURLAddr . . . 344
8.352.2.4 serverType . . . 344
8.353 pack_loc_Start_t Struct Reference . . . 344
8.353.1 Detailed Description . . . 344
8.353.2 Field Documentation . . . 345
8.353.2.1 pApplicationInfo . . . 345
8.353.2.2 pConfigAltitudeAssumed . . . 345
8.353.2.3 pHorizontalAccuracyLvl . . . 345
8.353.2.4 pIntermediateReportState . . . 345
8.353.2.5 pMinIntervalTime . . . 345
8.353.2.6 pRecurrenceType . . . 346
8.353.2.7 SessionId . . . 346
8.353.2.8 Tlvresult . . . 346

8.354 pack_loc_Stop_t Struct Reference . . . 346
8.354.1 Detailed Description . . . 346
8.354.2 Field Documentation . . . 346
8.354.2.1 SessionId . . . 346
8.354.2.2 Tlvresult . . . 346
8.355 pack_nas_InitiateDomainAttach_t Struct Reference . . . 346
8.355.1 Detailed Description . . . 346
8.355.2 Field Documentation . . . 346
8.355.2.1 action . . . 347
8.356 pack_nas_SetACCOLC_t Struct Reference . . . 347
8.356.1 Detailed Description . . . 347
8.356.2 Field Documentation . . . 347
8.356.2.1 accolc . . . 347
8.356.2.2 spc . . . 347
8.357 pack_nas_SetCDMANetworkParameters_t Struct Reference . . . 347
8.357.1 Detailed Description . . . 347
8.357.2 Field Documentation . . . 348
8.357.2.1 pApplication . . . 348
8.357.2.2 pBroadcast . . . 348
8.357.2.3 pCustomSCP . . . 348
8.357.2.4 pForceRev0 . . . 348
8.357.2.5 pProtocol . . . 348
8.357.2.6 pRoaming . . . 348
8.357.2.7 pSPC . . . 349
8.358 pack_nas_SetNetworkPreference_t Struct Reference . . . 349
8.358.1 Detailed Description . . . 349
8.358.2 Field Documentation . . . 349
8.358.2.1 Duration . . . 349
8.358.2.2 TechnologyPref . . . 349
8.358.2.3 Tlvresult . . . 349
8.359 pack_nas_SLQSConfigSigInfo_t Struct Reference . . . 349
8.359.1 Detailed Description . . . 350
8.359.2 Field Documentation . . . 351
8.359.2.1 pECIOThresh . . . 351
8.359.2.2 pHDRSINRThresh . . . 351
8.359.2.3 pIOThresh . . . 351
8.359.2.4 pLTESigRptCfg . . . 351
8.359.2.5 pLTESNRThresh . . . 351
8.359.2.6 pRSRPThresh . . . 351
8.359.2.7 pRSRQThresh . . . 351

8.359.2.8 pRSSIThresh . . . 351
8.359.2.9 pTDSCDMASINRCONFThresh . . . 351
8.360 pack_nas_SLQSGetPLMNName_t Struct Reference . . . 351
8.360.1 Detailed Description . . . 351
8.360.2 Field Documentation . . . 352
8.360.2.1 mcc . . . 352
8.360.2.2 mnc . . . 352
8.360.2.3 pMncPcsStatus . . . 352
8.361 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . 352
8.361.1 Detailed Description . . . 352
8.361.2 Field Documentation . . . 352
8.361.2.1 pChangeDuration . . . 352
8.361.2.2 pMncPcsDigitStatus . . . 352
8.361.2.3 pMNRInfo . . . 353
8.361.2.4 regAction . . . 353
8.362 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . 353
8.362.1 Detailed Description . . . 353
8.362.2 Field Documentation . . . 356
8.362.2.1 pCDMAECIODelta . . . 356
8.362.2.2 pCDMAECIOThresh . . . 356
8.362.2.3 pCDMARSSIDelta . . . 356
8.362.2.4 pCDMARSSIThresh . . . 356
8.362.2.5 pGSMRSSIDelta . . . 356
8.362.2.6 pGSMRSSIThresh . . . 356
8.362.2.7 pHDRECIODelta . . . 356
8.362.2.8 pHDRECIOThresh . . . 356
8.362.2.9 pHDRIODelta . . . 356
8.362.2.10 pHDRIOThresh . . . 356
8.362.2.11 pHDRRSSIDelta . . . 356
8.362.2.12 pHDRRSSIThresh . . . 356
8.362.2.13 pHDRSINRDelta . . . 356
8.362.2.14 pHDRSINRThresh . . . 356
8.362.2.15 pLTERSRPDelta . . . 357
8.362.2.16 pLTERSRPThresh . . . 357
8.362.2.17 pLTERSRQDelta . . . 357
8.362.2.18 pLTERSRQThresh . . . 357
8.362.2.19 pLTERSSIDelta . . . 357
8.362.2.20 pLTERSSIThresh . . . 357
8.362.2.21 pLTESigRptConfig . . . 357
8.362.2.22 pLTESNRDelta . . . 357

8.362.2.23 pLTESNRThresh . . . 357
8.362.2.24 pTDSCDMAECIODelta . . . 357
8.362.2.25 pTDSCDMAECIOThresh . . . 357
8.362.2.26 pTDSCDMARSCPDelta . . . 357
8.362.2.27 pTDSCDMARSCPThresh . . . 357
8.362.2.28 pTDSCDMARSSIDelta . . . 357
8.362.2.29 pTDSCDMARSSIThresh . . . 357
8.362.2.30 pTDSCDMASINRDelta . . . 357
8.362.2.31 pTDSCDMASINRThresh . . . 357
8.362.2.32 pWCDMAECIODelta . . . 357
8.362.2.33 pWCDMAECIOThresh . . . 357
8.362.2.34 pWCDMARSSIDelta . . . 357
8.362.2.35 pWCDMARSSIThresh . . . 357
8.363 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 357
8.363.1 Detailed Description . . . 357
8.363.2 Field Documentation . . . 358
8.363.2.1 namID . . . 358
8.364 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 358
8.364.1 Detailed Description . . . 358
8.364.2 Field Documentation . . . 358
8.364.2.1 radio_if . . . 358
8.365 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference . . . 358
8.365.1 Detailed Description . . . 359
8.365.2 Field Documentation . . . 360
8.365.2.1 pDDTMInd . . . 360
8.365.2.2 pDualStandByPrefInd . . . 361
8.365.2.3 pErrorRateInd . . . 361
8.365.2.4 pHDRNewUATIAssInd . . . 361
8.365.2.5 pHDRSessionCloseInd . . . 361
8.365.2.6 pLTECphyCa . . . 361
8.365.2.7 pManagedRoamingInd . . . 361
8.365.2.8 pNetworkRejectInd . . . 361
8.365.2.9 pNetworkTimeInd . . . 361
8.365.2.10 pServingSystemInd . . . 361
8.365.2.11 pSignalStrengthInd . . . 361
8.365.2.12 pSubscriptionInfoInd . . . 361
8.365.2.13 pSuppressSysInfoInd . . . 361
8.365.2.14 pSysInfoInd . . . 361
8.365.2.15 pSystemSelectionInd . . . 361
8.366 pack_nas_SLQSNASSeteDRXParams_t Struct Reference . . . 361

8.366.1 Detailed Description . . . 361
8.366.2 Field Documentation . . . 362
8.366.2.1 pCycleLen . . . 362
8.366.2.2 pEdrxEnable . . . 362
8.366.2.3 pPagingTimeWindow . . . 362
8.367 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference . . . 362
8.367.1 Detailed Description . . . 362
8.367.2 Field Documentation . . . 363
8.367.2.1 gsmUmtsDl . . . 363
8.367.2.2 gsmUmtsUl . . . 363
8.367.2.3 lteEmmDl . . . 363
8.367.2.4 lteEmmUl . . . 363
8.367.2.5 lteEsmDl . . . 363
8.367.2.6 lteEsmUl . . . 363
8.367.2.7 pRankIndicatorInd . . . 363
8.367.2.8 pTimer . . . 363
8.368 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference . . . 363
8.368.1 Detailed Description . . . 363
8.368.2 Field Documentation . . . 364
8.368.2.1 pLteEARFCN . . . 364
8.368.2.2 pLtePCI . . . 364
8.368.2.3 pWcdmaUARFCN . . . 364
8.369 pack_nas_SLQSSetBandPreference_t Struct Reference . . . 364
8.369.1 Detailed Description . . . 364
8.369.2 Field Documentation . . . 365
8.369.2.1 bandPref . . . 366
8.370 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . 366
8.370.1 Detailed Description . . . 366
8.370.2 Field Documentation . . . 366
8.370.2.1 bEnable . . . 366
8.370.2.2 pSigIndReq . . . 366
8.371 pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . 366
8.371.1 Detailed Description . . . 366
8.371.2 Field Documentation . . . 371
8.371.2.1 pAcqOrderPref . . . 371
8.371.2.2 pBandPref . . . 371
8.371.2.3 pChgDuration . . . 371
8.371.2.4 pCSGID . . . 371
8.371.2.5 pEmerMode . . . 371
8.371.2.6 pGWAcqOrderPref . . . 371

8.371.2.7 pLTEBandPref . . . 371
8.371.2.8 pMNCIncPCSDigStat . . . 371
8.371.2.9 pModePref . . . 371
8.371.2.10 pNetSelPref . . . 371
8.371.2.11 pPRLPref . . . 371
8.371.2.12 pRAT . . . 371
8.371.2.13 pRoamPref . . . 371
8.371.2.14 pSrvDomainPref . . . 371
8.371.2.15 pSrvRegRestriction . . . 372
8.371.2.16 pTdscdmaBandPref . . . 372
8.372 pack_nas_SLQSSetSysSelectionPrefExt_t Struct Reference . . . 372
8.372.1 Detailed Description . . . 372
8.372.2 Field Documentation . . . 381
8.372.2.1 pAcqOrderPref . . . 381
8.372.2.2 pBandPref . . . 381
8.372.2.3 pChgDuration . . . 381
8.372.2.4 pCiotAcqOrderPref . . . 381
8.372.2.5 pCiotLteOpMode . . . 381
8.372.2.6 pCiotLteOpModePref . . . 381
8.372.2.7 pCSGID . . . 381
8.372.2.8 pEmerMode . . . 381
8.372.2.9 pGWAcqOrderPref . . . 381
8.372.2.10 pLTEBandPref . . . 381
8.372.2.11 pLTEBandPrefExt . . . 381
8.372.2.12 pLteM1BandPref . . . 381
8.372.2.13 pLteNb1BandPref . . . 381
8.372.2.14 pMNCIncPCSDigStat . . . 381
8.372.2.15 pModePref . . . 381
8.372.2.16 pNetSelPref . . . 381
8.372.2.17 pNr5gBandPref . . . 381
8.372.2.18 pPRLPref . . . 382
8.372.2.19 pRAT . . . 382
8.372.2.20 pRoamPref . . . 382
8.372.2.21 pSrvDomainPref . . . 382
8.372.2.22 pSrvRegRestriction . . . 382
8.372.2.23 pTdscdmaBandPref . . . 382
8.373 pack_nas_SLQSSwiPSDetach_t Struct Reference . . . 382
8.373.1 Detailed Description . . . 382
8.373.2 Field Documentation . . . 382
8.373.2.1 pDetachAction . . . 382

8.374 pack_pds_PDSInjectTimeReference_t Struct Reference . . . 382
8.374.1 Detailed Description . . . 382
8.374.2 Field Documentation . . . 383
8.374.2.1 systemDiscontinuities . . . 383
8.374.2.2 systemTime . . . 383
8.375 pack_pds_ResetPDSData_t Struct Reference . . . 383
8.375.1 Detailed Description . . . 383
8.375.2 Field Documentation . . . 383
8.375.2.1 pCellDataMask . . . 384
8.375.2.2 pGPSDataMask . . . 384
8.376 pack_pds_SetEventReportCallback_t Struct Reference . . . 384
8.376.1 Detailed Description . . . 384
8.376.2 Field Documentation . . . 384
8.376.2.1 posDataNmea . . . 384
8.376.2.2 rptPosData . . . 384
8.377 pack_pds_SetPDSDefaults_t Struct Reference . . . 384
8.377.1 Detailed Description . . . 384
8.377.2 Field Documentation . . . 385
8.377.2.1 accuracy . . . 385
8.377.2.2 interval . . . 385
8.377.2.3 operation . . . 385
8.377.2.4 timeout . . . 385
8.378 pack_pds_SetPDSState_t Struct Reference . . . 385
8.378.1 Detailed Description . . . 385
8.378.2 Field Documentation . . . 385
8.378.2.1 enable . . . 385
8.379 pack_pds_SetPortAutomaticTracking_t Struct Reference . . . 385
8.379.1 Detailed Description . . . 386
8.379.2 Field Documentation . . . 386
8.379.2.1 bAuto . . . 386
8.380 pack_pds_SetServiceAutomaticTracking_t Struct Reference . . . 386
8.380.1 Detailed Description . . . 386
8.380.2 Field Documentation . . . 386
8.380.2.1 bAuto . . . 386
8.381 pack_pds_SetXTRAAutomaticDownload_t Struct Reference . . . 386
8.381.1 Detailed Description . . . 387
8.381.2 Field Documentation . . . 387
8.381.2.1 bEnabled . . . 387
8.381.2.2 interval . . . 387
8.382 pack_pds_SetXTRANetwork_t Struct Reference . . . 387

8.382.1 Detailed Description . . . 387
8.382.2 Field Documentation . . . 387
8.382.2.1 preference . . . 387
8.383 pack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 387
8.383.1 Detailed Description . . . 388
8.383.2 Field Documentation . . . 388
8.383.2.1 pNetworkMode . . . 388
8.384 pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference . . . 388
8.384.1 Detailed Description . . . 388
8.384.2 Field Documentation . . . 388
8.384.2.1 forceFlag . . . 388
8.384.2.2 timeBase . . . 388
8.384.2.3 timeMsec . . . 389
8.384.2.4 timeUncMsec . . . 389
8.385 pack_pds_SLQSPDSInjectPositionData_t Struct Reference . . . 389
8.385.1 Detailed Description . . . 389
8.385.2 Field Documentation . . . 390
8.385.2.1 pAltitudeWrtEllipsoid . . . 390
8.385.2.2 pAltitudeWrtSealevel . . . 390
8.385.2.3 pHorizontalConfidence . . . 390
8.385.2.4 pHorizontalUncCircular . . . 390
8.385.2.5 pLatitude . . . 390
8.385.2.6 pLongitude . . . 390
8.385.2.7 pPositionSource . . . 390
8.385.2.8 pTimeStamp . . . 390
8.385.2.9 pTimeType . . . 390
8.385.2.10 pVerticalConfidence . . . 390
8.385.2.11 pVerticalUnc . . . 391
8.386 pack_pds_SLQSSetAGPSConfig_t Struct Reference . . . 391
8.386.1 Detailed Description . . . 391
8.386.2 Field Documentation . . . 391
8.386.2.1 pNetworkMode . . . 391
8.386.2.2 pServerAddress . . . 391
8.386.2.3 pServerPort . . . 391
8.386.2.4 pServerURL . . . 391
8.386.2.5 pServerURLLength . . . 391
8.387 pack_pds_SLQSSetPositionMethodState_t Struct Reference . . . 391
8.387.1 Detailed Description . . . 392
8.387.2 Field Documentation . . . 392
8.387.2.1 pWifiState . . . 392

8.387.2.2 pXtraDataState . . . 392
8.387.2.3 pXtraTimeState . . . 392
8.388 pack_pds_StartPDSTrackingSessionExt_t Struct Reference . . . 392
8.388.1 Detailed Description . . . 392
8.388.2 Field Documentation . . . 393
8.388.2.1 fixAccuracy . . . 393
8.388.2.2 fixCount . . . 393
8.388.2.3 fixInterval . . . 393
8.388.2.4 fixTimeout . . . 393
8.388.2.5 sessionControl . . . 393
8.388.2.6 sessionOperation . . . 393
8.388.2.7 sessionServerOption . . . 393
8.388.2.8 sessionType . . . 393
8.389 pack_qmi_t Struct Reference . . . 393
8.389.1 Detailed Description . . . 394
8.389.2 Field Documentation . . . 394
8.389.2.1 msgid . . . 394
8.389.2.2 svc . . . 394
8.389.2.3 timeout . . . 394
8.389.2.4 xid . . . 394
8.390 pack_qos_BindDataPort_t Struct Reference . . . 394
8.390.1 Detailed Description . . . 394
8.390.2 Field Documentation . . . 395
8.390.2.1 pMuxID . . . 395
8.390.2.2 pPeripheralEndPointID . . . 395
8.390.2.3 pSIODataPort . . . 395
8.391 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 395
8.391.1 Detailed Description . . . 395
8.391.2 Field Documentation . . . 395
8.391.2.1 apnId . . . 395
8.392 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 395
8.392.1 Detailed Description . . . 395
8.392.2 Field Documentation . . . 395
8.392.2.1 apnId . . . 395
8.393 pack_qos_SLQSSetQosEventCallback_t Struct Reference . . . 395
8.393.1 Detailed Description . . . 396
8.393.2 Field Documentation . . . 396
8.393.2.1 enable . . . 396
8.394 pack_rms_SetSMSWake_t Struct Reference . . . 396
8.394.1 Detailed Description . . . 396

8.394.2 Field Documentation . . . 396
8.394.2.1 enabled . . . 396
8.394.2.2 wake_mask . . . 396
8.395 pack_sar_SLQSSetRfSarState_t Struct Reference . . . 396
8.395.1 Detailed Description . . . 397
8.395.2 Field Documentation . . . 397
8.395.2.1 RfSarState . . . 397
8.396 pack_sms_SaveSMS_t Struct Reference . . . 397
8.396.1 Detailed Description . . . 397
8.396.2 Field Documentation . . . 398
8.396.2.1 messageFormat . . . 398
8.396.2.2 messageSize . . . 398
8.396.2.3 pMessage . . . 398
8.396.2.4 storageType . . . 398
8.397 pack_sms_SendSMS_t Struct Reference . . . 398
8.397.1 Detailed Description . . . 398
8.397.2 Field Documentation . . . 398
8.397.2.1 messageFormat . . . 398
8.397.2.2 messageSize . . . 398
8.397.2.3 pLinktimer . . . 399
8.397.2.4 pMessage . . . 399
8.398 pack_sms_SetNewSMSCallback_t Struct Reference . . . 399
8.398.1 Detailed Description . . . 399
8.398.2 Field Documentation . . . 399
8.398.2.1 status . . . 399
8.399 pack_sms_SetSMSCAddress_t Struct Reference . . . 399
8.399.1 Detailed Description . . . 399
8.399.2 Field Documentation . . . 399
8.399.2.1 pSMSCAddress . . . 399
8.399.2.2 pSMSCType . . . 399
8.400 pack_sms_SLQSDeleteSMS_t Struct Reference . . . 399
8.400.1 Detailed Description . . . 400
8.400.2 Field Documentation . . . 400
8.400.2.1 pMessageIndex . . . 400
8.400.2.2 pMessageMode . . . 400
8.400.2.3 pMessageTag . . . 400
8.400.2.4 storageType . . . 400
8.401 pack_sms_SLQSGetSMS_t Struct Reference . . . 400
8.401.1 Detailed Description . . . 401
8.401.2 Field Documentation . . . 401

8.401.2.1 messageIndex . . . 401
8.401.2.2 pMessageMode . . . 401
8.401.2.3 storageType . . . 401
8.402 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 401
8.402.1 Detailed Description . . . 401
8.402.2 Field Documentation . . . 401
8.402.2.1 mode . . . 401
8.403 pack_sms_SLQSGetSMSList_t Struct Reference . . . 401
8.403.1 Detailed Description . . . 402
8.403.2 Field Documentation . . . 402
8.403.2.1 pMessageMode . . . 402
8.403.2.2 pRequestedTag . . . 402
8.403.2.3 storageType . . . 402
8.404 pack_sms_SLQSModifySMSStatus_t Struct Reference . . . 402
8.404.1 Detailed Description . . . 402
8.404.2 Field Documentation . . . 403
8.404.2.1 messageIndex . . . 403
8.404.2.2 messageTag . . . 403
8.404.2.3 pMessageMode . . . 403
8.404.2.4 storageType . . . 403
8.405 pack_sms_SLQSSendAsyncSMS_t Struct Reference . . . 403
8.405.1 Detailed Description . . . 403
8.405.2 Field Documentation . . . 403
8.405.2.1 pSendSmsParams . . . 403
8.406 pack_sms_SLQSSetIndicationRegister_t Struct Reference . . . 403
8.406.1 Detailed Description . . . 404
8.406.2 Field Documentation . . . 404
8.406.2.1 pSetIndicationRegReq . . . 404
8.407 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . 404
8.407.1 Detailed Description . . . 404
8.407.2 Field Documentation . . . 404
8.407.2.1 broadcastActivate . . . 404
8.407.2.2 mode . . . 404
8.408 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . 404
8.408.1 Detailed Description . . . 405
8.408.2 Field Documentation . . . 405
8.408.2.1 mode . . . 405
8.408.2.2 pBroadcastConfig . . . 405
8.408.2.3 pCDMABroadcastConfig . . . 405
8.409 pack_sms_SLQSSetSmsStorage_t Struct Reference . . . 405

8.409.1 Detailed Description . . . 405
8.409.2 Field Documentation . . . 405
8.409.2.1 smsStorage . . . 406
8.410 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 406
8.410.1 Detailed Description . . . 406
8.410.2 Field Documentation . . . 406
8.410.2.1 pMaxStorageSizeReq . . . 406
8.411 pack_sms_SLQSSmsSetRoutes_t Struct Reference . . . 406
8.411.1 Detailed Description . . . 406
8.411.2 Field Documentation . . . 406
8.411.2.1 pSetRoutesReq . . . 406
8.412 pack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference . . . 406
8.412.1 Detailed Description . . . 407
8.412.2 Field Documentation . . . 407
8.412.2.1 pGenerator . . . 407
8.413 pack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference . . . 407
8.413.1 Detailed Description . . . 407
8.413.2 Field Documentation . . . 407
8.413.2.1 Generator . . . 407
8.413.2.2 Profile . . . 407
8.414 pack_swiaudio_SLQSGetM2MAVMute_t Struct Reference . . . 407
8.414.1 Detailed Description . . . 407
8.414.2 Field Documentation . . . 408
8.414.2.1 Profile . . . 408
8.415 pack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference . . . 408
8.415.1 Detailed Description . . . 408
8.415.2 Field Documentation . . . 408
8.415.2.1 Profile . . . 408
8.416 pack_swiaudio_SLQSSetM2MAudioAVCFG_t Struct Reference . . . 408
8.416.1 Detailed Description . . . 408
8.416.2 Field Documentation . . . 409
8.416.2.1 Device . . . 409
8.416.2.2 PIFACEId . . . 409
8.416.2.3 pPCMParams . . . 409
8.416.2.4 Profile . . . 409
8.417 pack_swiaudio_SLQSSetM2MAudioLPBK_t Struct Reference . . . 409
8.417.1 Detailed Description . . . 409
8.417.2 Field Documentation . . . 409
8.417.2.1 Enable . . . 409
8.418 pack_swiaudio_SLQSSetM2MAudioProfile_t Struct Reference . . . 409

8.418.1 Detailed Description . . . 410
8.418.2 Field Documentation . . . 410
8.418.2.1 pCwtMute . . . 410
8.418.2.2 pEarMute . . . 410
8.418.2.3 pGenerator . . . 410
8.418.2.4 pMicMute . . . 410
8.418.2.5 Profile . . . 410
8.418.2.6 pVolume . . . 410
8.419 pack_swiaudio_SLQSSetM2MAudioVolume_t Struct Reference . . . 411
8.419.1 Detailed Description . . . 411
8.419.2 Field Documentation . . . 411
8.419.2.1 Generator . . . 411
8.419.2.2 Level . . . 411
8.419.2.3 Profile . . . 411
8.420 pack_swiaudio_SLQSSetM2MAVMute_t Struct Reference . . . 411
8.420.1 Detailed Description . . . 411
8.420.2 Field Documentation . . . 412
8.420.2.1 EarMute . . . 412
8.420.2.2 MicMute . . . 412
8.420.2.3 pCwtMute . . . 412
8.420.2.4 Profile . . . 412
8.421 pack_swiaudio_SLQSSetM2MSpkrGain_t Struct Reference . . . 412
8.421.1 Detailed Description . . . 412
8.421.2 Field Documentation . . . 412
8.421.2.1 Profile . . . 412
8.421.2.2 Value . . . 412
8.422 pack_swiavms_SLQSAVMSSendSelection_t Struct Reference . . . 413
8.422.1 Detailed Description . . . 413
8.422.2 Field Documentation . . . 413
8.422.2.1 pClientPerformOperationFlag . . . 413
8.422.2.2 pDeferTime . . . 413
8.422.2.3 pPackageID . . . 413
8.422.2.4 pRejectReason . . . 413
8.422.2.5 selection . . . 413
8.423 pack_swiavms_SLQSAVMSSetSettings_t Struct Reference . . . 413
8.423.1 Detailed Description . . . 414
8.423.2 Field Documentation . . . 415
8.423.2.1 AutoConnect . . . 415
8.423.2.2 AutoReboot . . . 415
8.423.2.3 pAPNInfo . . . 415

8.423.2.4 pConnectionRetryTimers . . . 415
8.423.2.5 pFwAutoSDM . . . 415
8.423.2.6 pNotifStore . . . 415
8.423.2.7 pPeriodInfo . . . 415
8.423.2.8 pPollingTimer . . . 415
8.423.2.9 PromptFwDownload . . . 415
8.423.2.10 PromptFwUpdate . . . 415
8.424 pack_swiavms_SLQSAVMSSetSettings_v2_t Struct Reference . . . 415
8.424.1 Detailed Description . . . 415
8.424.2 Field Documentation . . . 416
8.424.2.1 AutoConnect . . . 417
8.424.2.2 pAPNInfo . . . 417
8.424.2.3 pAutoReboot . . . 417
8.424.2.4 pConnectionRetryTimers . . . 417
8.424.2.5 pFwAutoSDM . . . 417
8.424.2.6 pNotifStore . . . 417
8.424.2.7 pPeriodInfo . . . 417
8.424.2.8 pPollingTimer . . . 417
8.424.2.9 PromptFwDownload . . . 417
8.424.2.10 PromptFwUpdate . . . 417
8.425 pack_swiavms_SLQSAVMSStartSession_t Struct Reference . . . 417
8.425.1 Detailed Description . . . 417
8.425.2 Field Documentation . . . 417
8.425.2.1 sessionType . . . 417
8.426 pack_swiavms_SLQSAVMSStopSession_t Struct Reference . . . 417
8.426.1 Detailed Description . . . 417
8.426.2 Field Documentation . . . 418
8.426.2.1 sessionType . . . 418
8.427 pack_swidms_SLQSSwiDmsSetMTU_t Struct Reference . . . 418
8.427.1 Detailed Description . . . 418
8.427.2 Field Documentation . . . 418
8.427.2.1 MTUSize . . . 418
8.428 pack_swidms_SLQSSwiDmsSetUsbNetNum_t Struct Reference . . . 418
8.428.1 Detailed Description . . . 418
8.428.2 Field Documentation . . . 419
8.428.2.1 nUsbNetNum . . . 419
8.429 pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . 419
8.429.1 Detailed Description . . . 419
8.429.2 Field Documentation . . . 420
8.429.2.1 fix_rate . . . 420

8.429.2.2 fix_type . . . 420
8.429.2.3 function . . . 420
8.429.2.4 max_dist . . . 420
8.429.2.5 max_time . . . 420
8.429.2.6 set_fix_rate . . . 420
8.429.2.7 set_fix_type . . . 420
8.429.2.8 set_function . . . 420
8.429.2.9 set_max_dist . . . 420
8.429.2.10 set_max_time . . . 420
8.430 pack_swioma_SLQSOMADMCancelSession_t Struct Reference . . . 420
8.430.1 Detailed Description . . . 421
8.430.2 Field Documentation . . . 421
8.430.2.1 sessionType . . . 421
8.431 pack_swioma_SLQSOMADMCancelSessionExt_t Struct Reference . . . 421
8.431.1 Detailed Description . . . 421
8.431.2 Field Documentation . . . 421
8.431.2.1 sessionType . . . 421
8.432 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 421
8.432.1 Detailed Description . . . 421
8.432.2 Field Documentation . . . 422
8.432.2.1 SessionType . . . 422
8.433 pack_swioma_SLQSOMADMSendSelection_t Struct Reference . . . 422
8.433.1 Detailed Description . . . 422
8.433.2 Field Documentation . . . 422
8.433.2.1 pDeferTime . . . 422
8.433.2.2 pRejectReason . . . 422
8.433.2.3 selection . . . 422
8.434 pack_swioma_SLQSOMADMSendSelectionExt_t Struct Reference . . . 422
8.434.1 Detailed Description . . . 423
8.434.2 Field Documentation . . . 423
8.434.2.1 selection . . . 423
8.435 pack_swioma_SLQSOMADMSetSettings_t Struct Reference . . . 423
8.435.1 Detailed Description . . . 423
8.435.2 Field Documentation . . . 424
8.435.2.1 FOTAdownload . . . 424
8.435.2.2 FOTAUpdate . . . 424
8.435.2.3 pAutosdm . . . 424
8.435.2.4 pFwAutoCheck . . . 424
8.436 pack_swioma_SLQSOMADMSetSettingsExt_t Struct Reference . . . 424
8.436.1 Detailed Description . . . 424

8.436.2 Field Documentation . . . 425
8.436.2.1 FOTAdownload . . . 425
8.436.2.2 FOTAUpdate . . . 425
8.436.2.3 FUMOEnable . . . 425
8.436.2.4 OMADMEnable . . . 425
8.436.2.5 OMADMLogEnable . . . 425
8.436.2.6 PRLEnable . . . 425
8.437 pack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 425
8.437.1 Detailed Description . . . 425
8.437.2 Field Documentation . . . 426
8.437.2.1 sessionType . . . 426
8.438 pack_swioma_SLQSOMADMStartSessionExt_t Struct Reference . . . 426
8.438.1 Detailed Description . . . 426
8.438.2 Field Documentation . . . 426
8.438.2.1 sessionType . . . 426
8.439 pack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference . . . 426
8.439.1 Detailed Description . . . 426
8.439.2 Field Documentation . . . 426
8.439.2.1 mitigationDevID . . . 426
8.439.2.2 mitigationDevIDLen . . . 427
8.440 pack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference . . . 427
8.440.1 Detailed Description . . . 427
8.440.2 Field Documentation . . . 427
8.440.2.1 mitigationDevID . . . 427
8.440.2.2 mitigationDevIDLen . . . 427
8.441 pack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference . . . 427
8.441.1 Detailed Description . . . 427
8.441.2 Field Documentation . . . 427
8.441.2.1 mitigationDevID . . . 428
8.441.2.2 mitigationDevIDLen . . . 428
8.442 pack_uim_ChangePin_t Struct Reference . . . 428
8.442.1 Detailed Description . . . 428
8.442.2 Field Documentation . . . 428
8.442.2.1 changePIN . . . 428
8.442.2.2 EncryptedPIN1 . . . 428
8.442.2.3 pIndicationToken . . . 428
8.442.2.4 pKeyReferenceID . . . 429
8.442.2.5 sessionInfo . . . 429
8.442.2.6 Tlvresult . . . 429
8.443 pack_uim_ReadTransparent_t Struct Reference . . . 429

8.443.1 Detailed Description . . . 429
8.443.2 Field Documentation . . . 429
8.443.2.1 fileIndex . . . 429
8.443.2.2 pEncryptData . . . 430
8.443.2.3 pIndicationToken . . . 430
8.443.2.4 readTransparent . . . 430
8.443.2.5 sessionInfo . . . 430
8.443.2.6 Tlvresult . . . 430
8.444 pack_uim_SetPinProtection_t Struct Reference . . . 430
8.444.1 Detailed Description . . . 430
8.444.2 Field Documentation . . . 430
8.444.2.1 EncryptedPIN1 . . . 431
8.444.2.2 pIndicationToken . . . 431
8.444.2.3 pinProtection . . . 431
8.444.2.4 pKeyReferenceID . . . 431
8.444.2.5 sessionInfo . . . 431
8.444.2.6 Tlvresult . . . 431
8.445 pack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 431
8.445.1 Detailed Description . . . 431
8.445.2 Field Documentation . . . 431
8.445.2.1 authData . . . 431
8.445.2.2 pIndicationToken . . . 431
8.445.2.3 sessionInfo . . . 431
8.446 pack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 431
8.446.1 Detailed Description . . . 432
8.446.2 Field Documentation . . . 432
8.446.2.1 depersonilisationInfo . . . 432
8.447 pack_uim_SLQSUIMEventRegister_t Struct Reference . . . 432
8.447.1 Detailed Description . . . 432
8.447.2 Field Documentation . . . 432
8.447.2.1 eventMask . . . 432
8.448 pack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 432
8.448.1 Detailed Description . . . 432
8.448.2 Field Documentation . . . 433
8.448.2.1 pConfigurationMask . . . 433
8.449 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 433
8.449.1 Detailed Description . . . 433
8.449.2 Field Documentation . . . 433
8.449.2.1 fileIndex . . . 433
8.449.2.2 pIndicationToken . . . 433

8.449.2.3 sessionInfo . . . 434
8.450 pack_uim_SLQSUIMPowerDown_t Struct Reference . . . 434
8.450.1 Detailed Description . . . 434
8.450.2 Field Documentation . . . 434
8.450.2.1 slot . . . 434
8.451 pack_uim_SLQSUIMPowerUp_t Struct Reference . . . 434
8.451.1 Detailed Description . . . 434
8.451.2 Field Documentation . . . 434
8.451.2.1 pIgnoreHotSwapSwitch . . . 434
8.451.2.2 slot . . . 435
8.452 pack_uim_SLQSUIMRefreshComplete_t Struct Reference . . . 435
8.452.1 Detailed Description . . . 435
8.452.2 Field Documentation . . . 435
8.452.2.1 refreshComplete . . . 435
8.452.2.2 sessionInfo . . . 435
8.453 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 435
8.453.1 Detailed Description . . . 435
8.453.2 Field Documentation . . . 435
8.453.2.1 sessionInfo . . . 435
8.454 pack_uim_SLQSUIMRefreshOK_t Struct Reference . . . 436
8.454.1 Detailed Description . . . 436
8.454.2 Field Documentation . . . 436
8.454.2.1 OKtoRefresh . . . 436
8.454.2.2 sessionInfo . . . 436
8.455 pack_uim_SLQSUIMRefreshRegister_t Struct Reference . . . 436
8.455.1 Detailed Description . . . 436
8.455.2 Field Documentation . . . 436
8.455.2.1 regRefresh . . . 437
8.455.2.2 sessionInfo . . . 437
8.456 pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . 437
8.456.1 Detailed Description . . . 437
8.456.2 Field Documentation . . . 437
8.456.2.1 bLogicalSlot . . . 437
8.456.2.2 ulPhysicalSlot . . . 437
8.457 pack_uim_UnblockPin_t Struct Reference . . . 437
8.457.1 Detailed Description . . . 438
8.457.2 Field Documentation . . . 438
8.457.2.1 EncryptedPIN1 . . . 438
8.457.2.2 pIndicationToken . . . 438
8.457.2.3 pinProtection . . . 438

8.457.2.4 pKeyReferenceID . . . 438
8.457.2.5 sessionInfo . . . 438
8.457.2.6 Tlvresult . . . 438
8.458 pack_uim_VerifyPin_t Struct Reference . . . 438
8.458.1 Detailed Description . . . 439
8.458.2 Field Documentation . . . 439
8.458.2.1 pEncryptedPIN1 . . . 439
8.458.2.2 pIndicationToken . . . 439
8.458.2.3 pKeyReferenceID . . . 439
8.458.2.4 sessionInfo . . . 439
8.458.2.5 Tlvresult . . . 439
8.458.2.6 verifyPIN . . . 439
8.459 pack_voice_AnswerUSSD_t Struct Reference . . . 439
8.459.1 Detailed Description . . . 440
8.459.2 Field Documentation . . . 440
8.459.2.1 pInfo . . . 440
8.460 pack_voice_OriginateUSSD_t Struct Reference . . . 440
8.460.1 Detailed Description . . . 440
8.460.2 Field Documentation . . . 440
8.460.2.1 pInfo . . . 440
8.461 pack_voice_SLQSOriginateUSSD_t Struct Reference . . . 440
8.461.1 Detailed Description . . . 440
8.461.2 Field Documentation . . . 441
8.461.2.1 ussData . . . 441
8.461.2.2 ussDCS . . . 441
8.461.2.3 ussLen . . . 441
8.462 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference . . . 441
8.462.1 Detailed Description . . . 441
8.462.2 Field Documentation . . . 441
8.462.2.1 lineValue . . . 441
8.463 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference . . . 441
8.463.1 Detailed Description . . . 442
8.463.2 Field Documentation . . . 442
8.463.2.1 switchOption . . . 442
8.464 pack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 442
8.464.1 Detailed Description . . . 442
8.464.2 Field Documentation . . . 442
8.464.2.1 pCallId . . . 442
8.465 pack_voice_SLQSVoiceBindSubscription_t Struct Reference . . . 442
8.465.1 Detailed Description . . . 442

8.465.2 Field Documentation . . . 443
8.465.2.1 subsType . . . 443
8.466 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 443
8.466.1 Detailed Description . . . 443
8.466.2 Field Documentation . . . 443
8.466.2.1 BurstDTMFInfo . . . 443
8.466.2.2 pBurstDTMFLengths . . . 443
8.467 pack_voice_SLQSVoiceDialCall_t Struct Reference . . . 443
8.467.1 Detailed Description . . . 443
8.467.2 Field Documentation . . . 445
8.467.2.1 callNumber . . . 445
8.467.2.2 pCallPartySubAdd . . . 445
8.467.2.3 pCallType . . . 445
8.467.2.4 pCLIRType . . . 445
8.467.2.5 pCUGInfo . . . 445
8.467.2.6 pEmergencyCategory . . . 445
8.467.2.7 pSvcType . . . 445
8.467.2.8 pUUSInFo . . . 445
8.468 pack_voice_SLQSVoiceEndCall_t Struct Reference . . . 445
8.468.1 Detailed Description . . . 445
8.468.2 Field Documentation . . . 445
8.468.2.1 pCallId . . . 445
8.469 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 445
8.469.1 Detailed Description . . . 445
8.469.2 Field Documentation . . . 446
8.469.2.1 pSvcClass . . . 446
8.469.2.2 reason . . . 446
8.470 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 446
8.470.1 Detailed Description . . . 446
8.470.2 Field Documentation . . . 447
8.470.2.1 pSvcClass . . . 447
8.470.2.2 Reason . . . 447
8.471 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 447
8.471.1 Detailed Description . . . 447
8.471.2 Field Documentation . . . 447
8.471.2.1 callID . . . 447
8.472 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 447
8.472.1 Detailed Description . . . 448
8.472.2 Field Documentation . . . 448
8.472.2.1 pSvcClass . . . 448

8.473 pack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 448
8.473.1 Detailed Description . . . 448
8.473.2 Field Documentation . . . 449
8.473.2.1 pAirTimer . . . 449
8.473.2.2 pAMRStatus . . . 449
8.473.2.3 pAutoAnswer . . . 449
8.473.2.4 pNamID . . . 449
8.473.2.5 pPrefVoicePrivacy . . . 449
8.473.2.6 pPrefVoiceSO . . . 449
8.473.2.7 pRoamTimer . . . 449
8.473.2.8 pTTYMode . . . 449
8.473.2.9 pVoiceDomainPref . . . 449
8.474 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference . . . 450
8.474.1 Detailed Description . . . 450
8.474.2 Field Documentation . . . 450
8.474.2.1 pRegDTMFEvents . . . 450
8.474.2.2 pRegVoicePrivacyEvents . . . 450
8.474.2.3 pSuppsNotifEvents . . . 450
8.475 pack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 450
8.475.1 Detailed Description . . . 451
8.475.2 Field Documentation . . . 451
8.475.2.1 pCallID . . . 451
8.475.2.2 SUPSType . . . 451
8.476 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference . . . 451
8.476.1 Detailed Description . . . 451
8.476.2 Field Documentation . . . 452
8.476.2.1 USSInformation . . . 452
8.477 pack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 452
8.477.1 Detailed Description . . . 452
8.477.2 Field Documentation . . . 452
8.477.2.1 pCallID . . . 452
8.477.2.2 pFlashPayLd . . . 452
8.477.2.3 pFlashType . . . 452
8.478 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 452
8.478.1 Detailed Description . . . 453
8.478.2 Field Documentation . . . 453
8.478.2.1 newPasswd . . . 453
8.478.2.2 newPasswdAgain . . . 453
8.478.2.3 oldPasswd . . . 453
8.478.2.4 Reason . . . 453

8.479 pack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 453
8.479.1 Detailed Description . . . 454
8.479.2 Field Documentation . . . 455
8.479.2.1 pAirTimerConfig . . . 455
8.479.2.2 pAutoAnswer . . . 455
8.479.2.3 pPrefVoiceDomain . . . 455
8.479.2.4 pPrefVoiceSO . . . 455
8.479.2.5 pRoamTimerConfig . . . 455
8.479.2.6 pTTYMode . . . 455
8.480 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference . . . 455
8.480.1 Detailed Description . . . 455
8.480.2 Field Documentation . . . 455
8.480.2.1 privacyPref . . . 455
8.481 pack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 455
8.481.1 Detailed Description . . . 455
8.481.2 Field Documentation . . . 457
8.481.2.1 pCallBarringPasswd . . . 457
8.481.2.2 pCallForwardingNumber . . . 457
8.481.2.3 pCallFwdTypeAndPlan . . . 457
8.481.2.4 pServiceClass . . . 457
8.481.2.5 pTimerVal . . . 457
8.481.2.6 reason . . . 457
8.481.2.7 voiceSvc . . . 457
8.482 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 457
8.482.1 Detailed Description . . . 457
8.482.2 Field Documentation . . . 458
8.482.2.1 DTMFdigit . . . 458
8.482.2.2 pCallID . . . 458
8.483 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 458
8.483.1 Detailed Description . . . 458
8.483.2 Field Documentation . . . 458
8.483.2.1 callID . . . 458
8.484 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . 458
8.484.1 Detailed Description . . . 458
8.484.2 Field Documentation . . . 459
8.484.2.1 pEnableNotification . . . 459
8.485 pack_wds_GetDefaultProfile_t Struct Reference . . . 459
8.485.1 Detailed Description . . . 459
8.485.2 Field Documentation . . . 459
8.485.2.1 profiletype . . . 459

8.486 pack_wds_GetDefaultProfileNum_t Struct Reference . . . 459
8.486.1 Detailed Description . . . 459
8.486.2 Field Documentation . . . 459
8.486.2.1 family . . . 459
8.486.2.2 type . . . 460
8.487 pack_wds_GetDormancyState_t Struct Reference . . . 460
8.487.1 Detailed Description . . . 460
8.488 pack_wds_GetLastMobileIPError_t Struct Reference . . . 460
8.488.1 Detailed Description . . . 460
8.489 pack_wds_GetMobileIP_t Struct Reference . . . 460
8.489.1 Detailed Description . . . 460
8.490 pack_wds_GetMobileIPProfile_t Struct Reference . . . 460
8.490.1 Detailed Description . . . 460
8.490.2 Field Documentation . . . 461
8.490.2.1 index . . . 461
8.491 pack_wds_GetPacketStatistics_t Struct Reference . . . 461
8.491.1 Detailed Description . . . 461
8.491.2 Field Documentation . . . 461
8.491.2.1 pStatMask . . . 461
8.492 pack_wds_GetPacketStatus_t Struct Reference . . . 461
8.492.1 Detailed Description . . . 461
8.492.2 Field Documentation . . . 462
8.492.2.1 statmask . . . 462
8.493 pack_wds_GetSessionDuration_t Struct Reference . . . 462
8.493.1 Detailed Description . . . 462
8.494 pack_wds_RMSetTransferStatistics_t Struct Reference . . . 462
8.494.1 Detailed Description . . . 462
8.494.2 Field Documentation . . . 462
8.494.2.1 RmTrasnferStaticsReq . . . 462
8.495 pack_wds_SetAutoconnect_t Struct Reference . . . 463
8.495.1 Detailed Description . . . 463
8.495.2 Field Documentation . . . 463
8.495.2.1 acroamsetting . . . 463
8.495.2.2 acsetting . . . 463
8.496 pack_wds_SetDefaultProfile_t Struct Reference . . . 463
8.496.1 Detailed Description . . . 463
8.496.2 Field Documentation . . . 464
8.496.2.1 authentication . . . 465
8.496.2.2 ipAddress . . . 465
8.496.2.3 pApnname . . . 465

8.496.2.4 pdpType . . . 465
8.496.2.5 pName . . . 465
8.496.2.6 pPassword . . . 465
8.496.2.7 primaryDNS . . . 465
8.496.2.8 profileType . . . 465
8.496.2.9 pUsername . . . 465
8.496.2.10 secondaryDNS . . . 465
8.497 pack_wds_SetDefaultProfileNum_t Struct Reference . . . 465
8.497.1 Detailed Description . . . 465
8.497.2 Field Documentation . . . 465
8.497.2.1 family . . . 465
8.497.2.2 index . . . 465
8.497.2.3 type . . . 465
8.498 pack_wds_SetMobileIP_t Struct Reference . . . 466
8.498.1 Detailed Description . . . 466
8.498.2 Field Documentation . . . 466
8.498.2.1 mode . . . 466
8.499 pack_wds_SetMobileIPParameters_t Struct Reference . . . 466
8.499.1 Detailed Description . . . 466
8.499.2 Field Documentation . . . 467
8.499.2.1 pHA2002bis . . . 467
8.499.2.2 pHAAuthenticator . . . 467
8.499.2.3 pMode . . . 467
8.499.2.4 pReRegPeriod . . . 467
8.499.2.5 pReRegTraffic . . . 467
8.499.2.6 pRetryInterval . . . 467
8.499.2.7 pRetryLimit . . . 467
8.499.2.8 pSPC . . . 467
8.500 pack_wds_SetMobileIPProfile_t Struct Reference . . . 467
8.500.1 Detailed Description . . . 468
8.500.2 Field Documentation . . . 468
8.500.2.1 index . . . 469
8.500.2.2 pAAASPI . . . 469
8.500.2.3 pAddress . . . 469
8.500.2.4 pEnabled . . . 469
8.500.2.5 pHASPI . . . 469
8.500.2.6 pMNAAA . . . 469
8.500.2.7 pMNHA . . . 469
8.500.2.8 pNAI . . . 469
8.500.2.9 pPrimaryHA . . . 469

8.500.2.10 pRevTunneling . . . 469
8.500.2.11 pSecondaryHA . . . 469
8.500.2.12 spc . . . 469
8.501 pack_wds_SLQSCreateProfile_t Struct Reference . . . 469
8.501.1 Detailed Description . . . 469
8.501.2 Field Documentation . . . 470
8.501.2.1 pCurProfile . . . 470
8.501.2.2 pProfileId . . . 470
8.501.2.3 pProfileType . . . 470
8.502 pack_wds_SLQSDeleteProfile_t Struct Reference . . . 470
8.502.1 Detailed Description . . . 470
8.502.2 Field Documentation . . . 470
8.502.2.1 profileIndex . . . 470
8.502.2.2 profileType . . . 470
8.503 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 470
8.503.1 Detailed Description . . . 470
8.504 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 471
8.504.1 Detailed Description . . . 471
8.505 pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 471
8.505.1 Detailed Description . . . 471
8.505.2 Field Documentation . . . 472
8.505.2.1 Mask . . . 472
8.505.2.2 pReportChannelRate . . . 472
8.505.2.3 pReportConnStatus . . . 472
8.505.2.4 pReportDataBearerTech . . . 472
8.505.2.5 pReportDormStatus . . . 472
8.505.2.6 pTransferStatInd . . . 472
8.506 pack_wds_SLQSGetProfileSettings_t Struct Reference . . . 472
8.506.1 Detailed Description . . . 472
8.506.2 Field Documentation . . . 473
8.506.2.1 ProfileId . . . 473
8.506.2.2 ProfileType . . . 473
8.507 pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 473
8.507.1 Detailed Description . . . 473
8.507.2 Field Documentation . . . 474
8.507.2.1 pReqSettings . . . 474
8.508 pack_wds_SLQSModifyProfile_t Struct Reference . . . 474
8.508.1 Detailed Description . . . 474
8.508.2 Field Documentation . . . 475
8.508.2.1 curProfile . . . 475

8.508.2.2 pProfileId . . . 475
8.508.2.3 pProfileType . . . 475
8.509 pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . 475
8.509.1 Detailed Description . . . 475
8.509.2 Field Documentation . . . 476
8.509.2.1 LTEAttachProfileListLen . . . 476
8.509.2.2 p3gppRelease . . . 476
8.509.2.3 pDefaultPDNEnabled . . . 476
8.509.2.4 pLTEAttachProfile . . . 476
8.509.2.5 pLTEAttachProfileList . . . 476
8.509.2.6 pProfileList . . . 477
8.510 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 477
8.510.1 Detailed Description . . . 477
8.510.2 Field Documentation . . . 477
8.510.2.1 IPFamilyPreference . . . 477
8.511 pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . 477
8.511.1 Detailed Description . . . 477
8.511.2 Field Documentation . . . 478
8.511.2.1 currentDataBearer . . . 478
8.511.2.2 dataBearer . . . 478
8.511.2.3 dataBearerTechExt . . . 478
8.511.2.4 dataSystemStatus . . . 478
8.511.2.5 dormancyStatus . . . 478
8.511.2.6 interval . . . 478
8.511.2.7 mobileIP . . . 478
8.511.2.8 transferStats . . . 478
8.512 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 478
8.512.1 Detailed Description . . . 478
8.512.2 Field Documentation . . . 478
8.512.2.1 pProfileId . . . 478
8.513 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference . . . 478
8.513.1 Detailed Description . . . 478
8.513.2 Field Documentation . . . 479
8.513.2.1 pHwConfig . . . 479
8.513.2.2 pProfileId . . . 479
8.513.2.3 pRequestOptionList . . . 479
8.514 pack_wds_SLQSSSetLoopback_t Struct Reference . . . 479
8.514.1 Detailed Description . . . 479
8.514.2 Field Documentation . . . 479
8.514.2.1 loopbackMode . . . 479

8.514.2.2 loopbackMultiplier . . . 479
8.515 pack_wds_SLQSStartDataSession_t Struct Reference . . . 479
8.515.1 Detailed Description . . . 480
8.515.2 Field Documentation . . . 480
8.515.2.1 pAuth . . . 480
8.515.2.2 pPass . . . 480
8.515.2.3 pprofileid3gpp . . . 480
8.515.2.4 pprofileid3gpp2 . . . 480
8.515.2.5 pTech . . . 480
8.515.2.6 pUser . . . 481
8.516 pack_wds_SLQSStopDataSession_t Struct Reference . . . 481
8.516.1 Detailed Description . . . 481
8.516.2 Field Documentation . . . 481
8.516.2.1 psid . . . 481
8.517 pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . 481
8.517.1 Detailed Description . . . 481
8.517.2 Field Documentation . . . 482
8.517.2.1 pCurrChannelRateInd . . . 482
8.517.2.2 pCurrDataBearerTechInd . . . 482
8.517.2.3 pCurrPrefDataSysInd . . . 482
8.517.2.4 pDataBearerTechInd . . . 482
8.517.2.5 pDataCallStatusChangeInd . . . 482
8.517.2.6 pDataSystemStatusChangeInd . . . 483
8.517.2.7 pDormancyStatusInd . . . 483
8.517.2.8 pEVDOPageMonPerChangeInd . . . 483
8.517.2.9 pMIPStatusInd . . . 483
8.517.2.10 pTransferStatInd . . . 483
8.518 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 483
8.518.1 Detailed Description . . . 483
8.518.2 Field Documentation . . . 483
8.518.2.1 contextId . . . 483
8.518.2.2 contextType . . . 483
8.519 PackCreateProfileOut Struct Reference . . . 483
8.519.1 Detailed Description . . . 483
8.519.2 Field Documentation . . . 484
8.519.2.1 ExtErrorCode . . . 484
8.519.2.2 ProfileIndex . . . 484
8.519.2.3 ProfileType . . . 484
8.520 packgetDyingGaspCfg Struct Reference . . . 484
8.520.1 Detailed Description . . . 484

8.520.2 Field Documentation . . . 484
8.520.2.1 pDestSMSContent . . . 484
8.520.2.2 pDestSMSNum . . . 484
8.521 packgetDyingGaspStatistics Struct Reference . . . 484
8.521.1 Detailed Description . . . 485
8.521.2 Field Documentation . . . 485
8.521.2.1 pSMSAttemptedFlag . . . 485
8.521.2.2 pTimeStamp . . . 485
8.522 PackSwiAvmsSetSettingsAPNInfo Struct Reference . . . 485
8.522.1 Detailed Description . . . 485
8.522.2 Field Documentation . . . 485
8.522.2.1 bAPNLength . . . 485
8.522.2.2 bPWDLength . . . 486
8.522.2.3 bUnameLength . . . 486
8.522.2.4 szAPN . . . 486
8.522.2.5 szPWD . . . 486
8.522.2.6 szUname . . . 486
8.523 PackSwiAvmsSetSettingsConnectionRetryTimers Struct Reference . . . 486
8.523.1 Detailed Description . . . 486
8.523.2 Field Documentation . . . 486
8.523.2.1 Timers . . . 486
8.524 PackSwiAvmsSetSettingsPeriodInfo Struct Reference . . . 487
8.524.1 Detailed Description . . . 487
8.524.2 Field Documentation . . . 487
8.524.2.1 ulMax . . . 487
8.524.2.2 ulMin . . . 487
8.525 PackSwiAVMSSettingsAPNInfo Struct Reference . . . 487
8.525.1 Detailed Description . . . 487
8.525.2 Field Documentation . . . 487
8.525.2.1 bAPNLength . . . 487
8.525.2.2 bPWDLength . . . 487
8.525.2.3 bUnameLength . . . 488
8.525.2.4 pAPN . . . 488
8.525.2.5 pPWD . . . 488
8.525.2.6 pUname . . . 488
8.526 PackSwiAVMSSettingsConnectionRetryTimers Struct Reference . . . 488
8.526.1 Detailed Description . . . 488
8.526.2 Field Documentation . . . 488
8.526.2.1 Timers . . . 488
8.527 PackSwiAVMSSettingsPeriodsInfo Struct Reference . . . 488

8.527.1 Detailed Description . . . 489
8.527.2 Field Documentation . . . 489
8.527.2.1 max . . . 489
8.527.2.2 min . . . 489
8.528 qmiSmsMessageList Struct Reference . . . 489
8.528.1 Detailed Description . . . 489
8.528.2 Field Documentation . . . 489
8.528.2.1 messageIndex . . . 489
8.528.2.2 messageTag . . . 489
8.529 qmiWDSDataBearerTechnology Struct Reference . . . 489
8.529.1 Detailed Description . . . 490
8.529.2 Field Documentation . . . 490
8.529.2.1 currentNetwork . . . 490
8.529.2.2 ratMask . . . 490
8.529.2.3 soMask . . . 490
8.530 qos_BindDataPortMuxID_t Struct Reference . . . 490
8.530.1 Detailed Description . . . 490
8.530.2 Field Documentation . . . 490
8.530.2.1 MuxID . . . 490
8.531 qos_BindDataPortPeripheralEndPointID_t Struct Reference . . . 490
8.531.1 Detailed Description . . . 491
8.531.2 Field Documentation . . . 491
8.531.2.1 EndPointType . . . 491
8.531.2.2 IfaceID . . . 491
8.532 qos_BindDataPortSIODataPort_t Struct Reference . . . 491
8.532.1 Detailed Description . . . 491
8.532.2 Field Documentation . . . 491
8.532.2.1 SIODataPort . . . 491
8.533 RFBandInfoElements Struct Reference . . . 491
8.533.1 Detailed Description . . . 491
8.533.2 Field Documentation . . . 492
8.533.2.1 activeBandClass . . . 492
8.533.2.2 activeChannel . . . 492
8.533.2.3 radioInterface . . . 492
8.534 rmTrasnferStaticsReq Struct Reference . . . 492
8.534.1 Detailed Description . . . 492
8.534.2 Field Documentation . . . 493
8.534.2.1 bResetStatistics . . . 493
8.534.2.2 ulMask . . . 493
8.535 sensorData_t Struct Reference . . . 493

8.535.1 Detailed Description . . . 493
8.535.2 Field Documentation . . . 494
8.535.2.1 flags . . . 494
8.535.2.2 sensorDataLen . . . 494
8.535.2.3 timeOfFirstSample . . . 494
8.535.2.4 timeOffset . . . 494
8.535.2.5 xAxis . . . 494
8.535.2.6 yAxis . . . 494
8.535.2.7 zAxis . . . 494
8.536 slot_t Struct Reference . . . 494
8.536.1 Detailed Description . . . 494
8.536.2 Field Documentation . . . 495
8.536.2.1 bICCID . . . 495
8.536.2.2 bICCIDLength . . . 495
8.536.2.3 bLogicalSlot . . . 495
8.536.2.4 uPhyCardStatus . . . 495
8.536.2.5 uPhySlotStatus . . . 495
8.537 slotInf Struct Reference . . . 495
8.537.1 Detailed Description . . . 495
8.537.2 Field Documentation . . . 496
8.537.2.1 AppStatus . . . 496
8.537.2.2 cardState . . . 497
8.537.2.3 errorState . . . 497
8.537.2.4 numApp . . . 497
8.537.2.5 upinRetries . . . 497
8.537.2.6 upinState . . . 497
8.537.2.7 upukRetries . . . 497
8.538 slots_t Struct Reference . . . 497
8.538.1 Detailed Description . . . 497
8.538.2 Field Documentation . . . 497
8.538.2.1 uimSlotStatus . . . 497
8.539 sms_BroadcastConfig Struct Reference . . . 497
8.539.1 Detailed Description . . . 497
8.539.2 Field Documentation . . . 498
8.539.2.1 fromServiceId . . . 498
8.539.2.2 selected . . . 498
8.539.2.3 toServiceId . . . 498
8.540 sms_CDMABroadcastConfig Struct Reference . . . 498
8.540.1 Detailed Description . . . 498
8.540.2 Field Documentation . . . 498

8.540.2.1 language . . . 498
8.540.2.2 selected . . . 498
8.540.2.3 serviceCategory . . . 498
8.541 sms_getIndicationReg Struct Reference . . . 498
8.541.1 Detailed Description . . . 499
8.541.2 Field Documentation . . . 499
8.541.2.1 pRegCallStatInfoEvt . . . 499
8.541.2.2 pRegTransLayerInfoEvt . . . 499
8.541.2.3 pRegTransNWRegInfoEvt . . . 499
8.542 sms_getMsgWaitingInfo Struct Reference . . . 499
8.542.1 Detailed Description . . . 499
8.542.2 Field Documentation . . . 500
8.542.2.1 msgWaitInfo . . . 500
8.542.2.2 numInstances . . . 500
8.543 sms_getTransLayerInfo Struct Reference . . . 500
8.543.1 Detailed Description . . . 500
8.543.2 Field Documentation . . . 500
8.543.2.1 pRegInd . . . 500
8.543.2.2 pTransLayerInfo . . . 500
8.544 sms_getTransNWRegInfo Struct Reference . . . 500
8.544.1 Detailed Description . . . 501
8.544.2 Field Documentation . . . 501
8.544.2.1 pRegStatus . . . 501
8.545 sms_maxStorageSizeReq Struct Reference . . . 501
8.545.1 Detailed Description . . . 501
8.545.2 Field Documentation . . . 502
8.545.2.1 pMessageMode . . . 502
8.545.2.2 storageType . . . 502
8.546 sms_maxStorageSizeResp Struct Reference . . . 502
8.546.1 Detailed Description . . . 502
8.546.2 Field Documentation . . . 502
8.546.2.1 freeSlots . . . 502
8.546.2.2 maxStorageSize . . . 502
8.547 sms_messageWaitingInfoContent Struct Reference . . . 502
8.547.1 Detailed Description . . . 502
8.547.2 Field Documentation . . . 503
8.547.2.1 activeInd . . . 503
8.547.2.2 msgCount . . . 503
8.547.2.3 msgType . . . 503
8.548 sms_msgProtocolResp Struct Reference . . . 503

8.548.1 Detailed Description . . . 503
8.548.2 Field Documentation . . . 503
8.548.2.1 msgProtocol . . . 503
8.549 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . 503
8.549.1 Detailed Description . . . 504
8.549.2 Field Documentation . . . 504
8.549.2.1 activated_ind . . . 504
8.549.2.2 CDMABroadcastConfig . . . 504
8.549.2.3 num_instances . . . 504
8.550 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . 504
8.550.1 Detailed Description . . . 504
8.550.2 Field Documentation . . . 505
8.550.2.1 activated_ind . . . 505
8.550.2.2 broadcastConfig . . . 505
8.550.2.3 num_instances . . . 505
8.551 sms_routeEntry Struct Reference . . . 505
8.551.1 Detailed Description . . . 505
8.551.2 Field Documentation . . . 506
8.551.2.1 messageClass . . . 506
8.551.2.2 messageType . . . 506
8.551.2.3 receiptAction . . . 506
8.551.2.4 routeStorage . . . 506
8.552 sms_sendAsyncsmsParams Struct Reference . . . 506
8.552.1 Detailed Description . . . 507
8.552.2 Field Documentation . . . 508
8.552.2.1 messageFormat . . . 508
8.552.2.2 messageSize . . . 508
8.552.2.3 pFollowOnDC . . . 508
8.552.2.4 pForceOnDC . . . 508
8.552.2.5 pLinktimer . . . 508
8.552.2.6 pMessage . . . 508
8.552.2.7 pRetryMessage . . . 508
8.552.2.8 pRetryMessageId . . . 508
8.552.2.9 pServiceOption . . . 508
8.552.2.10 pSmsOnIms . . . 508
8.552.2.11 pUserData . . . 508
8.553 sms_setIndicationReg Struct Reference . . . 508
8.553.1 Detailed Description . . . 509
8.553.2 Field Documentation . . . 509
8.553.2.1 pRegCallStatInfoEvt . . . 509

8.553.2.2 pRegTransLayerInfoEvt . . . 509
8.553.2.3 pRegTransNWRegInfoEvt . . . 509
8.554 sms_setRoutesReq Struct Reference . . . 509
8.554.1 Detailed Description . . . 510
8.554.2 Field Documentation . . . 510
8.554.2.1 numOfRoutes . . . 510
8.554.2.2 pTransferStatusReport . . . 510
8.554.2.3 routeList . . . 510
8.555 sms_transLayerInfo Struct Reference . . . 510
8.555.1 Detailed Description . . . 510
8.555.2 Field Documentation . . . 510
8.555.2.1 TransCap . . . 510
8.555.2.2 TransType . . . 510
8.556 sMSCAddressInfo Struct Reference . . . 511
8.556.1 Detailed Description . . . 511
8.556.2 Field Documentation . . . 511
8.556.2.1 data . . . 511
8.556.2.2 length . . . 511
8.557 sMSCAddressTlv Struct Reference . . . 511
8.557.1 Detailed Description . . . 511
8.557.2 Field Documentation . . . 511
8.557.2.1 SMSCInfo . . . 511
8.557.2.2 TlvPresent . . . 511
8.558 sMSEtwsMessageInfo Struct Reference . . . 512
8.558.1 Detailed Description . . . 512
8.558.2 Field Documentation . . . 512
8.558.2.1 data . . . 512
8.558.2.2 length . . . 512
8.558.2.3 notificationType . . . 512
8.559 sMSEtwsMessageTlv Struct Reference . . . 512
8.559.1 Detailed Description . . . 512
8.559.2 Field Documentation . . . 513
8.559.2.1 EtwsMessageInfo . . . 513
8.559.2.2 TlvPresent . . . 513
8.560 sMSEtwsPlmnInfo Struct Reference . . . 513
8.560.1 Detailed Description . . . 513
8.560.2 Field Documentation . . . 513
8.560.2.1 mobileCountryCode . . . 513
8.560.2.2 mobileNetworkCode . . . 513
8.561 sMSMessageModeInfo Struct Reference . . . 513

8.561.1 Detailed Description . . . 513
8.561.2 Field Documentation . . . 513
8.561.2.1 messageMode . . . 513
8.562 sMSMTMessageInfo Struct Reference . . . 513
8.562.1 Detailed Description . . . 514
8.562.2 Field Documentation . . . 514
8.562.2.1 messageIndex . . . 514
8.562.2.2 storageType . . . 514
8.563 sMSOnIMSInfo Struct Reference . . . 514
8.563.1 Detailed Description . . . 514
8.563.2 Field Documentation . . . 514
8.563.2.1 smsOnIMS . . . 514
8.564 sMSOnIMSTlv Struct Reference . . . 514
8.564.1 Detailed Description . . . 514
8.564.2 Field Documentation . . . 515
8.564.2.1 IMSInfo . . . 515
8.564.2.2 TlvPresent . . . 515
8.565 sMSTransferRouteMTMessageInfo Struct Reference . . . 515
8.565.1 Detailed Description . . . 515
8.565.2 Field Documentation . . . 515
8.565.2.1 ackIndicator . . . 515
8.565.2.2 data . . . 516
8.565.2.3 format . . . 516
8.565.2.4 length . . . 516
8.565.2.5 transactionID . . . 516
8.566 swi_uint256_t Struct Reference . . . 516
8.566.1 Detailed Description . . . 516
8.566.2 Field Documentation . . . 516
8.566.2.1 word . . . 516
8.567 swiaudio_PCMparams Struct Reference . . . 516
8.567.1 Detailed Description . . . 516
8.567.2 Field Documentation . . . 516
8.567.2.1 iFaceTab . . . 516
8.567.2.2 iFaceTabLen . . . 516
8.568 swidms_ehrpdMTUSizeTlv Struct Reference . . . 516
8.568.1 Detailed Description . . . 517
8.568.2 Field Documentation . . . 517
8.568.2.1 ehrpdMTUSize . . . 517
8.568.2.2 TlvPresent . . . 517
8.569 swidms_hrpdMTUSizeTlv Struct Reference . . . 517

8.569.1 Detailed Description . . . 517
8.569.2 Field Documentation . . . 517
8.569.2.1 hrpdMTUSize . . . 517
8.569.2.2 TlvPresent . . . 517
8.570 swidms_mtuSize3gppTlv Struct Reference . . . 517
8.570.1 Detailed Description . . . 518
8.570.2 Field Documentation . . . 518
8.570.2.1 MTUSize3gpp . . . 518
8.570.2.2 TlvPresent . . . 518
8.571 swidms_usbMTUSizeTlv Struct Reference . . . 518
8.571.1 Detailed Description . . . 518
8.571.2 Field Documentation . . . 518
8.571.2.1 TlvPresent . . . 518
8.571.2.2 UsbMTUSize . . . 518
8.572 tdscdmaSigInfoExt Struct Reference . . . 518
8.572.1 Detailed Description . . . 519
8.572.2 Field Documentation . . . 519
8.572.2.1 ecio . . . 519
8.572.2.2 rscp . . . 519
8.572.2.3 rssi . . . 519
8.572.2.4 sinr . . . 519
8.573 tempData_t Struct Reference . . . 519
8.573.1 Detailed Description . . . 519
8.573.2 Field Documentation . . . 520
8.573.2.1 temperature . . . 520
8.573.2.2 temperatureDataLen . . . 520
8.573.2.3 timeOfFirstSample . . . 520
8.573.2.4 timeOffset . . . 520
8.573.2.5 timeSource . . . 520
8.574 tmd_mitigationDevList Struct Reference . . . 520
8.574.1 Detailed Description . . . 520
8.574.2 Field Documentation . . . 521
8.574.2.1 maxMitigationLevel . . . 521
8.574.2.2 mitigationDevId . . . 521
8.574.2.3 mitigationDevIdLen . . . 521
8.575 transferRouteMessageTlv Struct Reference . . . 521
8.575.1 Detailed Description . . . 521
8.575.2 Field Documentation . . . 521
8.575.2.1 TlvPresent . . . 521
8.575.2.2 TransferRouteMTMessageInfo . . . 521

8.576 uim_appStatus Struct Reference . . . 521
8.576.1 Detailed Description . . . 522
8.576.2 Field Documentation . . . 524
8.576.2.1 aidLength . . . 524
8.576.2.2 aidVal . . . 524
8.576.2.3 appState . . . 524
8.576.2.4 appType . . . 524
8.576.2.5 persoFeature . . . 524
8.576.2.6 persoRetries . . . 524
8.576.2.7 persoState . . . 524
8.576.2.8 persoUnblockRetries . . . 524
8.576.2.9 pin1Retries . . . 524
8.576.2.10 pin1State . . . 524
8.576.2.11 pin2Retries . . . 524
8.576.2.12 pin2State . . . 524
8.576.2.13 puk1Retries . . . 524
8.576.2.14 puk2Retries . . . 524
8.576.2.15 univPin . . . 524
8.577 uim_authenticateResult Struct Reference . . . 524
8.577.1 Detailed Description . . . 525
8.577.2 Field Documentation . . . 525
8.577.2.1 content . . . 525
8.577.2.2 contentLen . . . 525
8.578 uim_authenticationData Struct Reference . . . 525
8.578.1 Detailed Description . . . 525
8.578.2 Field Documentation . . . 526
8.578.2.1 context . . . 526
8.578.2.2 data . . . 526
8.578.2.3 dataLen . . . 526
8.579 uim_cardResult Struct Reference . . . 526
8.579.1 Detailed Description . . . 526
8.579.2 Field Documentation . . . 527
8.579.2.1 sw1 . . . 527
8.579.2.2 sw2 . . . 527
8.580 uim_cardStatus Struct Reference . . . 527
8.580.1 Detailed Description . . . 527
8.580.2 Field Documentation . . . 528
8.580.2.1 index1xPri . . . 528
8.580.2.2 index1xSec . . . 528
8.580.2.3 indexGwPri . . . 528

8.580.2.4 indexGwSec . . . 528
8.580.2.5 numSlot . . . 528
8.580.2.6 SlotInfo . . . 528
8.581 uim_changeUIMPIN Struct Reference . . . 528
8.581.1 Detailed Description . . . 528
8.581.2 Field Documentation . . . 529
8.581.2.1 oldPINLen . . . 529
8.581.2.2 oldPINVal . . . 529
8.581.2.3 pinID . . . 529
8.581.2.4 pinLen . . . 529
8.581.2.5 pinVal . . . 529
8.582 uim_depersonalizationInformation Struct Reference . . . 529
8.582.1 Detailed Description . . . 529
8.582.2 Field Documentation . . . 530
8.582.2.1 ckLen . . . 530
8.582.2.2 ckVal . . . 530
8.582.2.3 feature . . . 530
8.582.2.4 operation . . . 530
8.583 uim_encryptedPIN1 Struct Reference . . . 530
8.583.1 Detailed Description . . . 530
8.583.2 Field Documentation . . . 531
8.583.2.1 pin1Len . . . 531
8.583.2.2 pin1Val . . . 531
8.584 uim_fileAttributes Struct Reference . . . 531
8.584.1 Detailed Description . . . 531
8.584.2 Field Documentation . . . 534
8.584.2.1 fileID . . . 534
8.584.2.2 fileSize . . . 534
8.584.2.3 fileType . . . 534
8.584.2.4 rawLen . . . 534
8.584.2.5 rawValue . . . 534
8.584.2.6 recordCount . . . 534
8.584.2.7 recordSize . . . 534
8.584.2.8 secActivate . . . 534
8.584.2.9 secActivateMask . . . 534
8.584.2.10 secDeactivate . . . 534
8.584.2.11 secDeactivateMask . . . 534
8.584.2.12 secIncrease . . . 534
8.584.2.13 secIncreaseMask . . . 534
8.584.2.14 secRead . . . 534

8.584.2.15 secReadMask . . . 534
8.584.2.16 secWrite . . . 534
8.584.2.17 secWriteMask . . . 534
8.585 uim_fileInfo Struct Reference . . . 534
8.585.1 Detailed Description . . . 534
8.585.2 Field Documentation . . . 535
8.585.2.1 fileID . . . 535
8.585.2.2 path . . . 535
8.585.2.3 pathLen . . . 535
8.586 uim_hotSwapStatus Struct Reference . . . 535
8.586.1 Detailed Description . . . 535
8.586.2 Field Documentation . . . 535
8.586.2.1 hotSwap . . . 535
8.586.2.2 hotSwapLength . . . 535
8.587 uim_personalizationStatus Struct Reference . . . 535
8.587.1 Detailed Description . . . 536
8.587.2 Field Documentation . . . 536
8.587.2.1 feature . . . 536
8.587.2.2 numFeatures . . . 536
8.587.2.3 unblockLeft . . . 536
8.587.2.4 verifyLeft . . . 536
8.588 uim_readResult Struct Reference . . . 536
8.588.1 Detailed Description . . . 537
8.588.2 Field Documentation . . . 537
8.588.2.1 content . . . 537
8.588.2.2 contentLen . . . 537
8.589 uim_readTransparentInfo Struct Reference . . . 537
8.589.1 Detailed Description . . . 537
8.589.2 Field Documentation . . . 537
8.589.2.1 length . . . 537
8.589.2.2 offset . . . 537
8.590 uim_refreshevent Struct Reference . . . 537
8.590.1 Detailed Description . . . 538
8.590.2 Field Documentation . . . 539
8.590.2.1 aid . . . 539
8.590.2.2 aidLength . . . 539
8.590.2.3 arrfileInfo . . . 539
8.590.2.4 mode . . . 539
8.590.2.5 numOfFiles . . . 539
8.590.2.6 sessionType . . . 539

8.590.2.7 stage . . . 539
8.591 uim_registerRefresh Struct Reference . . . 539
8.591.1 Detailed Description . . . 539
8.591.2 Field Documentation . . . 540
8.591.2.1 arrfileInfo . . . 540
8.591.2.2 numFiles . . . 540
8.591.2.3 registerFlag . . . 540
8.591.2.4 voteForInit . . . 540
8.592 uim_remainingRetries Struct Reference . . . 540
8.592.1 Detailed Description . . . 540
8.592.2 Field Documentation . . . 540
8.592.2.1 unblockLeft . . . 540
8.592.2.2 verifyLeft . . . 541
8.593 uim_sessionInformation Struct Reference . . . 541
8.593.1 Detailed Description . . . 541
8.593.2 Field Documentation . . . 541
8.593.2.1 aid . . . 541
8.593.2.2 aidLength . . . 541
8.593.2.3 sessionType . . . 541
8.594 uim_setPINProtection Struct Reference . . . 541
8.594.1 Detailed Description . . . 542
8.594.2 Field Documentation . . . 542
8.594.2.1 pinID . . . 542
8.594.2.2 pinLength . . . 542
8.594.2.3 pinOperation . . . 542
8.594.2.4 pinValue . . . 542
8.595 uim_slotInfo Struct Reference . . . 542
8.595.1 Detailed Description . . . 543
8.595.2 Field Documentation . . . 544
8.595.2.1 AppStatus . . . 544
8.595.2.2 cardState . . . 544
8.595.2.3 errorState . . . 544
8.595.2.4 numApp . . . 544
8.595.2.5 upinRetries . . . 544
8.595.2.6 upinState . . . 544
8.595.2.7 upukRetries . . . 544
8.596 uim_UIMSessionInformation Struct Reference . . . 544
8.596.1 Detailed Description . . . 544
8.596.2 Field Documentation . . . 545
8.596.2.1 aid . . . 545

8.596.2.2 aidLength . . . 545
8.596.2.3 sessionType . . . 545
8.597 uim_unblockUIMPIN Struct Reference . . . 545
8.597.1 Detailed Description . . . 545
8.597.2 Field Documentation . . . 545
8.597.2.1 newPINLen . . . 545
8.597.2.2 newPINVal . . . 545
8.597.2.3 pinID . . . 545
8.597.2.4 pukLen . . . 546
8.597.2.5 pukVal . . . 546
8.598 uim_verifyUIMPIN Struct Reference . . . 546
8.598.1 Detailed Description . . . 546
8.598.2 Field Documentation . . . 546
8.598.2.1 pinID . . . 546
8.598.2.2 pinLen . . . 546
8.598.2.3 pinVal . . . 546
8.599 unpack_audio_SLQSGetAudioPathConfig_t Struct Reference . . . 546
8.599.1 Detailed Description . . . 547
8.599.2 Field Documentation . . . 548
8.599.2.1 ParamPresenceMask . . . 548
8.599.2.2 pCodecSTGain . . . 548
8.599.2.3 pDTMFTXGain . . . 548
8.599.2.4 pECMode . . . 548
8.599.2.5 pMICGainSelect . . . 548
8.599.2.6 pNSEnable . . . 548
8.599.2.7 pRXAGCList . . . 548
8.599.2.8 pRXAVCAGCSwitch . . . 548
8.599.2.9 pRXAVCList . . . 548
8.599.2.10 pRXPCMIIRFltr . . . 548
8.599.2.11 pTXAGCList . . . 548
8.599.2.12 pTXAVCSwitch . . . 548
8.599.2.13 pTXGain . . . 548
8.599.2.14 pTXPCMIIRFltr . . . 549
8.600 unpack_audio_SLQSGetAudioProfile_t Struct Reference . . . 549
8.600.1 Detailed Description . . . 549
8.600.2 Field Documentation . . . 549
8.600.2.1 EarMute . . . 550
8.600.2.2 MicMute . . . 550
8.600.2.3 ParamPresenceMask . . . 550
8.600.2.4 Profile . . . 550

8.600.2.5 Volume . . . 550
8.601 unpack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference . . . 550
8.601.1 Detailed Description . . . 550
8.601.2 Field Documentation . . . 550
8.601.2.1 ParamPresenceMask . . . 550
8.601.2.2 ResCode . . . 550
8.602 unpack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference . . . 550
8.602.1 Detailed Description . . . 550
8.602.2 Field Documentation . . . 551
8.602.2.1 ParamPresenceMask . . . 551
8.602.2.2 ResCode . . . 551
8.603 unpack_cat_SetCatEventCallback_ind_t Struct Reference . . . 551
8.603.1 Detailed Description . . . 551
8.603.2 Field Documentation . . . 551
8.603.2.1 CCETlv . . . 551
8.603.2.2 event_Index . . . 551
8.603.2.3 ParamPresenceMask . . . 551
8.604 unpack_cat_SetCATEventCallback_t Struct Reference . . . 551
8.604.1 Detailed Description . . . 552
8.604.2 Field Documentation . . . 552
8.604.2.1 errorMask . . . 552
8.604.2.2 ParamPresenceMask . . . 552
8.604.2.3 Tlvresult . . . 552
8.605 unpack_dms_GetActivationState_t Struct Reference . . . 552
8.605.1 Detailed Description . . . 552
8.605.2 Field Documentation . . . 553
8.605.2.1 ParamPresenceMask . . . 553
8.605.2.2 state . . . 553
8.606 unpack_dms_GetBandCapability_t Struct Reference . . . 553
8.606.1 Detailed Description . . . 553
8.606.2 Field Documentation . . . 555
8.606.2.1 BandCapability . . . 555
8.606.2.2 ParamPresenceMask . . . 555
8.606.2.3 Tlvresult . . . 555
8.607 unpack_dms_GetCrashAction_t Struct Reference . . . 555
8.607.1 Detailed Description . . . 555
8.607.2 Field Documentation . . . 555
8.607.2.1 DevCrashState . . . 555
8.607.2.2 ParamPresenceMask . . . 556
8.607.2.3 Tlvresult . . . 556

8.608 unpack_dms_GetCustFeature_t Struct Reference . . . 556
8.608.1 Detailed Description . . . 556
8.608.2 Field Documentation . . . 558
8.608.2.1 DHCPRelayEnabled . . . 558
8.608.2.2 DisableIMSI . . . 558
8.608.2.3 GpsEnable . . . 558
8.608.2.4 GPSLPM . . . 558
8.608.2.5 GPSSel . . . 558
8.608.2.6 IPFamSupport . . . 558
8.608.2.7 IsVoiceEnabled . . . 558
8.608.2.8 ParamPresenceMask . . . 558
8.608.2.9 RMAutoConnect . . . 558
8.608.2.10 SMSSupport . . . 558
8.608.2.11 Tlvresult . . . 558
8.609 unpack_dms_GetCustFeaturesV2_t Struct Reference . . . 558
8.609.1 Detailed Description . . . 558
8.609.2 Field Documentation . . . 559
8.609.2.1 GetCustomFeatureV2 . . . 559
8.609.2.2 ParamPresenceMask . . . 559
8.609.2.3 Tlvresult . . . 559
8.610 unpack_dms_GetDeviceCap_t Struct Reference . . . 559
8.610.1 Detailed Description . . . 559
8.610.2 Field Documentation . . . 560
8.610.2.1 DataServiceCapability . . . 560
8.610.2.2 MaxRXChannelRate . . . 560
8.610.2.3 MaxTXChannelRate . . . 560
8.610.2.4 ParamPresenceMask . . . 560
8.610.2.5 RadioIfaces . . . 560
8.610.2.6 RadioIfacesSize . . . 560
8.610.2.7 SimCapability . . . 560
8.610.2.8 Tlvresult . . . 560
8.611 unpack_dms_GetDeviceCapabilities_t Struct Reference . . . 560
8.611.1 Detailed Description . . . 561
8.611.2 Field Documentation . . . 561
8.611.2.1 dataServiceCaCapability . . . 562
8.611.2.2 maxRxChannelRate . . . 562
8.611.2.3 maxTxChannelRate . . . 562
8.611.2.4 ParamPresenceMask . . . 562
8.611.2.5 RadioIfaces . . . 562
8.611.2.6 radioIfacesSize . . . 562

8.611.2.7 simCapability . . . 562
8.612 unpack_dms_GetDeviceHardwareRev_t Struct Reference . . . 562
8.612.1 Detailed Description . . . 562
8.612.2 Field Documentation . . . 562
8.612.2.1 ParamPresenceMask . . . 562
8.612.2.2 String . . . 562
8.612.2.3 stringSize . . . 562
8.612.2.4 Tlvresult . . . 563
8.613 unpack_dms_GetDeviceMfr_t Struct Reference . . . 563
8.613.1 Detailed Description . . . 563
8.613.2 Field Documentation . . . 563
8.613.2.1 ParamPresenceMask . . . 563
8.613.2.2 String . . . 563
8.613.2.3 stringSize . . . 563
8.613.2.4 Tlvresult . . . 563
8.614 unpack_dms_GetDeviceSerialNumbers_t Struct Reference . . . 563
8.614.1 Detailed Description . . . 564
8.614.2 Field Documentation . . . 565
8.614.2.1 esnSize . . . 565
8.614.2.2 ESNString . . . 565
8.614.2.3 imeiSize . . . 565
8.614.2.4 IMEIString . . . 565
8.614.2.5 imeiSvnSize . . . 565
8.614.2.6 ImeiSvnString . . . 565
8.614.2.7 meidSize . . . 565
8.614.2.8 MEIDString . . . 565
8.614.2.9 ParamPresenceMask . . . 565
8.614.2.10 Tlvresult . . . 565
8.615 unpack_dms_GetFirmwareInfo_t Struct Reference . . . 565
8.615.1 Detailed Description . . . 566
8.615.2 Field Documentation . . . 566
8.615.2.1 appversion_str . . . 566
8.615.2.2 bootversion_str . . . 566
8.615.2.3 carrier_str . . . 567
8.615.2.4 cur_carr_name . . . 567
8.615.2.5 cur_carr_rev . . . 567
8.615.2.6 modelid_str . . . 567
8.615.2.7 packageid_str . . . 567
8.615.2.8 ParamPresenceMask . . . 567
8.615.2.9 priversion_str . . . 567

8.615.2.10 sku_str . . . 567
8.615.2.11 Tlvresult . . . 567
8.616 unpack_dms_GetFirmwareRevision_t Struct Reference . . . 567
8.616.1 Detailed Description . . . 567
8.616.2 Field Documentation . . . 568
8.616.2.1 amssSize . . . 568
8.616.2.2 AMSSString . . . 568
8.616.2.3 ParamPresenceMask . . . 568
8.616.2.4 PRIString . . . 568
8.616.2.5 Tlvresult . . . 568
8.617 unpack_dms_GetFirmwareRevisions_t Struct Reference . . . 568
8.617.1 Detailed Description . . . 568
8.617.2 Field Documentation . . . 569
8.617.2.1 amssSize . . . 569
8.617.2.2 AMSSString . . . 569
8.617.2.3 bootSize . . . 569
8.617.2.4 BootString . . . 569
8.617.2.5 ParamPresenceMask . . . 569
8.617.2.6 priSize . . . 569
8.617.2.7 PRIString . . . 569
8.617.2.8 Tlvresult . . . 569
8.618 unpack_dms_GetFSN_t Struct Reference . . . 569
8.618.1 Detailed Description . . . 569
8.618.2 Field Documentation . . . 570
8.618.2.1 ParamPresenceMask . . . 570
8.618.2.2 String . . . 570
8.618.2.3 Tlvresult . . . 570
8.619 unpack_dms_GetHardwareRevision_t Struct Reference . . . 570
8.619.1 Detailed Description . . . 570
8.619.2 Field Documentation . . . 570
8.619.2.1 hwVer . . . 570
8.619.2.2 ParamPresenceMask . . . 570
8.620 unpack_dms_GetIMSI_t Struct Reference . . . 570
8.620.1 Detailed Description . . . 570
8.620.2 Field Documentation . . . 571
8.620.2.1 imsi . . . 571
8.620.2.2 ParamPresenceMask . . . 571
8.620.2.3 Tlvresult . . . 571
8.621 unpack_dms_GetManufacturer_t Struct Reference . . . 571
8.621.1 Detailed Description . . . 571

8.621.2 Field Documentation . . . 571
8.621.2.1 manufacturer . . . 571
8.621.2.2 ParamPresenceMask . . . 571
8.621.2.3 Tlvresult . . . 571
8.622 unpack_dms_GetModelID_t Struct Reference . . . 572
8.622.1 Detailed Description . . . 572
8.622.2 Field Documentation . . . 572
8.622.2.1 modelid . . . 572
8.622.2.2 ParamPresenceMask . . . 572
8.622.2.3 Tlvresult . . . 572
8.623 unpack_dms_GetNetworkTime_t Struct Reference . . . 572
8.623.1 Detailed Description . . . 572
8.623.2 Field Documentation . . . 573
8.623.2.1 ParamPresenceMask . . . 573
8.623.2.2 source . . . 573
8.623.2.3 timestamp . . . 573
8.623.2.4 Tlvresult . . . 573
8.624 unpack_dms_GetOfflineReason_t Struct Reference . . . 573
8.624.1 Detailed Description . . . 573
8.624.2 Field Documentation . . . 574
8.624.2.1 ParamPresenceMask . . . 574
8.624.2.2 pbPlatform . . . 574
8.624.2.3 pReasonMask . . . 574
8.624.2.4 Tlvresult . . . 574
8.625 unpack_dms_GetPower_t Struct Reference . . . 574
8.625.1 Detailed Description . . . 574
8.625.2 Field Documentation . . . 575
8.625.2.1 HardwareControlledMode . . . 575
8.625.2.2 OfflineReason . . . 575
8.625.2.3 OperationMode . . . 575
8.625.2.4 ParamPresenceMask . . . 575
8.625.2.5 Tlvresult . . . 575
8.626 unpack_dms_GetPRLVersion_t Struct Reference . . . 575
8.626.1 Detailed Description . . . 575
8.626.2 Field Documentation . . . 576
8.626.2.1 ParamPresenceMask . . . 576
8.626.2.2 Tlvresult . . . 576
8.626.2.3 u16PRLVersion . . . 576
8.626.2.4 u8PRLPreference . . . 576
8.627 unpack_dms_GetSerialNumbers_t Struct Reference . . . 576

8.627.1 Detailed Description . . . 576
8.627.2 Field Documentation . . . 577
8.627.2.1 esn . . . 577
8.627.2.2 imei_no . . . 577
8.627.2.3 imeisv_svn . . . 577
8.627.2.4 meid . . . 577
8.627.2.5 ParamPresenceMask . . . 577
8.628 unpack_dms_GetUSBComp_t Struct Reference . . . 577
8.628.1 Detailed Description . . . 577
8.628.2 Field Documentation . . . 579
8.628.2.1 NumSupUSBComps . . . 579
8.628.2.2 ParamPresenceMask . . . 579
8.628.2.3 SupUSBComps . . . 579
8.628.2.4 Tlvresult . . . 579
8.628.2.5 USBComp . . . 579
8.629 unpack_dms_GetVoiceNumber_t Struct Reference . . . 579
8.629.1 Detailed Description . . . 580
8.629.2 Field Documentation . . . 580
8.629.2.1 MIN . . . 580
8.629.2.2 minSize . . . 580
8.629.2.3 ParamPresenceMask . . . 580
8.629.2.4 Tlvresult . . . 580
8.629.2.5 VoiceNumber . . . 580
8.629.2.6 voiceNumberSize . . . 580
8.630 unpack_dms_PSMCfgChange_ind_t Struct Reference . . . 581
8.630.1 Detailed Description . . . 581
8.630.2 Field Documentation . . . 581
8.630.2.1 ActiveTimer . . . 581
8.630.2.2 EnableState . . . 581
8.630.2.3 ParamPresenceMask . . . 581
8.630.2.4 PeriodicUpdateTimer . . . 581
8.630.2.5 Tlvresult . . . 581
8.631 unpack_dms_ResetToFactoryDefaults_t Struct Reference . . . 581
8.631.1 Detailed Description . . . 582
8.631.2 Field Documentation . . . 582
8.631.2.1 ParamPresenceMask . . . 582
8.631.2.2 Tlvresult . . . 582
8.632 unpack_dms_SetActivationStatusCallback_t Struct Reference . . . 582
8.632.1 Detailed Description . . . 582
8.632.2 Field Documentation . . . 582

8.632.2.1 ParamPresenceMask . . . 582
8.632.2.2 Tlvresult . . . 582
8.633 unpack_dms_SetCrashAction_t Struct Reference . . . 582
8.633.1 Detailed Description . . . 583
8.633.2 Field Documentation . . . 583
8.633.2.1 notused . . . 583
8.633.2.2 ParamPresenceMask . . . 583
8.634 unpack_dms_SetCustFeature_t Struct Reference . . . 583
8.634.1 Detailed Description . . . 583
8.634.2 Field Documentation . . . 583
8.634.2.1 ParamPresenceMask . . . 583
8.634.2.2 Tlvresult . . . 583
8.635 unpack_dms_SetCustFeaturesV2_t Struct Reference . . . 583
8.635.1 Detailed Description . . . 584
8.635.2 Field Documentation . . . 584
8.635.2.1 ParamPresenceMask . . . 584
8.635.2.2 Tlvresult . . . 584
8.636 unpack_dms_SetEventReport_ind_t Struct Reference . . . 584
8.636.1 Detailed Description . . . 584
8.636.2 Field Documentation . . . 584
8.636.2.1 ActivationStatusTlv . . . 584
8.636.2.2 OperatingModeTlv . . . 585
8.636.2.3 ParamPresenceMask . . . 585
8.636.2.4 Tlvresult . . . 585
8.637 unpack_dms_SetEventReport_t Struct Reference . . . 585
8.637.1 Detailed Description . . . 585
8.637.2 Field Documentation . . . 585
8.637.2.1 ParamPresenceMask . . . 585
8.637.2.2 Tlvresult . . . 585
8.638 unpack_dms_SetFirmwarePreference_t Struct Reference . . . 585
8.638.1 Detailed Description . . . 585
8.638.2 Field Documentation . . . 586
8.638.2.1 ParamPresenceMask . . . 586
8.638.2.2 Tlvresult . . . 586
8.639 unpack_dms_SetIndicationRegister_t Struct Reference . . . 586
8.639.1 Detailed Description . . . 586
8.639.2 Field Documentation . . . 586
8.639.2.1 ParamPresenceMask . . . 586
8.639.2.2 Tlvresult . . . 586
8.640 unpack_dms_SetPower_t Struct Reference . . . 586

8.640.1 Detailed Description . . . 586
8.640.2 Field Documentation . . . 587
8.640.2.1 ParamPresenceMask . . . 587
8.640.2.2 Tlvresult . . . 587
8.641 unpack_dms_SetUSBComp_t Struct Reference . . . 587
8.641.1 Detailed Description . . . 587
8.641.2 Field Documentation . . . 587
8.641.2.1 ParamPresenceMask . . . 587
8.641.2.2 Tlvresult . . . 587
8.642 unpack_dms_SLQSDmsSwiGetPCInfo_t Struct Reference . . . 587
8.642.1 Detailed Description . . . 587
8.642.2 Field Documentation . . . 589
8.642.2.1 has_LpmFlag . . . 589
8.642.2.2 has_PersistentLpm . . . 589
8.642.2.3 has_PowerOffMode . . . 589
8.642.2.4 has_Wdisable . . . 590
8.642.2.5 LpmFlag . . . 590
8.642.2.6 opMode . . . 590
8.642.2.7 ParamPresenceMask . . . 590
8.642.2.8 PersistentLpm . . . 590
8.642.2.9 PowerOffMode . . . 590
8.642.2.10 Wdisable . . . 590
8.643 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference . . . 590
8.643.1 Detailed Description . . . 590
8.643.2 Field Documentation . . . 591
8.643.2.1 ParamPresenceMask . . . 591
8.643.2.2 source . . . 591
8.643.2.3 Tlvresult . . . 591
8.643.2.4 type . . . 591
8.644 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . 591
8.644.1 Detailed Description . . . 591
8.644.2 Field Documentation . . . 592
8.644.2.1 ParamPresenceMask . . . 592
8.644.2.2 source . . . 592
8.644.2.3 Tlvresult . . . 592
8.644.2.4 type . . . 592
8.645 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . 592
8.645.1 Detailed Description . . . 592
8.645.2 Field Documentation . . . 593
8.645.2.1 ParamPresenceMask . . . 593

8.645.2.2 Tlvresult . . . 593
8.646 unpack_dms_SLQSGetBandCapability_t Struct Reference . . . 593
8.646.1 Detailed Description . . . 593
8.646.2 Field Documentation . . . 597
8.646.2.1 bandCapability . . . 597
8.646.2.2 is_LteBandCapability_Available . . . 597
8.646.2.3 is_TdsBandCapability_Available . . . 597
8.646.2.4 LteBandCapability . . . 597
8.646.2.5 ParamPresenceMask . . . 597
8.646.2.6 TdsBandCapability . . . 597
8.647 unpack_dms_SLQSGetERIFile_t Struct Reference . . . 597
8.647.1 Detailed Description . . . 597
8.647.2 Field Documentation . . . 598
8.647.2.1 eriFile . . . 598
8.647.2.2 ParamPresenceMask . . . 598
8.647.2.3 Tlvresult . . . 598
8.648 unpack_dms_SLQSGetPowerSaveModeConfig_t Struct Reference . . . 598
8.648.1 Detailed Description . . . 598
8.648.2 Field Documentation . . . 599
8.648.2.1 pActiveTimer . . . 599
8.648.2.2 ParamPresenceMask . . . 599
8.648.2.3 pDurationDueToOOS . . . 599
8.648.2.4 pDurationThreshold . . . 599
8.648.2.5 pEarlyWakeupTime . . . 599
8.648.2.6 pPeriodicUpdateTimer . . . 599
8.648.2.7 pPsmEnableState . . . 599
8.648.2.8 pRandomizationWindow . . . 599
8.649 unpack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference . . . 599
8.649.1 Detailed Description . . . 599
8.649.2 Field Documentation . . . 600
8.649.2.1 ParamPresenceMask . . . 600
8.649.2.2 Tlvresult . . . 600
8.650 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference . . . 600
8.650.1 Detailed Description . . . 600
8.650.2 Field Documentation . . . 600
8.650.2.1 ParamPresenceMask . . . 600
8.650.2.2 Tlvresult . . . 600
8.651 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference . . . 600
8.651.1 Detailed Description . . . 600
8.651.2 Field Documentation . . . 601

8.651.2.1 crashInfoParam . . . 601
8.651.2.2 ParamPresenceMask . . . 601
8.651.2.3 Tlvresult . . . 601
8.652 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference . . . 601
8.652.1 Detailed Description . . . 601
8.652.2 Field Documentation . . . 602
8.652.2.1 ParamPresenceMask . . . 602
8.652.2.2 pGetDyingGaspCfg . . . 602
8.652.2.3 Tlvresult . . . 602
8.653 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference . . . 602
8.653.1 Detailed Description . . . 602
8.653.2 Field Documentation . . . 602
8.653.2.1 ParamPresenceMask . . . 602
8.653.2.2 pGetDyingGaspStatistics . . . 602
8.653.2.3 Tlvresult . . . 602
8.654 unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference . . . 602
8.654.1 Detailed Description . . . 603
8.654.2 Field Documentation . . . 603
8.654.2.1 carrier . . . 603
8.654.2.2 fwvers . . . 603
8.654.2.3 numEntries . . . 603
8.654.2.4 ParamPresenceMask . . . 603
8.654.2.5 pCurrImgInfo . . . 603
8.654.2.6 pkgver . . . 603
8.654.2.7 priver . . . 603
8.655 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference . . . 603
8.655.1 Detailed Description . . . 604
8.655.2 Field Documentation . . . 605
8.655.2.1 imgType . . . 605
8.655.2.2 logString . . . 605
8.655.2.3 ParamPresenceMask . . . 605
8.655.2.4 refData . . . 605
8.655.2.5 refString . . . 605
8.655.2.6 ResCode . . . 605
8.655.2.7 Tlvresult . . . 605
8.656 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference . . . 605
8.656.1 Detailed Description . . . 605
8.656.2 Field Documentation . . . 606
8.656.2.1 manString . . . 606
8.656.2.2 modelString . . . 606

8.656.2.3 ParamPresenceMask . . . 606
8.656.2.4 plasmaIDString . . . 606
8.656.2.5 swVerString . . . 606
8.656.2.6 Tlvresult . . . 606
8.657 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference . . . 606
8.657.1 Detailed Description . . . 606
8.657.2 Field Documentation . . . 607
8.657.2.1 nameString . . . 607
8.657.2.2 ParamPresenceMask . . . 607
8.657.2.3 Tlvresult . . . 607
8.657.2.4 versionString . . . 607
8.658 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference . . . 607
8.658.1 Detailed Description . . . 607
8.658.2 Field Documentation . . . 607
8.658.2.1 meidString . . . 607
8.658.2.2 ParamPresenceMask . . . 608
8.658.2.3 Tlvresult . . . 608
8.659 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . 608
8.659.1 Detailed Description . . . 608
8.659.2 Field Documentation . . . 608
8.659.2.1 ParamPresenceMask . . . 608
8.659.2.2 Tlvresult . . . 608
8.660 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . 608
8.660.1 Detailed Description . . . 608
8.660.2 Field Documentation . . . 608
8.660.2.1 ParamPresenceMask . . . 609
8.660.2.2 Tlvresult . . . 609
8.661 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . 609
8.661.1 Detailed Description . . . 609
8.661.2 Field Documentation . . . 609
8.661.2.1 ParamPresenceMask . . . 609
8.661.2.2 Tlvresult . . . 609
8.662 unpack_dms_SLQSUIMGetState_t Struct Reference . . . 609
8.662.1 Detailed Description . . . 609
8.662.2 Field Documentation . . . 610
8.662.2.1 ParamPresenceMask . . . 610
8.662.2.2 state . . . 610
8.662.2.3 Tlvresult . . . 610
8.663 unpack_dms_SwiEventReportCallBack_ind_t Struct Reference . . . 610
8.663.1 Detailed Description . . . 610

8.663.2 Field Documentation . . . 610
8.663.2.1 ParamPresenceMask . . . 610
8.663.2.2 TempTlv . . . 610
8.663.2.3 VoltTlv . . . 610
8.664 unpack_dms_SwiSetEventReport_t Struct Reference . . . 610
8.664.1 Detailed Description . . . 611
8.664.2 Field Documentation . . . 611
8.664.2.1 ParamPresenceMask . . . 611
8.664.2.2 Tlvresult . . . 611
8.665 unpack_dms_UIMGetControlKeyStatus_t Struct Reference . . . 611
8.665.1 Detailed Description . . . 611
8.665.2 Field Documentation . . . 612
8.665.2.1 facilityState . . . 612
8.665.2.2 ParamPresenceMask . . . 612
8.665.2.3 Tlvresult . . . 612
8.665.2.4 unblockRetriesLeft . . . 612
8.665.2.5 verifyRetriesLeft . . . 612
8.666 unpack_dms_UIMGetICCID_t Struct Reference . . . 612
8.666.1 Detailed Description . . . 612
8.666.2 Field Documentation . . . 612
8.666.2.1 ParamPresenceMask . . . 613
8.666.2.2 String . . . 613
8.666.2.3 stringSize . . . 613
8.666.2.4 Tlvresult . . . 613
8.667 unpack_dms_UIMGetPINStatus_t Struct Reference . . . 613
8.667.1 Detailed Description . . . 613
8.667.2 Field Documentation . . . 614
8.667.2.1 p1Status . . . 614
8.667.2.2 p1UnblockRetriesLeft . . . 614
8.667.2.3 p1VerifyRetriesLeft . . . 614
8.667.2.4 p2Status . . . 614
8.667.2.5 p2UnblockRetriesLeft . . . 614
8.667.2.6 p2VerifyRetriesLeft . . . 614
8.667.2.7 ParamPresenceMask . . . 614
8.667.2.8 Tlvresult . . . 614
8.668 unpack_dms_UIMSetControlKeyProtection_t Struct Reference . . . 614
8.668.1 Detailed Description . . . 615
8.668.2 Field Documentation . . . 615
8.668.2.1 ParamPresenceMask . . . 615
8.668.2.2 Tlvresult . . . 615

8.668.2.3 verifyRetriesLeft . . . 615
8.669 unpack_dms_UIMSetPINProtection_t Struct Reference . . . 615
8.669.1 Detailed Description . . . 615
8.669.2 Field Documentation . . . 616
8.669.2.1 ParamPresenceMask . . . 616
8.669.2.2 Tlvresult . . . 616
8.669.2.3 unblockRetriesLeft . . . 616
8.669.2.4 verifyRetriesLeft . . . 616
8.670 unpack_dms_UIMUnblockControlKey_t Struct Reference . . . 616
8.670.1 Detailed Description . . . 616
8.670.2 Field Documentation . . . 616
8.670.2.1 ParamPresenceMask . . . 617
8.670.2.2 Tlvresult . . . 617
8.670.2.3 unblockRetriesLeft . . . 617
8.671 unpack_fms_GetImagesPreference_t Struct Reference . . . 617
8.671.1 Detailed Description . . . 617
8.671.2 Field Documentation . . . 617
8.671.2.1 ImageListSize . . . 617
8.671.2.2 ParamPresenceMask . . . 617
8.671.2.3 pImageList . . . 617
8.671.2.4 Tlvresult . . . 617
8.672 unpack_fms_GetStoredImages_t Struct Reference . . . 617
8.672.1 Detailed Description . . . 618
8.672.2 Field Documentation . . . 618
8.672.2.1 imageList . . . 618
8.672.2.2 imagelistSize . . . 618
8.672.2.3 ParamPresenceMask . . . 618
8.672.2.4 Tlvresult . . . 618
8.673 unpack_fms_SetImagesPreference_t Struct Reference . . . 618
8.673.1 Detailed Description . . . 618
8.673.2 Field Documentation . . . 619
8.673.2.1 ImageTypes . . . 619
8.673.2.2 ImageTypesSize . . . 619
8.673.2.3 ParamPresenceMask . . . 619
8.673.2.4 Tlvresult . . . 619
8.674 unpack_ims_SLQSGetIMSSMSConfig_t Struct Reference . . . 619
8.674.1 Detailed Description . . . 619
8.674.2 Field Documentation . . . 620
8.674.2.1 ParamPresenceMask . . . 620
8.674.2.2 pPhoneCtxtURI . . . 620

8.674.2.3 pPhoneCtxtURILen . . . 620
8.674.2.4 pSettingResp . . . 620
8.674.2.5 pSMSFormat . . . 620
8.674.2.6 pSMSOverIPNwInd . . . 620
8.675 unpack_ims_SLQSGetIMSUserConfig_t Struct Reference . . . 620
8.675.1 Detailed Description . . . 620
8.675.2 Field Documentation . . . 620
8.675.2.1 ParamPresenceMask . . . 620
8.675.2.2 pIMSDomain . . . 620
8.675.2.3 pIMSDomainLen . . . 620
8.675.2.4 pSettingResp . . . 620
8.676 unpack_ims_SLQSGetIMSVoIPConfig_t Struct Reference . . . 621
8.676.1 Detailed Description . . . 621
8.676.2 Field Documentation . . . 623
8.676.2.1 pAmrMode . . . 623
8.676.2.2 pAmrOctetAligned . . . 623
8.676.2.3 pAmrWbEnable . . . 623
8.676.2.4 pAmrWBMode . . . 623
8.676.2.5 pAmrWBOctetAligned . . . 623
8.676.2.6 ParamPresenceMask . . . 623
8.676.2.7 pMinSessionExpiryTimer . . . 623
8.676.2.8 pRingBackTimer . . . 623
8.676.2.9 pRingingTimer . . . 623
8.676.2.10 pRTPRTCPInactTimer . . . 623
8.676.2.11 pScrAmrEnable . . . 623
8.676.2.12 pScrAmrWbEnable . . . 623
8.676.2.13 pSessionExpiryTimer . . . 623
8.676.2.14 pSettingResp . . . 623
8.677 unpack_ims_SLQSGetRegMgrConfig_t Struct Reference . . . 623
8.677.1 Detailed Description . . . 624
8.677.2 Field Documentation . . . 624
8.677.2.1 ParamPresenceMask . . . 624
8.677.2.2 pIMSTestMode . . . 624
8.677.2.3 pPCSCFPort . . . 624
8.677.2.4 pPriCSCFPortName . . . 624
8.677.2.5 pPriCSCFPortNameLen . . . 624
8.677.2.6 pSettingResp . . . 624
8.678 unpack_ims_SLQSGetSIPConfig_t Struct Reference . . . 624
8.678.1 Detailed Description . . . 625
8.678.2 Field Documentation . . . 626

8.678.2.1 ParamPresenceMask . . . 626
8.678.2.2 pSettingResp . . . 626
8.678.2.3 pSigCompEnabled . . . 626
8.678.2.4 pSIPLocalPort . . . 626
8.678.2.5 pSubscribeTimer . . . 626
8.678.2.6 pTimerSIPReg . . . 626
8.678.2.7 pTimerT1 . . . 626
8.678.2.8 pTimerT2 . . . 626
8.678.2.9 pTimerTf . . . 626
8.679 unpack_ims_SLQSRegMgrCfgCallBack_ind_t Struct Reference . . . 626
8.679.1 Detailed Description . . . 626
8.679.2 Field Documentation . . . 627
8.679.2.1 ParamPresenceMask . . . 627
8.679.2.2 PCTlv . . . 627
8.679.2.3 PNTlv . . . 627
8.679.2.4 TMTlv . . . 627
8.680 unpack_ims_SLQSSetIMSSMSConfig_t Struct Reference . . . 627
8.680.1 Detailed Description . . . 627
8.680.2 Field Documentation . . . 627
8.680.2.1 ParamPresenceMask . . . 627
8.680.2.2 pSettingResp . . . 627
8.681 unpack_ims_SLQSSetIMSUserConfig_t Struct Reference . . . 627
8.681.1 Detailed Description . . . 627
8.681.2 Field Documentation . . . 628
8.681.2.1 ParamPresenceMask . . . 628
8.681.2.2 pSettingResp . . . 628
8.682 unpack_ims_SLQSSetIMSVoIPConfig_t Struct Reference . . . 628
8.682.1 Detailed Description . . . 628
8.682.2 Field Documentation . . . 628
8.682.2.1 ParamPresenceMask . . . 628
8.682.2.2 pSettingResp . . . 628
8.683 unpack_ims_SLQSSetRegMgrConfig_t Struct Reference . . . 628
8.683.1 Detailed Description . . . 628
8.683.2 Field Documentation . . . 628
8.683.2.1 ParamPresenceMask . . . 629
8.683.2.2 pSettingResp . . . 629
8.684 unpack_ims_SLQSSetSIPConfig_t Struct Reference . . . 629
8.684.1 Detailed Description . . . 629
8.684.2 Field Documentation . . . 629
8.684.2.1 ParamPresenceMask . . . 629

8.684.2.2 pSettingResp . . . 629
8.685 unpack_ims_SLQSSIPCfgCallBack_ind_t Struct Reference . . . 629
8.685.1 Detailed Description . . . 629
8.685.2 Field Documentation . . . 630
8.685.2.1 ParamPresenceMask . . . 630
8.685.2.2 SCTlv . . . 630
8.685.2.3 SPTlv . . . 630
8.685.2.4 SRTlv . . . 630
8.685.2.5 STTlv . . . 630
8.685.2.6 TT1Tlv . . . 630
8.685.2.7 TT2Tlv . . . 630
8.685.2.8 TTfTlv . . . 630
8.686 unpack_ims_SLQSSMSCfgCallBack_ind_t Struct Reference . . . 631
8.686.1 Detailed Description . . . 631
8.686.2 Field Documentation . . . 631
8.686.2.1 ParamPresenceMask . . . 631
8.686.2.2 PCURTlv . . . 631
8.686.2.3 SFTlv . . . 631
8.686.2.4 SINTlv . . . 631
8.687 unpack_ims_SLQSUserCfgCallBack_ind_t Struct Reference . . . 631
8.687.1 Detailed Description . . . 632
8.687.2 Field Documentation . . . 632
8.687.2.1 IDTlv . . . 632
8.687.2.2 ParamPresenceMask . . . 632
8.688 unpack_ims_SLQSVoIPCfgCallBack_ind_t Struct Reference . . . 632
8.688.1 Detailed Description . . . 632
8.688.2 Field Documentation . . . 634
8.688.2.1 AMTlv . . . 634
8.688.2.2 AOATlv . . . 634
8.688.2.3 AWMTlv . . . 634
8.688.2.4 AWOATlv . . . 634
8.688.2.5 EAWTlv . . . 634
8.688.2.6 ESATlv . . . 634
8.688.2.7 ESAWTlv . . . 634
8.688.2.8 MSETlv . . . 634
8.688.2.9 ParamPresenceMask . . . 634
8.688.2.10 RBTTlv . . . 634
8.688.2.11 RTIDTlv . . . 634
8.688.2.12 RTTlv . . . 634
8.688.2.13 SDTlv . . . 634

8.689 unpack_imsa_SLQSGetIMSARegStatus_t Struct Reference . . . 634
8.689.1 Detailed Description . . . 634
8.689.2 Field Documentation . . . 635
8.689.2.1 ImsRegErrCode . . . 635
8.689.2.2 ImsRegStatus . . . 635
8.689.2.3 NewImsRegStatus . . . 635
8.689.2.4 ParamPresenceMask . . . 635
8.690 unpack_imsa_SLQSGetIMSAServiceStatus_t Struct Reference . . . 635
8.690.1 Detailed Description . . . 636
8.690.2 Field Documentation . . . 637
8.690.2.1 ParamPresenceMask . . . 637
8.690.2.2 SmsServiceRat . . . 637
8.690.2.3 SmsServiceStatus . . . 637
8.690.2.4 UtServiceRat . . . 637
8.690.2.5 UtServiceStatus . . . 637
8.690.2.6 VoipServiceRat . . . 637
8.690.2.7 VoipServiceStatus . . . 637
8.690.2.8 VsServiceRat . . . 637
8.690.2.9 VsServiceStatus . . . 637
8.690.2.10 VtServiceRat . . . 638
8.690.2.11 VtServiceStatus . . . 638
8.691 unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t Struct Reference . . . 638
8.691.1 Detailed Description . . . 638
8.691.2 Field Documentation . . . 638
8.691.2.1 FailErrCode . . . 638
8.691.2.2 ParamPresenceMask . . . 638
8.691.2.3 PdpConnState . . . 638
8.692 unpack_imsa_SLQSImsaRatStatusCallBack_ind_t Struct Reference . . . 638
8.692.1 Detailed Description . . . 638
8.692.2 Field Documentation . . . 639
8.692.2.1 ParamPresenceMask . . . 639
8.692.2.2 RatHandover . . . 639
8.693 unpack_imsa_SLQSImsaRegStatusCallBack_ind_t Struct Reference . . . 639
8.693.1 Detailed Description . . . 639
8.693.2 Field Documentation . . . 640
8.693.2.1 IMSRegistration . . . 640
8.693.2.2 IMSRegistrationError . . . 640
8.693.2.3 NewIMSRegistration . . . 640
8.693.2.4 ParamPresenceMask . . . 640
8.694 unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t Struct Reference . . . 640

8.694.1 Detailed Description . . . 640
8.694.2 Field Documentation . . . 641
8.694.2.1 ParamPresenceMask . . . 641
8.694.2.2 SmsRat . . . 641
8.694.2.3 SmsService . . . 641
8.694.2.4 UtRat . . . 641
8.694.2.5 UtService . . . 641
8.694.2.6 VoipRat . . . 641
8.694.2.7 VoipService . . . 641
8.694.2.8 VtRat . . . 641
8.694.2.9 VtService . . . 641
8.695 unpack_loc_BestAvailPos_Ind_t Struct Reference . . . 641
8.695.1 Detailed Description . . . 642
8.695.2 Field Documentation . . . 647
8.695.2.1 pAltitudeWrtEllipsoid . . . 647
8.695.2.2 pAltitudeWrtMeanSeaLevel . . . 647
8.695.2.3 ParamPresenceMask . . . 647
8.695.2.4 pGpsTime . . . 647
8.695.2.5 pHeading . . . 647
8.695.2.6 pHeadingUnc . . . 647
8.695.2.7 pHorCirConf . . . 647
8.695.2.8 pHorEllpConf . . . 647
8.695.2.9 pHorReliability . . . 647
8.695.2.10 pHorUncCircular . . . 647
8.695.2.11 pHorUncEllipseOrientAzimuth . . . 647
8.695.2.12 pHorUncEllipseSemiMajor . . . 647
8.695.2.13 pHorUncEllipseSemiMinor . . . 647
8.695.2.14 pLatitude . . . 647
8.695.2.15 pLongitude . . . 647
8.695.2.16 pMagneticDeviation . . . 647
8.695.2.17 pPrecisionDilution . . . 647
8.695.2.18 pSensorDataUsage . . . 647
8.695.2.19 pSpeedHorizontal . . . 647
8.695.2.20 pSpeedUnc . . . 647
8.695.2.21 pSpeedVertical . . . 647
8.695.2.22 pSpeedVerticalUnc . . . 647
8.695.2.23 pSvUsedforFix . . . 647
8.695.2.24 pTechnologyMask . . . 647
8.695.2.25 pTimeSrc . . . 647
8.695.2.26 pTimestampUtc . . . 647

8.695.2.27 pTimeUnc . . . 647
8.695.2.28 pVertConfidence . . . 648
8.695.2.29 pVertReliability . . . 648
8.695.2.30 pVertUnc . . . 648
8.695.2.31 pXid . . . 648
8.695.2.32 status . . . 648
8.695.2.33 Tlvresult . . . 648
8.696 unpack_loc_CradleMountCallback_Ind_t Struct Reference . . . 648
8.696.1 Detailed Description . . . 648
8.696.2 Field Documentation . . . 648
8.696.2.1 cradleMountConfigStatus . . . 648
8.696.2.2 ParamPresenceMask . . . 648
8.697 unpack_loc_Delete_Assist_Data_t Struct Reference . . . 648
8.697.1 Detailed Description . . . 649
8.697.2 Field Documentation . . . 649
8.697.2.1 ParamPresenceMask . . . 649
8.697.2.2 Tlvresult . . . 649
8.698 unpack_loc_DeleteAssistData_Ind_t Struct Reference . . . 649
8.698.1 Detailed Description . . . 649
8.698.2 Field Documentation . . . 650
8.698.2.1 ParamPresenceMask . . . 650
8.698.2.2 status . . . 650
8.698.2.3 Tlvresult . . . 650
8.699 unpack_loc_EngineState_Ind_t Struct Reference . . . 650
8.699.1 Detailed Description . . . 650
8.699.2 Field Documentation . . . 651
8.699.2.1 engineState . . . 651
8.699.2.2 ParamPresenceMask . . . 651
8.699.2.3 Tlvresult . . . 651
8.700 unpack_loc_EventNMEA_Ind_t Struct Reference . . . 651
8.700.1 Detailed Description . . . 651
8.700.2 Field Documentation . . . 651
8.700.2.1 NMEAData . . . 651
8.700.2.2 ParamPresenceMask . . . 651
8.700.2.3 Tlvresult . . . 651
8.701 unpack_loc_EventRegister_t Struct Reference . . . 652
8.701.1 Detailed Description . . . 652
8.701.2 Field Documentation . . . 652
8.701.2.1 ParamPresenceMask . . . 652
8.701.2.2 Tlvresult . . . 652

8.702 unpack_loc_EventTimeSyncCallback_Ind_t Struct Reference . . . 652
8.702.1 Detailed Description . . . 652
8.702.2 Field Documentation . . . 652
8.702.2.1 ParamPresenceMask . . . 652
8.702.2.2 timeSyncRefCounter . . . 652
8.703 unpack_loc_GetServer_Ind_t Struct Reference . . . 652
8.703.1 Detailed Description . . . 653
8.703.2 Field Documentation . . . 654
8.703.2.1 ParamPresenceMask . . . 654
8.703.2.2 pIPv4AddrInfo . . . 654
8.703.2.3 pIPv6AddrInfo . . . 654
8.703.2.4 pURL . . . 654
8.703.2.5 serverStatus . . . 654
8.703.2.6 serverType . . . 654
8.703.2.7 Tlvresult . . . 654
8.704 unpack_loc_GnssSvInfo_Ind_t Struct Reference . . . 654
8.704.1 Detailed Description . . . 654
8.704.2 Field Documentation . . . 655
8.704.2.1 altitudeAssumed . . . 655
8.704.2.2 ParamPresenceMask . . . 655
8.704.2.3 pSatelliteInfo . . . 655
8.704.2.4 Tlvresult . . . 655
8.705 unpack_loc_InjectPositionCallback_Ind_t Struct Reference . . . 655
8.705.1 Detailed Description . . . 655
8.705.2 Field Documentation . . . 655
8.705.2.1 ParamPresenceMask . . . 655
8.705.2.2 status . . . 655
8.706 unpack_loc_InjectSensorDataCallback_Ind_t Struct Reference . . . 655
8.706.1 Detailed Description . . . 656
8.706.2 Field Documentation . . . 657
8.706.2.1 injectSensorDataStatus . . . 657
8.706.2.2 pAccelSamplesAccepted . . . 657
8.706.2.3 pAccelTempSamplesAccepted . . . 657
8.706.2.4 ParamPresenceMask . . . 657
8.706.2.5 pGyroSamplesAccepted . . . 657
8.706.2.6 pGyroTempSamplesAccepted . . . 657
8.706.2.7 pOpaqueIdentifier . . . 657
8.707 unpack_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference . . . 657
8.707.1 Detailed Description . . . 657
8.707.2 Field Documentation . . . 658

8.707.2.1 injectTimeSyncStatus . . . 658
8.707.2.2 ParamPresenceMask . . . 658
8.708 unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference . . . 658
8.708.1 Detailed Description . . . 658
8.708.2 Field Documentation . . . 658
8.708.2.1 ParamPresenceMask . . . 658
8.708.2.2 status . . . 658
8.709 unpack_loc_PositionRpt_Ind_t Struct Reference . . . 658
8.709.1 Detailed Description . . . 659
8.709.2 Field Documentation . . . 664
8.709.2.1 pAltitudeAssumed . . . 664
8.709.2.2 pAltitudeWrtEllipsoid . . . 664
8.709.2.3 pAltitudeWrtMeanSeaLevel . . . 664
8.709.2.4 ParamPresenceMask . . . 664
8.709.2.5 pFixId . . . 664
8.709.2.6 pGpsTime . . . 664
8.709.2.7 pHeading . . . 664
8.709.2.8 pHeadingUnc . . . 664
8.709.2.9 pHorConfidence . . . 664
8.709.2.10 pHorReliability . . . 664
8.709.2.11 pHorUncCircular . . . 664
8.709.2.12 pHorUncEllipseOrientAzimuth . . . 664
8.709.2.13 pHorUncEllipseSemiMajor . . . 664
8.709.2.14 pHorUncEllipseSemiMinor . . . 664
8.709.2.15 pLatitude . . . 664
8.709.2.16 pLeapSeconds . . . 664
8.709.2.17 pLongitude . . . 664
8.709.2.18 pMagneticDeviation . . . 665
8.709.2.19 pPrecisionDilution . . . 665
8.709.2.20 pSensorDataUsage . . . 665
8.709.2.21 pSpeedHorizontal . . . 665
8.709.2.22 pSpeedUnc . . . 665
8.709.2.23 pSpeedVertical . . . 665
8.709.2.24 pSvUsedforFix . . . 665
8.709.2.25 pTechnologyMask . . . 665
8.709.2.26 pTimeSrc . . . 665
8.709.2.27 pTimestampUtc . . . 665
8.709.2.28 pTimeUnc . . . 665
8.709.2.29 pVertConfidence . . . 665
8.709.2.30 pVertReliability . . . 665

8.709.2.31 pVertUnc . . . 665
8.709.2.32 sessionId . . . 665
8.709.2.33 sessionStatus . . . 665
8.709.2.34 Tlvresult . . . 665
8.710 unpack_loc_SensorStreamingCallback_Ind_t Struct Reference . . . 665
8.710.1 Detailed Description . . . 665
8.710.2 Field Documentation . . . 666
8.710.2.1 pAccelAcceptReady . . . 666
8.710.2.2 pAccelTempAcceptReady . . . 666
8.710.2.3 ParamPresenceMask . . . 666
8.710.2.4 pGyroAcceptReady . . . 666
8.710.2.5 pGyroTempAcceptReady . . . 666
8.711 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . 666
8.711.1 Detailed Description . . . 666
8.711.2 Field Documentation . . . 667
8.711.2.1 ParamPresenceMask . . . 667
8.711.2.2 status . . . 667
8.711.2.3 Tlvresult . . . 667
8.712 unpack_loc_SetExtPowerState_t Struct Reference . . . 667
8.712.1 Detailed Description . . . 667
8.712.2 Field Documentation . . . 667
8.712.2.1 ParamPresenceMask . . . 667
8.712.2.2 Tlvresult . . . 667
8.713 unpack_loc_SetOperationMode_Ind_t Struct Reference . . . 668
8.713.1 Detailed Description . . . 668
8.713.2 Field Documentation . . . 668
8.713.2.1 ParamPresenceMask . . . 668
8.713.2.2 status . . . 668
8.713.2.3 Tlvresult . . . 668
8.714 unpack_loc_SetOperationMode_t Struct Reference . . . 668
8.714.1 Detailed Description . . . 668
8.714.2 Field Documentation . . . 669
8.714.2.1 ParamPresenceMask . . . 669
8.714.2.2 Tlvresult . . . 669
8.715 unpack_loc_SetServer_Ind_t Struct Reference . . . 669
8.715.1 Detailed Description . . . 669
8.715.2 Field Documentation . . . 669
8.715.2.1 serverStatus . . . 669
8.715.2.2 Tlvresult . . . 670
8.716 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . 670

8.716.1 Detailed Description . . . 670
8.716.2 Field Documentation . . . 670
8.716.2.1 ParamPresenceMask . . . 670
8.716.2.2 Tlvresult . . . 670
8.717 unpack_loc_Start_t Struct Reference . . . 670
8.717.1 Detailed Description . . . 670
8.717.2 Field Documentation . . . 670
8.717.2.1 ParamPresenceMask . . . 670
8.717.2.2 Tlvresult . . . 670
8.718 unpack_loc_Stop_t Struct Reference . . . 670
8.718.1 Detailed Description . . . 671
8.718.2 Field Documentation . . . 671
8.718.2.1 ParamPresenceMask . . . 671
8.718.2.2 Tlvresult . . . 671
8.719 unpack_nas_GetACCOLC_t Struct Reference . . . 671
8.719.1 Detailed Description . . . 671
8.719.2 Field Documentation . . . 671
8.719.2.1 pAccolc . . . 671
8.719.2.2 ParamPresenceMask . . . 671
8.720 unpack_nas_GetANAAAAuthenticationStatus_t Struct Reference . . . 671
8.720.1 Detailed Description . . . 672
8.720.2 Field Documentation . . . 672
8.720.2.1 ParamPresenceMask . . . 672
8.720.2.2 pAuthStatus . . . 672
8.721 unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . 672
8.721.1 Detailed Description . . . 672
8.721.2 Field Documentation . . . 674
8.721.2.1 Application . . . 674
8.721.2.2 Broadcast . . . 674
8.721.2.3 CustomSCP . . . 674
8.721.2.4 ForceRev0 . . . 674
8.721.2.5 ParamPresenceMask . . . 674
8.721.2.6 Protocol . . . 674
8.721.2.7 RegForeignNID . . . 674
8.721.2.8 RegForeignSID . . . 674
8.721.2.9 RegHomeSID . . . 674
8.721.2.10 Roaming . . . 674
8.721.2.11 SCI . . . 674
8.721.2.12 SCM . . . 674
8.722 unpack_nas_GetHomeNetwork3GPP2_t Struct Reference . . . 674

8.722.1 Detailed Description . . . 675
8.722.2 Field Documentation . . . 676
8.722.2.1 nameSize . . . 676
8.722.2.2 ParamPresenceMask . . . 676
8.722.2.3 pMCC . . . 676
8.722.2.4 pMNC . . . 676
8.722.2.5 pName . . . 676
8.722.2.6 pNID . . . 676
8.722.2.7 pNw2DescDisp . . . 677
8.722.2.8 pNw2DescEnc . . . 677
8.722.2.9 pNw2DescLen . . . 677
8.722.2.10 pNw2MCC . . . 677
8.722.2.11 pNw2MNC . . . 677
8.722.2.12 pNw2Name . . . 677
8.722.2.13 pSID . . . 677
8.723 unpack_nas_GetHomeNetwork_t Struct Reference . . . 677
8.723.1 Detailed Description . . . 677
8.723.2 Field Documentation . . . 678
8.723.2.1 mcc . . . 678
8.723.2.2 mnc . . . 678
8.723.2.3 name . . . 678
8.723.2.4 nid . . . 678
8.723.2.5 ParamPresenceMask . . . 678
8.723.2.6 sid . . . 678
8.724 unpack_nas_GetNetworkPreference_t Struct Reference . . . 678
8.724.1 Detailed Description . . . 678
8.724.2 Field Documentation . . . 679
8.724.2.1 ActiveTechPref . . . 679
8.724.2.2 Duration . . . 679
8.724.2.3 ParamPresenceMask . . . 679
8.724.2.4 PersistentTechPref . . . 679
8.724.2.5 Tlvresult . . . 679
8.725 unpack_nas_GetRFInfo_t Struct Reference . . . 679
8.725.1 Detailed Description . . . 680
8.725.2 Field Documentation . . . 680
8.725.2.1 instancesSize . . . 680
8.725.2.2 ParamPresenceMask . . . 680
8.725.2.3 RFBandInfoElements . . . 680
8.726 unpack_nas_GetServingNetwork_t Struct Reference . . . 680
8.726.1 Detailed Description . . . 681

8.726.2 Field Documentation . . . 682
8.726.2.1 CSDomain . . . 682
8.726.2.2 DataCaps . . . 682
8.726.2.3 DataCapsLen . . . 682
8.726.2.4 MCC . . . 682
8.726.2.5 MNC . . . 682
8.726.2.6 Name . . . 683
8.726.2.7 nameSize . . . 683
8.726.2.8 ParamPresenceMask . . . 683
8.726.2.9 PSDomain . . . 683
8.726.2.10 RadioIfaces . . . 683
8.726.2.11 RadioIfacesSize . . . 683
8.726.2.12 RAN . . . 683
8.726.2.13 RegistrationState . . . 683
8.726.2.14 Roaming . . . 683
8.727 unpack_nas_GetServingNetworkCapabilities_t Struct Reference . . . 683
8.727.1 Detailed Description . . . 683
8.727.2 Field Documentation . . . 684
8.727.2.1 DataCaps . . . 684
8.727.2.2 DataCapsLen . . . 684
8.727.2.3 ParamPresenceMask . . . 684
8.728 unpack_nas_GetSignalStrengths_t Struct Reference . . . 684
8.728.1 Detailed Description . . . 684
8.728.2 Field Documentation . . . 684
8.728.2.1 len . . . 684
8.728.2.2 ParamPresenceMask . . . 684
8.728.2.3 radio . . . 684
8.728.2.4 rssi . . . 684
8.729 unpack_nas_PerformNetworkScan_t Struct Reference . . . 685
8.729.1 Detailed Description . . . 685
8.729.2 Field Documentation . . . 686
8.729.2.1 p3GppNetworkInfoInstances . . . 686
8.729.2.2 p3GppNetworkInstanceSize . . . 686
8.729.2.3 ParamPresenceMask . . . 686
8.729.2.4 pPCSInstance . . . 686
8.729.2.5 pPCSInstanceSize . . . 686
8.729.2.6 pRATINstance . . . 686
8.729.2.7 pRATInstanceSize . . . 686
8.729.2.8 pScanResult . . . 686
8.730 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference . . . 686

8.730.1 Detailed Description . . . 686
8.730.2 Field Documentation . . . 687
8.730.2.1 dataCaps . . . 687
8.730.2.2 dataCapsSize . . . 687
8.730.2.3 ParamPresenceMask . . . 687
8.731 unpack_nas_SetEventReportInd_t Struct Reference . . . 687
8.731.1 Detailed Description . . . 687
8.731.2 Field Documentation . . . 688
8.731.2.1 ParamPresenceMask . . . 688
8.731.2.2 RFTlv . . . 688
8.731.2.3 RRTlv . . . 688
8.731.2.4 SLQSSSTlv . . . 688
8.731.2.5 SSTlv . . . 688
8.732 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . 688
8.732.1 Detailed Description . . . 688
8.732.2 Field Documentation . . . 689
8.732.2.1 ParamPresenceMask . . . 689
8.732.2.2 sPhyCaAggPcellInfo . . . 689
8.732.2.3 sPhyCaAggScellDlBw . . . 689
8.732.2.4 sPhyCaAggScellIndex . . . 689
8.732.2.5 sPhyCaAggScellIndType . . . 689
8.732.2.6 sPhyCaAggScellInfo . . . 689
8.733 unpack_nas_SetNetworkPreference_t Struct Reference . . . 689
8.733.1 Detailed Description . . . 689
8.733.2 Field Documentation . . . 689
8.733.2.1 ParamPresenceMask . . . 689
8.733.2.2 Tlvresult . . . 689
8.734 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . 689
8.734.1 Detailed Description . . . 690
8.734.2 Field Documentation . . . 690
8.734.2.1 ParamPresenceMask . . . 690
8.734.2.2 roaming . . . 690
8.735 unpack_nas_SetServingSystemCallback_ind_t Struct Reference . . . 690
8.735.1 Detailed Description . . . 690
8.735.2 Field Documentation . . . 690
8.735.2.1 ParamPresenceMask . . . 691
8.735.2.2 SSInfo . . . 691
8.735.2.3 Tlvresult . . . 691
8.736 unpack_nas_SLQSGetErrorRate_t Struct Reference . . . 691
8.736.1 Detailed Description . . . 691

8.736.2 Field Documentation . . . 692
8.736.2.1 ParamPresenceMask . . . 692
8.736.2.2 pCDMAFrameErrRate . . . 692
8.736.2.3 pGSMBER . . . 692
8.736.2.4 pHDRPackErrRate . . . 692
8.736.2.5 pWCDMABER . . . 692
8.737 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . 692
8.737.1 Detailed Description . . . 692
8.737.2 Field Documentation . . . 692
8.737.2.1 LTECphyCaInfo . . . 692
8.737.2.2 ParamPresenceMask . . . 692
8.737.2.3 Tlvresult . . . 692
8.738 unpack_nas_SLQSGetNetworkTime_t Struct Reference . . . 692
8.738.1 Detailed Description . . . 693
8.738.2 Field Documentation . . . 693
8.738.2.1 p3GPP2TimeInfo . . . 693
8.738.2.2 p3GPPTimeInfo . . . 693
8.738.2.3 ParamPresenceMask . . . 693
8.739 unpack_nas_SLQSGetOperatorNameData_t Struct Reference . . . 693
8.739.1 Detailed Description . . . 693
8.739.2 Field Documentation . . . 694
8.739.2.1 ParamPresenceMask . . . 694
8.739.2.2 pNITZInformation . . . 694
8.739.2.3 pOperatorNameString . . . 694
8.739.2.4 pOperatorPLMNList . . . 694
8.739.2.5 pPLMNNetworkName . . . 694
8.739.2.6 pSrvcProviderName . . . 694
8.740 unpack_nas_SLQSGetPLMNName_t Struct Reference . . . 694
8.740.1 Detailed Description . . . 695
8.740.2 Field Documentation . . . 697
8.740.2.1 longName . . . 697
8.740.2.2 longNameCI . . . 697
8.740.2.3 longNameEn . . . 697
8.740.2.4 longNameLen . . . 697
8.740.2.5 longNameSB . . . 697
8.740.2.6 ParamPresenceMask . . . 697
8.740.2.7 shortName . . . 697
8.740.2.8 shortNameCI . . . 697
8.740.2.9 shortNameEn . . . 697
8.740.2.10 shortNameLen . . . 697

8.740.2.11 shortNameSB . . . 697
8.740.2.12 spn . . . 697
8.740.2.13 spnEncoding . . . 698
8.740.2.14 spnLength . . . 698
8.741 unpack_nas_SLQSGetServingSystem_t Struct Reference . . . 698
8.741.1 Detailed Description . . . 698
8.741.2 Field Documentation . . . 701
8.741.2.1 BasestationID . . . 701
8.741.2.2 BasestationLatitude . . . 701
8.741.2.3 BasestationLongitude . . . 701
8.741.2.4 CallBarStatus . . . 701
8.741.2.5 CDMA_P_Rev . . . 701
8.741.2.6 CDMASystemInfoExt . . . 701
8.741.2.7 CellID . . . 702
8.741.2.8 ConcSvcInfo . . . 702
8.741.2.9 CurrentPLMN . . . 702
8.741.2.10 DataSrvCapabilities . . . 702
8.741.2.11 DefaultRoamInd . . . 702
8.741.2.12 DetailedSvcInfo . . . 702
8.741.2.13 DTMInd . . . 702
8.741.2.14 Gpp2TimeZone . . . 702
8.741.2.15 GppNetworkDSTAdjustment . . . 702
8.741.2.16 GppTimeZone . . . 702
8.741.2.17 HdrPersonality . . . 702
8.741.2.18 Lac . . . 702
8.741.2.19 NetworkID . . . 702
8.741.2.20 ParamPresenceMask . . . 702
8.741.2.21 PRLInd . . . 702
8.741.2.22 RoamIndicatorVal . . . 702
8.741.2.23 RoamingIndicatorList . . . 702
8.741.2.24 ServingSystem . . . 702
8.741.2.25 SystemID . . . 702
8.741.2.26 TrackAreaCode . . . 702
8.742 unpack_nas_SLQSGetSignalStrength_t Struct Reference . . . 702
8.742.1 Detailed Description . . . 703
8.742.2 Field Documentation . . . 704
8.742.2.1 ecioList . . . 704
8.742.2.2 ecioListLen . . . 704
8.742.2.3 errorRateList . . . 704
8.742.2.4 errorRateListLen . . . 704

8.742.2.5 lo . . . 705
8.742.2.6 ltersrp . . . 705
8.742.2.7 ltesnr . . . 705
8.742.2.8 ParamPresenceMask . . . 705
8.742.2.9 rsrqInfo . . . 705
8.742.2.10 rxSignalStrengthList . . . 705
8.742.2.11 rxSignalStrengthListLen . . . 705
8.742.2.12 signalStrengthReqMask . . . 705
8.742.2.13 sinr . . . 705
8.743 unpack_nas_SLQSGetSysInfo_t Struct Reference . . . 705
8.743.1 Detailed Description . . . 705
8.743.2 Field Documentation . . . 707
8.743.2.1 pAddCDMASysInfo . . . 707
8.743.2.2 pAddGSMSysInfo . . . 707
8.743.2.3 pAddHDRSysInfo . . . 707
8.743.2.4 pAddLTESysInfo . . . 708
8.743.2.5 pAddWCDMASysInfo . . . 708
8.743.2.6 ParamPresenceMask . . . 708
8.743.2.7 pCDMASrvStatusInfo . . . 708
8.743.2.8 pCDMASysInfo . . . 708
8.743.2.9 pGSMCallBarringSysInfo . . . 708
8.743.2.10 pGSMCipherDomainSysInfo . . . 708
8.743.2.11 pGSMSrvStatusInfo . . . 708
8.743.2.12 pGSMSysInfo . . . 708
8.743.2.13 pHDRSrvStatusInfo . . . 708
8.743.2.14 pHDRSysInfo . . . 708
8.743.2.15 pLTESrvStatusInfo . . . 708
8.743.2.16 pLTESysInfo . . . 708
8.743.2.17 pLTEVoiceSupportSysInfo . . . 708
8.743.2.18 pWCDMACallBarringSysInfo . . . 708
8.743.2.19 pWCDMACipherDomainSysInfo . . . 708
8.743.2.20 pWCDMASrvStatusInfo . . . 708
8.743.2.21 pWCDMASysInfo . . . 708
8.744 unpack_nas_SLQSGetSysSelectionPref_t Struct Reference . . . 708
8.744.1 Detailed Description . . . 708
8.744.2 Field Documentation . . . 713
8.744.2.1 ParamPresenceMask . . . 713
8.744.2.2 pBandPref . . . 713
8.744.2.3 pEmerMode . . . 713
8.744.2.4 pGWAcqOrderPref . . . 713

8.744.2.5 pLTEBandPref . . . 713
8.744.2.6 pModePref . . . 713
8.744.2.7 pNetSelPref . . . 713
8.744.2.8 pPRLPref . . . 713
8.744.2.9 pRoamPref . . . 713
8.744.2.10 pSrvDomainPref . . . 713
8.745 unpack_nas_SLQSGetSysSelectionPrefExt_t Struct Reference . . . 713
8.745.1 Detailed Description . . . 714
8.745.2 Field Documentation . . . 715
8.745.2.1 pAcqOrderPref . . . 715
8.745.2.2 pBandPref . . . 715
8.745.2.3 pCiotAcqOrderPref . . . 715
8.745.2.4 pCiotLteOpModePref . . . 715
8.745.2.5 pEmerMode . . . 715
8.745.2.6 pGWAcqOrderPref . . . 715
8.745.2.7 pLTEBandPref . . . 716
8.745.2.8 pLteM1BandPref . . . 716
8.745.2.9 pLteNb1BandPref . . . 716
8.745.2.10 pModePref . . . 716
8.745.2.11 pNetSelPref . . . 716
8.745.2.12 pNr5gBandPref . . . 716
8.745.2.13 pPRLPref . . . 716
8.745.2.14 pRatDisabledMask . . . 716
8.745.2.15 pRoamPref . . . 716
8.745.2.16 pSrvDomainPref . . . 716
8.746 unpack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . 716
8.746.1 Detailed Description . . . 716
8.746.2 Field Documentation . . . 717
8.746.2.1 ParamPresenceMask . . . 717
8.746.2.2 pCDMAChannel . . . 717
8.746.2.3 pDirNum . . . 717
8.746.2.4 pHomeSIDNID . . . 717
8.746.2.5 pMinBasedIMSI . . . 717
8.746.2.6 pNAMNameInfo . . . 717
8.746.2.7 pTrueIMSI . . . 717
8.747 unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . 717
8.747.1 Detailed Description . . . 717
8.747.2 Field Documentation . . . 718
8.747.2.1 ParamPresenceMask . . . 718
8.747.2.2 pCDMAInfo . . . 718

8.747.2.3 pGERANInfo . . . 718
8.747.2.4 pLTEInfoInterfreq . . . 718
8.747.2.5 pLTEInfoIntrafreq . . . 718
8.747.2.6 pLTEInfoNeighboringGSM . . . 718
8.747.2.7 pLTEInfoNeighboringWCDMA . . . 718
8.747.2.8 pUMTSCellID . . . 719
8.747.2.9 pUMTSInfo . . . 719
8.747.2.10 pWCDMAInfoLTENeighborCell . . . 719
8.748 unpack_nas_SLQSNASGeteDRXParams_t Struct Reference . . . 719
8.748.1 Detailed Description . . . 719
8.748.2 Field Documentation . . . 719
8.748.2.1 ParamPresenceMask . . . 719
8.748.2.2 pCycleLen . . . 719
8.748.2.3 pEdrxEnable . . . 719
8.748.2.4 pPagingTimeWindow . . . 719
8.749 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference . . . 720
8.749.1 Detailed Description . . . 720
8.749.2 Field Documentation . . . 720
8.749.2.1 ParamPresenceMask . . . 720
8.749.2.2 pColorCode . . . 720
8.750 unpack_nas_SLQSNasGetRFInfo_t Struct Reference . . . 720
8.750.1 Detailed Description . . . 720
8.750.2 Field Documentation . . . 721
8.750.2.1 pLTEOperationMode . . . 721
8.750.2.2 pRfBandInfoExtFormat . . . 721
8.750.2.3 pRfBandwidthInfo . . . 721
8.750.2.4 pRfDedicatedBandInfo . . . 721
8.750.2.5 rfbandInfoList . . . 721
8.750.2.6 Tlvresult . . . 721
8.751 unpack_nas_SLQSNasGetSigInfo_t Struct Reference . . . 721
8.751.1 Detailed Description . . . 721
8.751.2 Field Documentation . . . 722
8.751.2.1 CDMASSInfo . . . 722
8.751.2.2 GSMSSInfo . . . 722
8.751.2.3 HDRSSInfo . . . 722
8.751.2.4 LTESSInfo . . . 722
8.751.2.5 ParamPresenceMask . . . 722
8.751.2.6 WCDMASSInfo . . . 722
8.752 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . 722
8.752.1 Detailed Description . . . 722

8.752.2 Field Documentation . . . 723
8.752.2.1 ParamPresenceMask . . . 723
8.752.2.2 pRXChain0Info . . . 723
8.752.2.3 pRXChain1Info . . . 723
8.752.2.4 pTXInfo . . . 723
8.753 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . 723
8.753.1 Detailed Description . . . 723
8.753.2 Field Documentation . . . 724
8.753.2.1 ParamPresenceMask . . . 724
8.753.2.2 pDayltSavAdj . . . 724
8.753.2.3 pRadioInterface . . . 724
8.753.2.4 pTimeZone . . . 724
8.753.2.5 universalTime . . . 724
8.754 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . 724
8.754.1 Detailed Description . . . 724
8.754.2 Field Documentation . . . 725
8.754.2.1 ParamPresenceMask . . . 725
8.754.2.2 pCDMASigInfo . . . 725
8.754.2.3 pGSMSigInfo . . . 725
8.754.2.4 pHDRSigInfo . . . 725
8.754.2.5 pLTESigInfo . . . 725
8.754.2.6 pRscp . . . 725
8.754.2.7 pTDSCDMASigInfoExt . . . 725
8.754.2.8 pWCDMASigInfo . . . 725
8.755 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference . . . 725
8.755.1 Detailed Description . . . 726
8.755.2 Field Documentation . . . 726
8.755.2.1 ParamPresenceMask . . . 726
8.755.2.2 pLteEARFCN . . . 726
8.755.2.3 pLtePCI . . . 726
8.755.2.4 pWcdmaUARFCN . . . 726
8.756 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference . . . 726
8.756.1 Detailed Description . . . 726
8.756.2 Field Documentation . . . 727
8.756.2.1 commonInfo . . . 727
8.756.2.2 ParamPresenceMask . . . 727
8.756.2.3 pLTEInfo . . . 727
8.757 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . 727
8.757.1 Detailed Description . . . 727
8.757.2 Field Documentation . . . 727

8.757.2.1 Info . . . 727
8.757.2.2 ParamPresenceMask . . . 727
8.757.2.3 Tlvresult . . . 727
8.758 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . 728
8.758.1 Detailed Description . . . 728
8.758.2 Field Documentation . . . 728
8.758.2.1 ParamPresenceMask . . . 728
8.758.2.2 t3396_apn . . . 728
8.758.2.3 t3396_plmn_id . . . 728
8.758.2.4 t3396_val . . . 728
8.759 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . 728
8.759.1 Detailed Description . . . 728
8.759.2 Field Documentation . . . 729
8.759.2.1 Info . . . 729
8.759.2.2 ParamPresenceMask . . . 729
8.759.2.3 Tlvresult . . . 729
8.760 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference . . . 729
8.760.1 Detailed Description . . . 729
8.760.2 Field Documentation . . . 730
8.760.2.1 ParamPresenceMask . . . 730
8.760.2.2 pCurrentPersonality . . . 730
8.760.2.3 pPersonalityListLength . . . 730
8.760.2.4 pProtocolSubtypeElement . . . 730
8.761 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference . . . 730
8.761.1 Detailed Description . . . 730
8.761.2 Field Documentation . . . 730
8.761.2.1 pAppSubType . . . 730
8.761.2.2 ParamPresenceMask . . . 731
8.761.2.3 pCurrentPrsnlty . . . 731
8.761.2.4 pPersonalityListLength . . . 731
8.761.2.5 pProtoSubTypElmnt . . . 731
8.762 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference . . . 731
8.762.1 Detailed Description . . . 731
8.762.2 Field Documentation . . . 731
8.762.2.1 ParamPresenceMask . . . 731
8.762.2.2 pDRCParams . . . 731
8.762.2.3 pPilotSetData . . . 731
8.762.2.4 pUATI . . . 731
8.763 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . 731
8.763.1 Detailed Description . . . 732

8.763.2 Field Documentation . . . 732
8.763.2.1 CQIValueCW0 . . . 732
8.763.2.2 CQIValueCW1 . . . 732
8.763.2.3 ParamPresenceMask . . . 732
8.763.2.4 ValidityCW0 . . . 732
8.763.2.5 ValidityCW1 . . . 732
8.764 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference . . . 733
8.764.1 Detailed Description . . . 733
8.764.2 Field Documentation . . . 733
8.764.2.1 ParamPresenceMask . . . 733
8.764.2.2 pSccRxInfo . . . 733
8.765 unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference . . . 733
8.765.1 Detailed Description . . . 733
8.765.2 Field Documentation . . . 733
8.765.2.1 ParamPresenceMask . . . 733
8.765.2.2 pHDRPersInd . . . 733
8.766 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference . . . 734
8.766.1 Detailed Description . . . 734
8.766.2 Field Documentation . . . 734
8.766.2.1 ParamPresenceMask . . . 734
8.766.2.2 pDataStatusDetail . . . 734
8.766.2.3 pDeviceConfigDetail . . . 734
8.766.2.4 pNetworkStat1x . . . 734
8.766.2.5 pNetworkStatEVDO . . . 735
8.766.2.6 pObjectVer . . . 735
8.767 unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t Struct Reference . . . 735
8.767.1 Detailed Description . . . 735
8.767.2 Field Documentation . . . 735
8.767.2.1 ParamPresenceMask . . . 735
8.767.2.2 rankIndicatorTlv . . . 735
8.768 unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . 735
8.768.1 Detailed Description . . . 736
8.768.2 Field Documentation . . . 738
8.768.2.1 pAddCDMASysInfo . . . 738
8.768.2.2 pAddGSMSysInfo . . . 738
8.768.2.3 pAddHDRSysInfo . . . 738
8.768.2.4 pAddLTESysInfo . . . 738
8.768.2.5 pAddWCDMASysInfo . . . 738
8.768.2.6 ParamPresenceMask . . . 738
8.768.2.7 pCDMASrvStatusInfo . . . 738

8.768.2.8 pCDMASysInfo . . . 738
8.768.2.9 pGSMCallBarringSysInfo . . . 738
8.768.2.10 pGSMCipherDomainSysInfo . . . 738
8.768.2.11 pGSMSrvStatusInfo . . . 738
8.768.2.12 pGSMSysInfo . . . 738
8.768.2.13 pHDRSrvStatusInfo . . . 738
8.768.2.14 pHDRSysInfo . . . 738
8.768.2.15 pLTESrvStatusInfo . . . 738
8.768.2.16 pLTESysInfo . . . 738
8.768.2.17 pLTEVoiceSupportSysInfo . . . 738
8.768.2.18 pSysInfoNoChange . . . 738
8.768.2.19 pWCDMACallBarringSysInfo . . . 738
8.768.2.20 pWCDMACipherDomainSysInfo . . . 738
8.768.2.21 pWCDMASrvStatusInfo . . . 738
8.768.2.22 pWCDMASysInfo . . . 739
8.769 unpack_omaDmConfigTlv_t Struct Reference . . . 739
8.769.1 Detailed Description . . . 739
8.769.2 Field Documentation . . . 739
8.769.2.1 alertmsg . . . 739
8.769.2.2 alertmsglength . . . 739
8.769.2.3 state . . . 739
8.769.2.4 userInputReq . . . 739
8.769.2.5 userInputTimeout . . . 739
8.770 unpack_omaDmFotaTlv_t Struct Reference . . . 739
8.770.1 Detailed Description . . . 740
8.770.2 Field Documentation . . . 741
8.770.2.1 description . . . 741
8.770.2.2 descriptionlength . . . 741
8.770.2.3 fwdloadsize . . . 741
8.770.2.4 fwloadComplete . . . 741
8.770.2.5 namelength . . . 741
8.770.2.6 package_name . . . 741
8.770.2.7 sessionType . . . 741
8.770.2.8 severity . . . 741
8.770.2.9 state . . . 741
8.770.2.10 updateCompleteStatus . . . 741
8.770.2.11 userInputReq . . . 741
8.770.2.12 userInputTimeout . . . 741
8.770.2.13 version . . . 741
8.770.2.14 versionlength . . . 741

8.771 unpack_omaDmNotificationsTlv_t Struct Reference . . . 742
8.771.1 Detailed Description . . . 742
8.771.2 Field Documentation . . . 742
8.771.2.1 notification . . . 742
8.771.2.2 sessionStatus . . . 742
8.772 unpack_pds_GetPDSDefaults_t Struct Reference . . . 742
8.772.1 Detailed Description . . . 743
8.772.2 Field Documentation . . . 743
8.772.2.1 pAccuracy . . . 743
8.772.2.2 ParamPresenceMask . . . 743
8.772.2.3 pInterval . . . 743
8.772.2.4 pOperation . . . 743
8.772.2.5 pTimeout . . . 743
8.773 unpack_pds_GetPDSState_t Struct Reference . . . 743
8.773.1 Detailed Description . . . 744
8.773.2 Field Documentation . . . 744
8.773.2.1 ParamPresenceMask . . . 744
8.773.2.2 pEnabledStatus . . . 744
8.773.2.3 pTrackingStatus . . . 744
8.774 unpack_pds_GetPortAutomaticTracking_t Struct Reference . . . 744
8.774.1 Detailed Description . . . 744
8.774.2 Field Documentation . . . 745
8.774.2.1 ParamPresenceMask . . . 745
8.774.2.2 pbAuto . . . 745
8.775 unpack_pds_GetServiceAutomaticTracking_t Struct Reference . . . 745
8.775.1 Detailed Description . . . 745
8.775.2 Field Documentation . . . 745
8.775.2.1 ParamPresenceMask . . . 745
8.775.2.2 pbAuto . . . 745
8.776 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference . . . 745
8.776.1 Detailed Description . . . 745
8.776.2 Field Documentation . . . 746
8.776.2.1 ParamPresenceMask . . . 746
8.776.2.2 pbEnabled . . . 746
8.776.2.3 pInterval . . . 746
8.777 unpack_pds_GetXTRANetwork_t Struct Reference . . . 746
8.777.1 Detailed Description . . . 746
8.777.2 Field Documentation . . . 746
8.777.2.1 ParamPresenceMask . . . 746
8.777.2.2 pPreference . . . 747

8.778 unpack_pds_GetXTRAValidity_t Struct Reference . . . 747
8.778.1 Detailed Description . . . 747
8.778.2 Field Documentation . . . 747
8.778.2.1 ParamPresenceMask . . . 747
8.778.2.2 pDuration . . . 747
8.778.2.3 pGPSWeek . . . 747
8.778.2.4 pGPSWeekOffset . . . 747
8.779 unpack_pds_SetEventReport_Ind_t Struct Reference . . . 747
8.779.1 Detailed Description . . . 748
8.779.2 Field Documentation . . . 749
8.779.2.1 dLatitude . . . 749
8.779.2.2 dLongitude . . . 749
8.779.2.3 has_dLatitude . . . 749
8.779.2.4 has_dLongitude . . . 749
8.779.2.5 has_PositionDataNMEA . . . 749
8.779.2.6 has_posSrc . . . 749
8.779.2.7 has_SessionStatus . . . 749
8.779.2.8 ParamPresenceMask . . . 749
8.779.2.9 PositionDataNMEA . . . 749
8.779.2.10 posSrc . . . 749
8.779.2.11 SessionStatus . . . 749
8.780 unpack_pds_SetPdsState_Ind_t Struct Reference . . . 749
8.780.1 Detailed Description . . . 749
8.780.2 Field Documentation . . . 750
8.780.2.1 EnabledStatus . . . 750
8.780.2.2 ParamPresenceMask . . . 750
8.780.2.3 TlvPresent . . . 750
8.780.2.4 TrackingStatus . . . 750
8.781 unpack_pds_SLQSGetAGPSConfig_t Struct Reference . . . 750
8.781.1 Detailed Description . . . 750
8.781.2 Field Documentation . . . 751
8.781.2.1 ParamPresenceMask . . . 751
8.781.2.2 pServerAddress . . . 751
8.781.2.3 pServerPort . . . 751
8.781.2.4 pServerURL . . . 751
8.781.2.5 pServerURLLength . . . 751
8.782 unpack_pds_SLQSGetGPSStateInfo_t Struct Reference . . . 751
8.782.1 Detailed Description . . . 752
8.782.2 Field Documentation . . . 755
8.782.2.1 Altitude . . . 755

8.782.2.2 EngineState . . . 755
8.782.2.3 glo_almanac_sv_msk . . . 755
8.782.2.4 glo_ephemeris_sv_msk . . . 755
8.782.2.5 glo_health_sv_msk . . . 755
8.782.2.6 glo_visible_sv_msk . . . 755
8.782.2.7 gps_almanac_sv_msk . . . 755
8.782.2.8 gps_ephemeris_sv_msk . . . 755
8.782.2.9 gps_health_sv_msk . . . 755
8.782.2.10 gps_visible_sv_msk . . . 755
8.782.2.11 HorizontalUncertainty . . . 755
8.782.2.12 iono_valid . . . 755
8.782.2.13 Latitude . . . 755
8.782.2.14 Longitude . . . 755
8.782.2.15 ParamPresenceMask . . . 755
8.782.2.16 sbas_almanac_sv_msk . . . 755
8.782.2.17 sbas_ephemeris_sv_msk . . . 755
8.782.2.18 sbas_health_sv_msk . . . 755
8.782.2.19 sbas_visible_sv_msk . . . 755
8.782.2.20 Time_uncert_ms . . . 755
8.782.2.21 TimeStmp_gps_week . . . 755
8.782.2.22 TimeStmp_tow_ms . . . 755
8.782.2.23 ValidMask . . . 755
8.782.2.24 VerticalUncertainty . . . 755
8.782.2.25 xtra_start_gps_minutes . . . 756
8.782.2.26 xtra_start_gps_week . . . 756
8.782.2.27 xtra_valid_duration_hours . . . 756
8.783 unpack_qmi_t Struct Reference . . . 756
8.783.1 Detailed Description . . . 756
8.783.2 Field Documentation . . . 756
8.783.2.1 msgid . . . 756
8.783.2.2 type . . . 756
8.783.2.3 xid . . . 756
8.784 unpack_qos_BindDataPort_t Struct Reference . . . 756
8.784.1 Detailed Description . . . 756
8.784.2 Field Documentation . . . 756
8.784.2.1 ParamPresenceMask . . . 756
8.785 unpack_qos_dataRate_t Struct Reference . . . 757
8.785.1 Detailed Description . . . 757
8.785.2 Field Documentation . . . 757
8.785.2.1 dataRateMax . . . 757

8.785.2.2 guaranteedRate . . . 757
8.786 unpack_qos_IPv4Addr_t Struct Reference . . . 757
8.786.1 Detailed Description . . . 757
8.786.2 Field Documentation . . . 757
8.786.2.1 addr . . . 757
8.786.2.2 subnetMask . . . 757
8.787 unpack_qos_IPv6Addr_t Struct Reference . . . 757
8.787.1 Detailed Description . . . 758
8.787.2 Field Documentation . . . 758
8.787.2.1 addr . . . 758
8.787.2.2 prefixLen . . . 758
8.788 unpack_qos_IPv6TrafCls_t Struct Reference . . . 758
8.788.1 Detailed Description . . . 758
8.788.2 Field Documentation . . . 758
8.788.2.1 mask . . . 758
8.788.2.2 val . . . 758
8.789 unpack_qos_pktErrRate_t Struct Reference . . . 758
8.789.1 Detailed Description . . . 759
8.789.2 Field Documentation . . . 759
8.789.2.1 exponent . . . 759
8.789.2.2 multiplier . . . 759
8.790 unpack_qos_Port_t Struct Reference . . . 759
8.790.1 Detailed Description . . . 759
8.790.2 Field Documentation . . . 759
8.790.2.1 port . . . 759
8.790.2.2 range . . . 759
8.791 unpack_qos_QosFlowInfo_t Struct Reference . . . 759
8.791.1 Detailed Description . . . 760
8.791.2 Field Documentation . . . 760
8.791.2.1 BearerID . . . 760
8.791.2.2 is_RxQFlowGranted_Available . . . 760
8.791.2.3 is_TxQFlowGranted_Available . . . 760
8.791.2.4 NumRxFilters . . . 761
8.791.2.5 NumTxFilters . . . 761
8.791.2.6 QFlowState . . . 761
8.791.2.7 RxQFilter . . . 761
8.791.2.8 RxQFlowGranted . . . 761
8.791.2.9 TxQFilter . . . 761
8.791.2.10 TxQFlowGranted . . . 761
8.792 unpack_qos_QosFlowInfoState_t Struct Reference . . . 761

8.792.1 Detailed Description . . . 761
8.792.2 Field Documentation . . . 761
8.792.2.1 id . . . 761
8.792.2.2 isNewFlow . . . 761
8.792.2.3 state . . . 761
8.793 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . 761
8.793.1 Detailed Description . . . 762
8.793.2 Field Documentation . . . 762
8.793.2.1 NWQoSStatus . . . 762
8.793.2.2 ParamPresenceMask . . . 762
8.794 unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . 762
8.794.1 Detailed Description . . . 762
8.794.2 Field Documentation . . . 763
8.794.2.1 ambr_dl . . . 763
8.794.2.2 ambr_dl_ext . . . 763
8.794.2.3 ambr_dl_ext2 . . . 763
8.794.2.4 ambr_ul . . . 763
8.794.2.5 ambr_ul_ext . . . 763
8.794.2.6 ambr_ul_ext2 . . . 763
8.794.2.7 apnId . . . 763
8.794.2.8 ParamPresenceMask . . . 763
8.795 unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . 764
8.795.1 Detailed Description . . . 764
8.795.2 Field Documentation . . . 765
8.795.2.1 apnId . . . 765
8.795.2.2 numQosFlow . . . 765
8.795.2.3 ParamPresenceMask . . . 765
8.795.2.4 qosFlow . . . 765
8.795.2.5 total_rx_bytes . . . 765
8.795.2.6 total_rx_pkt . . . 765
8.795.2.7 total_tx_bytes . . . 765
8.795.2.8 total_tx_bytes_drp . . . 765
8.795.2.9 total_tx_pkt . . . 765
8.795.2.10 total_tx_pkt_drp . . . 765
8.796 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . 765
8.796.1 Detailed Description . . . 765
8.796.2 Field Documentation . . . 766
8.796.2.1 NumFlows . . . 766
8.796.2.2 ParamPresenceMask . . . 766
8.796.2.3 QosFlowInfo . . . 766

8.797 unpack_qos_SLQSSetQosEventCallback_t Struct Reference . . . 766
8.797.1 Detailed Description . . . 766
8.797.2 Field Documentation . . . 766
8.797.2.1 ParamPresenceMask . . . 766
8.798 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . 766
8.798.1 Detailed Description . . . 766
8.798.2 Field Documentation . . . 767
8.798.2.1 ParamPresenceMask . . . 767
8.798.2.2 status . . . 767
8.799 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . 767
8.799.1 Detailed Description . . . 767
8.799.2 Field Documentation . . . 767
8.799.2.1 event . . . 767
8.799.2.2 ParamPresenceMask . . . 767
8.800 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . 767
8.800.1 Detailed Description . . . 768
8.800.2 Field Documentation . . . 769
8.800.2.1 event . . . 769
8.800.2.2 id . . . 769
8.800.2.3 ParamPresenceMask . . . 769
8.800.2.4 reason . . . 769
8.800.2.5 status . . . 769
8.801 unpack_qos_swiQosFilter_t Struct Reference . . . 769
8.801.1 Detailed Description . . . 770
8.801.2 Field Documentation . . . 771
8.801.2.1 EspSpi . . . 771
8.801.2.2 Id . . . 771
8.801.2.3 index . . . 771
8.801.2.4 IPv4DstAddr . . . 771
8.801.2.5 IPv4SrcAddr . . . 771
8.801.2.6 IPv4Tos . . . 771
8.801.2.7 IPv6DstAddr . . . 771
8.801.2.8 IPv6Label . . . 771
8.801.2.9 IPv6SrcAddr . . . 771
8.801.2.10 IPv6TrafCls . . . 771
8.801.2.11 is_EspSpi_Available . . . 771
8.801.2.12 is_Id_Available . . . 771
8.801.2.13 is_IPv4DstAddr_Available . . . 771
8.801.2.14 is_IPv4SrcAddr_Available . . . 771
8.801.2.15 is_IPv4Tos_Available . . . 771

8.801.2.16 is_IPv6DstAddr_Available . . . 771
8.801.2.17 is_IPv6Label_Available . . . 771
8.801.2.18 is_IPv6SrcAddr_Available . . . 771
8.801.2.19 is_IPv6TrafCls_Available . . . 772
8.801.2.20 is_NxtHdrProto_Available . . . 772
8.801.2.21 is_Precedence_Available . . . 772
8.801.2.22 is_TCPDstPort_Available . . . 772
8.801.2.23 is_TCPSrcPort_Available . . . 772
8.801.2.24 is_TranDstPort_Available . . . 772
8.801.2.25 is_TranSrcPort_Available . . . 772
8.801.2.26 is_UDPDstPort_Available . . . 772
8.801.2.27 is_UDPSrcPort_Available . . . 772
8.801.2.28 NxtHdrProto . . . 772
8.801.2.29 Precedence . . . 772
8.801.2.30 TCPDstPort . . . 772
8.801.2.31 TCPSrcPort . . . 772
8.801.2.32 TranDstPort . . . 772
8.801.2.33 TranSrcPort . . . 772
8.801.2.34 UDPDstPort . . . 772
8.801.2.35 UDPSrcPort . . . 772
8.801.2.36 version . . . 772
8.802 unpack_qos_swiQosFlow_t Struct Reference . . . 772
8.802.1 Detailed Description . . . 773
8.802.2 Field Documentation . . . 775
8.802.2.1 DataRate . . . 775
8.802.2.2 index . . . 775
8.802.2.3 is_DataRate_Available . . . 775
8.802.2.4 is_Jitter_Available . . . 775
8.802.2.5 is_Latency_Available . . . 775
8.802.2.6 is_LteQci_Available . . . 775
8.802.2.7 is_MaxAllowedPktSz_Available . . . 775
8.802.2.8 is_MinPolicedPktSz_Available . . . 775
8.802.2.9 is_PktErrRate_Available . . . 775
8.802.2.10 is_ProfileId3GPP2_Available . . . 775
8.802.2.11 is_TokenBucket_Available . . . 775
8.802.2.12 is_TrafficClass_Available . . . 775
8.802.2.13 is_val_3GPP2Pri_Available . . . 775
8.802.2.14 is_val_3GPPImCn_Available . . . 775
8.802.2.15 is_val_3GPPResResidualBER_Available . . . 776
8.802.2.16 is_val_3GPPSigInd_Available . . . 776

8.802.2.17 is_val_3GPPTraHdlPri_Available . . . 776
8.802.2.18 Jitter . . . 776
8.802.2.19 Latency . . . 776
8.802.2.20 LteQci . . . 776
8.802.2.21 MaxAllowedPktSz . . . 776
8.802.2.22 MinPolicedPktSz . . . 776
8.802.2.23 PktErrRate . . . 776
8.802.2.24 ProfileId3GPP2 . . . 776
8.802.2.25 TokenBucket . . . 776
8.802.2.26 TrafficClass . . . 776
8.802.2.27 val_3GPP2Pri . . . 776
8.802.2.28 val_3GPPImCn . . . 776
8.802.2.29 val_3GPPResResidualBER . . . 776
8.802.2.30 val_3GPPSigInd . . . 776
8.802.2.31 val_3GPPTraHdlPri . . . 776
8.803 unpack_qos_tokenBucket_t Struct Reference . . . 776
8.803.1 Detailed Description . . . 776
8.803.2 Field Documentation . . . 777
8.803.2.1 bucketSz . . . 777
8.803.2.2 peakRate . . . 777
8.803.2.3 tokenRate . . . 777
8.804 unpack_qos_Tos_t Struct Reference . . . 777
8.804.1 Detailed Description . . . 777
8.804.2 Field Documentation . . . 777
8.804.2.1 mask . . . 777
8.804.2.2 val . . . 777
8.805 unpack_QosFlowStat_t Struct Reference . . . 777
8.805.1 Detailed Description . . . 777
8.805.2 Field Documentation . . . 778
8.805.2.1 bearerId . . . 778
8.805.2.2 tx_bytes . . . 778
8.805.2.3 tx_bytes_drp . . . 778
8.805.2.4 tx_pkt . . . 778
8.805.2.5 tx_pkt_drp . . . 778
8.806 unpack_result_t Struct Reference . . . 778
8.806.1 Detailed Description . . . 778
8.806.2 Field Documentation . . . 778
8.806.2.1 ParamPresenceMask . . . 778
8.806.2.2 Tlvresult . . . 778
8.807 unpack_rms_GetSMSWake_t Struct Reference . . . 779

8.807.1 Detailed Description . . . 779
8.807.2 Field Documentation . . . 779
8.807.2.1 enabled . . . 779
8.807.2.2 ParamPresenceMask . . . 779
8.807.2.3 wake_mask . . . 779
8.808 unpack_rms_SetSMSWake_t Struct Reference . . . 779
8.808.1 Detailed Description . . . 779
8.808.2 Field Documentation . . . 780
8.808.2.1 ParamPresenceMask . . . 780
8.809 unpack_RMTransferStatistics_ind_t Struct Reference . . . 780
8.809.1 Detailed Description . . . 780
8.809.2 Field Documentation . . . 780
8.809.2.1 RxDropConutTlv . . . 780
8.809.2.2 RxOkByteCountTlv . . . 780
8.809.2.3 RxOkConutTlv . . . 780
8.809.2.4 TxDropConutTlv . . . 780
8.809.2.5 TxOkByteCountTlv . . . 780
8.809.2.6 TxOkConutTlv . . . 780
8.810 unpack_sar_SLQSGetRfSarState_t Struct Reference . . . 781
8.810.1 Detailed Description . . . 781
8.810.2 Field Documentation . . . 781
8.810.2.1 ParamPresenceMask . . . 781
8.810.2.2 pSarRFState . . . 781
8.811 unpack_sms_GetSMSCAddress_t Struct Reference . . . 781
8.811.1 Detailed Description . . . 781
8.811.2 Field Documentation . . . 782
8.811.2.1 addressSize . . . 782
8.811.2.2 ParamPresenceMask . . . 782
8.811.2.3 pSMSCAddress . . . 782
8.811.2.4 pSMSCType . . . 782
8.811.2.5 typeSize . . . 782
8.812 unpack_sms_SaveSMS_t Struct Reference . . . 782
8.812.1 Detailed Description . . . 782
8.812.2 Field Documentation . . . 783
8.812.2.1 ParamPresenceMask . . . 783
8.812.2.2 pMessageIndex . . . 783
8.813 unpack_sms_SendSMS_t Struct Reference . . . 783
8.813.1 Detailed Description . . . 783
8.813.2 Field Documentation . . . 783
8.813.2.1 messageFailureCode . . . 783

8.813.2.2 messageID . . . 783
8.813.2.3 ParamPresenceMask . . . 783
8.814 unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . 783
8.814.1 Detailed Description . . . 784
8.814.2 Field Documentation . . . 784
8.814.2.1 ETWSPLMNTlv . . . 785
8.814.2.2 ETWSTlv . . . 785
8.814.2.3 IMSTlv . . . 785
8.814.2.4 MMTlv . . . 785
8.814.2.5 NewMMTlv . . . 785
8.814.2.6 ParamPresenceMask . . . 785
8.814.2.7 SMSCTlv . . . 785
8.814.2.8 TRMessageTlv . . . 785
8.815 unpack_sms_SetNewSMSCallback_t Struct Reference . . . 785
8.815.1 Detailed Description . . . 785
8.815.2 Field Documentation . . . 785
8.815.2.1 ParamPresenceMask . . . 785
8.816 unpack_sms_SetSMSCAddress_t Struct Reference . . . 785
8.816.1 Detailed Description . . . 785
8.816.2 Field Documentation . . . 786
8.816.2.1 ParamPresenceMask . . . 786
8.817 unpack_sms_SLQSDeleteSMS_t Struct Reference . . . 786
8.817.1 Detailed Description . . . 786
8.817.2 Field Documentation . . . 786
8.817.2.1 ParamPresenceMask . . . 786
8.818 unpack_sms_SLQSGetIndicationRegister_t Struct Reference . . . 786
8.818.1 Detailed Description . . . 786
8.818.2 Field Documentation . . . 787
8.818.2.1 ParamPresenceMask . . . 787
8.818.2.2 pGetIndicationRegInfo . . . 787
8.819 unpack_sms_SLQSGetMessageWaiting_t Struct Reference . . . 787
8.819.1 Detailed Description . . . 787
8.819.2 Field Documentation . . . 787
8.819.2.1 ParamPresenceMask . . . 787
8.819.2.2 pGetMsgWaitingInfoResp . . . 787
8.820 unpack_sms_SLQSGetSMS_t Struct Reference . . . 787
8.820.1 Detailed Description . . . 787
8.820.2 Field Documentation . . . 788
8.820.2.1 message . . . 788
8.820.2.2 messageFormat . . . 788

8.820.2.3 messageSize . . . 788
8.820.2.4 messageTag . . . 788
8.820.2.5 ParamPresenceMask . . . 788
8.821 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . 788
8.821.1 Detailed Description . . . 789
8.821.2 Field Documentation . . . 789
8.821.2.1 ParamPresenceMask . . . 789
8.821.2.2 pBroadcastConfig . . . 789
8.821.2.3 pCDMABroadcastConfig . . . 789
8.822 unpack_sms_SLQSGetSMSList_t Struct Reference . . . 789
8.822.1 Detailed Description . . . 789
8.822.2 Field Documentation . . . 790
8.822.2.1 messageList . . . 790
8.822.2.2 messageListSize . . . 790
8.822.2.3 ParamPresenceMask . . . 790
8.823 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference . . . 790
8.823.1 Detailed Description . . . 790
8.823.2 Field Documentation . . . 790
8.823.2.1 ParamPresenceMask . . . 790
8.823.2.2 pGetTransLayerInfo . . . 790
8.824 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference . . . 790
8.824.1 Detailed Description . . . 791
8.824.2 Field Documentation . . . 791
8.824.2.1 ParamPresenceMask . . . 791
8.824.2.2 pGetTransNWRegInfo . . . 791
8.825 unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . 791
8.825.1 Detailed Description . . . 791
8.825.2 Field Documentation . . . 791
8.825.2.1 ParamPresenceMask . . . 791
8.826 unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference . . . 791
8.826.1 Detailed Description . . . 792
8.826.2 Field Documentation . . . 792
8.826.2.1 NWRegStat . . . 792
8.826.2.2 ParamPresenceMask . . . 792
8.827 unpack_sms_SLQSSendAsyncSMS_t Struct Reference . . . 792
8.827.1 Detailed Description . . . 792
8.827.2 Field Documentation . . . 792
8.827.2.1 ParamPresenceMask . . . 792
8.828 unpack_sms_SLQSSetIndicationRegister_t Struct Reference . . . 792
8.828.1 Detailed Description . . . 793

8.828.2 Field Documentation . . . 793
8.828.2.1 ParamPresenceMask . . . 793
8.829 unpack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . 793
8.829.1 Detailed Description . . . 793
8.829.2 Field Documentation . . . 793
8.829.2.1 ParamPresenceMask . . . 793
8.830 unpack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . 793
8.830.1 Detailed Description . . . 793
8.830.2 Field Documentation . . . 794
8.830.2.1 ParamPresenceMask . . . 794
8.831 unpack_sms_SLQSSetSmsStorage_t Struct Reference . . . 794
8.831.1 Detailed Description . . . 794
8.831.2 Field Documentation . . . 794
8.831.2.1 ParamPresenceMask . . . 794
8.832 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . 794
8.832.1 Detailed Description . . . 794
8.832.2 Field Documentation . . . 794
8.832.2.1 ParamPresenceMask . . . 795
8.832.2.2 pMaxStorageSizeResp . . . 795
8.833 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference . . . 795
8.833.1 Detailed Description . . . 795
8.833.2 Field Documentation . . . 795
8.833.2.1 ParamPresenceMask . . . 795
8.833.2.2 pMessageProtocol . . . 795
8.834 unpack_sms_SLQSSmsSetRoutes_t Struct Reference . . . 795
8.834.1 Detailed Description . . . 795
8.834.2 Field Documentation . . . 795
8.834.2.1 ParamPresenceMask . . . 795
8.835 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference . . . 796
8.835.1 Detailed Description . . . 796
8.835.2 Field Documentation . . . 796
8.835.2.1 ParamPresenceMask . . . 796
8.835.2.2 pSmsStorage . . . 796
8.836 unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference . . . 796
8.836.1 Detailed Description . . . 796
8.836.2 Field Documentation . . . 797
8.836.2.1 ParamPresenceMask . . . 797
8.836.2.2 pTransLayerInfo . . . 797
8.836.2.3 regInd . . . 797
8.837 unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t Struct Reference . . . 797

8.837.1 Detailed Description . . . 797
8.837.2 Field Documentation . . . 799
8.837.2.1 alphaIDLen . . . 799
8.837.2.2 causeCode . . . 799
8.837.2.3 errorClass . . . 799
8.837.2.4 messageID . . . 799
8.837.2.5 msgDelFailureCause . . . 799
8.837.2.6 msgDelFailureType . . . 799
8.837.2.7 pAlphaID . . . 799
8.837.2.8 ParamPresenceMask . . . 799
8.837.2.9 RPCause . . . 799
8.837.2.10 sendStatus . . . 799
8.837.2.11 TPCause . . . 799
8.837.2.12 userData . . . 799
8.838 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . 799
8.838.1 Detailed Description . . . 799
8.838.2 Field Documentation . . . 800
8.838.2.1 messageMode . . . 800
8.838.2.2 ParamPresenceMask . . . 800
8.838.2.3 storageType . . . 800
8.839 unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t Struct Reference . . . 800
8.839.1 Detailed Description . . . 800
8.839.2 Field Documentation . . . 800
8.839.2.1 msgWaitInfo . . . 800
8.839.2.2 numInstances . . . 800
8.839.2.3 ParamPresenceMask . . . 800
8.840 unpack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference . . . 800
8.840.1 Detailed Description . . . 801
8.840.2 Field Documentation . . . 801
8.840.2.1 CwtMute . . . 801
8.840.2.2 EarMute . . . 801
8.840.2.3 Generator . . . 801
8.840.2.4 MicMute . . . 802
8.840.2.5 ParamPresenceMask . . . 802
8.840.2.6 Profile . . . 802
8.840.2.7 Volume . . . 802
8.841 unpack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference . . . 802
8.841.1 Detailed Description . . . 802
8.841.2 Field Documentation . . . 802
8.841.2.1 Level . . . 802

8.841.2.2 ParamPresenceMask . . . 802
8.842 unpack_swiaudio_SLQSGetM2MAVMute_t Struct Reference . . . 802
8.842.1 Detailed Description . . . 802
8.842.2 Field Documentation . . . 803
8.842.2.1 CwtMute . . . 803
8.842.2.2 EarMute . . . 803
8.842.2.3 MicMute . . . 803
8.842.2.4 ParamPresenceMask . . . 803
8.843 unpack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference . . . 803
8.843.1 Detailed Description . . . 803
8.843.2 Field Documentation . . . 804
8.843.2.1 ParamPresenceMask . . . 804
8.843.2.2 Value . . . 804
8.844 unpack_swiavms_SLQSAVMSEventReportInd_t Struct Reference . . . 804
8.844.1 Detailed Description . . . 804
8.844.2 Field Documentation . . . 805
8.844.2.1 ParamPresenceMask . . . 805
8.844.2.2 pBinaryUpdateSessionInfo . . . 805
8.844.2.3 pConfig . . . 805
8.844.2.4 pConnectionRequest . . . 805
8.844.2.5 pDataSessionStatus . . . 805
8.844.2.6 pHTTPStatus . . . 805
8.844.2.7 pNotification . . . 805
8.844.2.8 pPackageID . . . 805
8.844.2.9 pRegStatus . . . 805
8.844.2.10 pSessionType . . . 805
8.844.2.11 pWAMSParaChanged . . . 805
8.844.2.12 resultcode . . . 805
8.845 unpack_swiavms_SLQSAVMSGetSettings_t Struct Reference . . . 805
8.845.1 Detailed Description . . . 806
8.845.2 Field Documentation . . . 807
8.845.2.1 AutoConnect . . . 807
8.845.2.2 AutoReboot . . . 807
8.845.2.3 FwAutoSDM . . . 807
8.845.2.4 FwPromptdownload . . . 808
8.845.2.5 FwPromptUpdate . . . 808
8.845.2.6 OMADMEnabled . . . 808
8.845.2.7 pAPNInfo . . . 808
8.845.2.8 ParamPresenceMask . . . 808
8.845.2.9 pConnectionRetryTimers . . . 808

8.845.2.10 pNotificationStore . . . 808
8.845.2.11 pPeroidsInfo . . . 808
8.845.2.12 pPollingTimer . . . 808
8.845.2.13 resultcode . . . 808
8.846 unpack_swiavms_SLQSAVMSGetSettings_v2_t Struct Reference . . . 808
8.846.1 Detailed Description . . . 808
8.846.2 Field Documentation . . . 810
8.846.2.1 AutoConnect . . . 810
8.846.2.2 FwAutoSDM . . . 810
8.846.2.3 FwPromptdownload . . . 810
8.846.2.4 FwPromptUpdate . . . 810
8.846.2.5 OMADMEnabled . . . 810
8.846.2.6 pAPNInfo . . . 810
8.846.2.7 ParamPresenceMask . . . 810
8.846.2.8 pAutoReboot . . . 810
8.846.2.9 pConnectionRetryTimers . . . 810
8.846.2.10 pNotificationStore . . . 810
8.846.2.11 pPeroidsInfo . . . 810
8.846.2.12 pPollingTimer . . . 810
8.846.2.13 resultcode . . . 810
8.847 unpack_swiavms_SLQSAVMSSendSelection_t Struct Reference . . . 810
8.847.1 Detailed Description . . . 810
8.847.2 Field Documentation . . . 811
8.847.2.1 ParamPresenceMask . . . 811
8.847.2.2 resultcode . . . 811
8.848 unpack_swiavms_SLQSAVMSSessionGetInfo_t Struct Reference . . . 811
8.848.1 Detailed Description . . . 811
8.848.2 Field Documentation . . . 811
8.848.2.1 ParamPresenceMask . . . 811
8.848.2.2 pBinaryUpdateSessionInfo . . . 811
8.848.2.3 pConfig . . . 812
8.848.2.4 pNotification . . . 812
8.848.2.5 pPackageID . . . 812
8.848.2.6 resultcode . . . 812
8.849 unpack_swiavms_SLQSAvmsSetEventReport_t Struct Reference . . . 812
8.849.1 Detailed Description . . . 812
8.849.2 Field Documentation . . . 812
8.849.2.1 ParamPresenceMask . . . 812
8.849.2.2 resultcode . . . 812
8.850 unpack_swiavms_SLQSAVMSSetSettings_t Struct Reference . . . 812

8.850.1 Detailed Description . . . 812
8.850.2 Field Documentation . . . 812
8.850.2.1 ParamPresenceMask . . . 812
8.850.2.2 resultcode . . . 813
8.851 unpack_swiavms_SLQSAVMSSetSettings_v2_t Struct Reference . . . 813
8.851.1 Detailed Description . . . 813
8.851.2 Field Documentation . . . 813
8.851.2.1 ParamPresenceMask . . . 813
8.851.2.2 resultcode . . . 813
8.852 unpack_swiavms_SLQSAVMSStartSession_t Struct Reference . . . 813
8.852.1 Detailed Description . . . 813
8.852.2 Field Documentation . . . 813
8.852.2.1 ParamPresenceMask . . . 813
8.852.2.2 resultcode . . . 814
8.852.2.3 sessionResponse . . . 814
8.853 unpack_swiavms_SLQSAVMSStopSession_t Struct Reference . . . 814
8.853.1 Detailed Description . . . 814
8.853.2 Field Documentation . . . 814
8.853.2.1 ParamPresenceMask . . . 814
8.853.2.2 resultcode . . . 814
8.854 unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference . . . 814
8.854.1 Detailed Description . . . 814
8.854.2 Field Documentation . . . 815
8.854.2.1 ParamPresenceMask . . . 815
8.854.2.2 pEhrpdMTUSize . . . 815
8.854.2.3 pHrpdMTUSize . . . 815
8.854.2.4 pMTUSize3gpp . . . 815
8.854.2.5 pUsbMTUSize . . . 815
8.855 unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference . . . 815
8.855.1 Detailed Description . . . 815
8.855.2 Field Documentation . . . 815
8.855.2.1 ParamPresenceMask . . . 815
8.855.2.2 usbNetNum . . . 815
8.856 unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference . . . 816
8.856.1 Detailed Description . . . 816
8.856.2 Field Documentation . . . 816
8.856.2.1 ParamPresenceMask . . . 816
8.856.2.2 Tlvresult . . . 816
8.857 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . 816
8.857.1 Detailed Description . . . 816

8.857.2 Field Documentation . . . 818
8.857.2.1 fix_rate . . . 818
8.857.2.2 fix_rate_reported . . . 818
8.857.2.3 fix_type . . . 818
8.857.2.4 fix_type_reported . . . 818
8.857.2.5 function . . . 818
8.857.2.6 function_reported . . . 818
8.857.2.7 max_dist . . . 818
8.857.2.8 max_dist_reported . . . 818
8.857.2.9 max_time . . . 818
8.857.2.10 max_time_reported . . . 818
8.857.2.11 ParamPresenceMask . . . 818
8.858 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference . . . 818
8.858.1 Detailed Description . . . 818
8.858.2 Field Documentation . . . 819
8.858.2.1 eventType . . . 819
8.858.2.2 SessionInfoConfig . . . 819
8.858.2.3 SessionInfoFota . . . 819
8.858.2.4 SessionInfoNotification . . . 819
8.859 unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . 819
8.859.1 Detailed Description . . . 819
8.859.2 Field Documentation . . . 821
8.859.2.1 Date . . . 821
8.859.2.2 DateLength . . . 821
8.859.2.3 PkgDescLength . . . 821
8.859.2.4 PkgDescription . . . 821
8.859.2.5 PkgName . . . 821
8.859.2.6 PkgNameLength . . . 821
8.859.2.7 RetryCount . . . 821
8.859.2.8 SessionState . . . 821
8.859.2.9 SessionType . . . 821
8.859.2.10 Severity . . . 821
8.859.2.11 Source . . . 821
8.859.2.12 SourceLength . . . 821
8.859.2.13 Status . . . 821
8.859.2.14 Time . . . 821
8.859.2.15 TimeLength . . . 821
8.859.2.16 UpdateCompleteStatus . . . 821
8.860 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . 821
8.860.1 Detailed Description . . . 822

8.860.2 Field Documentation . . . 823
8.860.2.1 Autosdm . . . 823
8.860.2.2 FOTAdownload . . . 823
8.860.2.3 FOTAUpdate . . . 823
8.860.2.4 FwAutoCheck . . . 823
8.860.2.5 OMADMEnabled . . . 823
8.861 unpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . 823
8.861.1 Detailed Description . . . 823
8.861.2 Field Documentation . . . 823
8.861.2.1 FwAvailability . . . 823
8.862 unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference . . . 823
8.862.1 Detailed Description . . . 824
8.862.2 Field Documentation . . . 824
8.862.2.1 Tlvresult . . . 824
8.863 unpack_tmd_SLQSTmdGetMitigationDevList_t Struct Reference . . . 824
8.863.1 Detailed Description . . . 824
8.863.2 Field Documentation . . . 824
8.863.2.1 MitigationDevList . . . 824
8.863.2.2 MitigationDevListLen . . . 824
8.863.2.3 Tlvresult . . . 824
8.864 unpack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference . . . 824
8.864.1 Detailed Description . . . 825
8.864.2 Field Documentation . . . 825
8.864.2.1 CurrentmitigationLvl . . . 825
8.864.2.2 ReqMitigationLvl . . . 825
8.864.2.3 Tlvresult . . . 825
8.865 unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t Struct Reference . . . 825
8.865.1 Detailed Description . . . 825
8.865.2 Field Documentation . . . 826
8.865.2.1 deviceID . . . 826
8.865.2.2 deviceIdLen . . . 826
8.865.2.3 lvl . . . 826
8.866 unpack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference . . . 826
8.866.1 Detailed Description . . . 826
8.866.2 Field Documentation . . . 826
8.866.2.1 Tlvresult . . . 826
8.867 unpack_uim_ChangePin_t Struct Reference . . . 826
8.867.1 Detailed Description . . . 826
8.867.2 Field Documentation . . . 827
8.867.2.1 pEncryptedPIN1 . . . 827

8.867.2.2 pIndicationToken . . . 827
8.867.2.3 pRemainingRetries . . . 827
8.867.2.4 Tlvresult . . . 827
8.868 unpack_uim_GetCardStatus_t Struct Reference . . . 827
8.868.1 Detailed Description . . . 827
8.868.2 Field Documentation . . . 827
8.868.2.1 pCardStatus . . . 827
8.868.2.2 pHotSwapStatus . . . 828
8.868.2.3 Tlvresult . . . 828
8.869 unpack_uim_ReadTransparent_t Struct Reference . . . 828
8.869.1 Detailed Description . . . 828
8.869.2 Field Documentation . . . 828
8.869.2.1 pCardResult . . . 828
8.869.2.2 pEncryptedData . . . 828
8.869.2.3 pIndicationToken . . . 828
8.869.2.4 pReadResult . . . 828
8.869.2.5 Tlvresult . . . 829
8.870 unpack_uim_SetPinProtection_t Struct Reference . . . 829
8.870.1 Detailed Description . . . 829
8.870.2 Field Documentation . . . 829
8.870.2.1 pEncryptedPIN1 . . . 829
8.870.2.2 pIndicationToken . . . 829
8.870.2.3 pRemainingRetries . . . 829
8.870.2.4 Tlvresult . . . 829
8.871 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . 829
8.871.1 Detailed Description . . . 830
8.871.2 Field Documentation . . . 830
8.871.2.1 bNumberOfPhySlots . . . 830
8.871.2.2 slotsstatusChange . . . 830
8.872 unpack_uim_SLQSUIMAuthenticate_t Struct Reference . . . 830
8.872.1 Detailed Description . . . 830
8.872.2 Field Documentation . . . 830
8.872.2.1 pAuthenticateResult . . . 831
8.872.2.2 pCardResult . . . 831
8.872.2.3 pIndicationToken . . . 831
8.873 unpack_uim_SLQSUIMDepersonalization_t Struct Reference . . . 831
8.873.1 Detailed Description . . . 831
8.873.2 Field Documentation . . . 831
8.873.2.1 pRemainingRetries . . . 831
8.874 unpack_uim_SLQSUIMEventRegister_t Struct Reference . . . 831

8.874.1 Detailed Description . . . 831
8.874.2 Field Documentation . . . 831
8.874.2.1 eventMask . . . 832
8.875 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . 832
8.875.1 Detailed Description . . . 832
8.875.2 Field Documentation . . . 832
8.875.2.1 pAutoSelection . . . 832
8.875.2.2 pHaltSubscription . . . 832
8.875.2.3 pPersonalizationStatus . . . 832
8.876 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . 832
8.876.1 Detailed Description . . . 833
8.876.2 Field Documentation . . . 833
8.876.2.1 pCardResult . . . 833
8.876.2.2 pFileAttributes . . . 833
8.876.2.3 pIndicationToken . . . 833
8.877 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference . . . 833
8.877.1 Detailed Description . . . 833
8.877.2 Field Documentation . . . 834
8.877.2.1 pNumberOfPhySlot . . . 834
8.877.2.2 pUimSlotsStatus . . . 834
8.878 unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference . . . 834
8.878.1 Detailed Description . . . 834
8.878.2 Field Documentation . . . 834
8.878.2.1 refreshEvent . . . 834
8.878.2.2 TlvPresent . . . 834
8.879 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . 834
8.879.1 Detailed Description . . . 834
8.879.2 Field Documentation . . . 834
8.879.2.1 pRefreshEvent . . . 835
8.880 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . 835
8.880.1 Detailed Description . . . 835
8.880.2 Field Documentation . . . 835
8.880.2.1 pCardStatus . . . 835
8.881 unpack_uim_UnblockPin_t Struct Reference . . . 835
8.881.1 Detailed Description . . . 835
8.881.2 Field Documentation . . . 836
8.881.2.1 pEncryptedPIN1 . . . 836
8.881.2.2 pIndicationToken . . . 836
8.881.2.3 pRemainingRetries . . . 836
8.881.2.4 Tlvresult . . . 836

8.882 unpack_uim_VerifyPin_t Struct Reference . . . 836
8.882.1 Detailed Description . . . 836
8.882.2 Field Documentation . . . 836
8.882.2.1 pEncryptedPIN1 . . . 836
8.882.2.2 pIndicationToken . . . 836
8.882.2.3 pRemainingRetries . . . 836
8.882.2.4 Tlvresult . . . 836
8.883 unpack_voice_allCallStatusCallback_ind_t Struct Reference . . . 837
8.883.1 Detailed Description . . . 837
8.883.2 Field Documentation . . . 839
8.883.2.1 arrCallInfomation . . . 839
8.883.2.2 pArrAlertingPattern . . . 839
8.883.2.3 pArrAlertingType . . . 839
8.883.2.4 pArrAlphaID . . . 839
8.883.2.5 pArrCalledPartyNum . . . 839
8.883.2.6 pArrCallEndReason . . . 839
8.883.2.7 pArrConnectPartyNum . . . 839
8.883.2.8 pArrDiagInfo . . . 839
8.883.2.9 pArrRedirPartyNum . . . 839
8.883.2.10 pArrRemotePartyName . . . 839
8.883.2.11 pArrRemotePartyNum . . . 839
8.883.2.12 pArrSvcOption . . . 839
8.884 unpack_voice_DTMFEventCallback_ind_t Struct Reference . . . 839
8.884.1 Detailed Description . . . 840
8.884.2 Field Documentation . . . 840
8.884.2.1 DTMFInformation . . . 840
8.884.2.2 pOffLength . . . 840
8.884.2.3 pOnLength . . . 840
8.885 unpack_voice_OTASPStatusCallback_ind_t Struct Reference . . . 840
8.885.1 Detailed Description . . . 840
8.885.2 Field Documentation . . . 841
8.885.2.1 callID . . . 841
8.885.2.2 OTASPStatus . . . 841
8.886 unpack_voice_SLQSOriginateUSSD_t Struct Reference . . . 841
8.886.1 Detailed Description . . . 841
8.886.2 Field Documentation . . . 844
8.886.2.1 pAlphaIDInfo . . . 844
8.886.2.2 pCallId . . . 844
8.886.2.3 pCcResultType . . . 844
8.886.2.4 pCCSuppsType . . . 844

8.886.2.5 pfailureCause . . . 844
8.886.2.6 pUSSDInfo . . . 844
8.887 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . 844
8.887.1 Detailed Description . . . 844
8.887.2 Field Documentation . . . 845
8.887.2.1 pCallId . . . 845
8.888 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . 845
8.888.1 Detailed Description . . . 845
8.888.2 Field Documentation . . . 845
8.888.2.1 pCallID . . . 845
8.889 unpack_voice_SLQSVoiceDialCall_t Struct Reference . . . 845
8.889.1 Detailed Description . . . 845
8.889.2 Field Documentation . . . 846
8.889.2.1 pAlphaIDInfo . . . 846
8.889.2.2 pCallID . . . 846
8.889.2.3 pCCResultType . . . 846
8.889.2.4 pCCSUPSType . . . 846
8.890 unpack_voice_SLQSVoiceEndCall_t Struct Reference . . . 846
8.890.1 Detailed Description . . . 846
8.890.2 Field Documentation . . . 847
8.890.2.1 pCallId . . . 847
8.891 unpack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference . . . 847
8.891.1 Detailed Description . . . 847
8.891.2 Field Documentation . . . 849
8.891.2.1 pArrAlertingPattern . . . 849
8.891.2.2 pArrAlertingType . . . 849
8.891.2.3 pArrAlphaID . . . 849
8.891.2.4 pArrCalledPartyNum . . . 849
8.891.2.5 pArrCallEndReason . . . 849
8.891.2.6 pArrCallInfo . . . 849
8.891.2.7 pArrConnectPartyNum . . . 849
8.891.2.8 pArrDiagInfo . . . 849
8.891.2.9 pArrRedirPartyNum . . . 849
8.891.2.10 pArrRemotePartyName . . . 849
8.891.2.11 pArrRemotePartyNum . . . 849
8.891.2.12 pArrSvcOption . . . 849
8.891.2.13 pArrUUSInfo . . . 849
8.891.2.14 pOTASPStatus . . . 849
8.891.2.15 pVoicePrivacy . . . 849
8.892 unpack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . 849

8.892.1 Detailed Description . . . 850
8.892.2 Field Documentation . . . 851
8.892.2.1 pAlphaIDInfo . . . 851
8.892.2.2 pCallID . . . 851
8.892.2.3 pCCResType . . . 851
8.892.2.4 pCCSUPSType . . . 851
8.892.2.5 pFailCause . . . 851
8.892.2.6 pSvcClass . . . 851
8.893 unpack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . 851
8.893.1 Detailed Description . . . 851
8.893.2 Field Documentation . . . 852
8.893.2.1 pAlphaIDInfo . . . 852
8.893.2.2 pCallID . . . 852
8.893.2.3 pCCResType . . . 852
8.893.2.4 pCCSUPSType . . . 852
8.893.2.5 pFailCause . . . 852
8.893.2.6 pGetCallFWExtInfo . . . 852
8.893.2.7 pGetCallFWInfo . . . 852
8.894 unpack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . 852
8.894.1 Detailed Description . . . 853
8.894.2 Field Documentation . . . 855
8.894.2.1 pAlertingPattern . . . 855
8.894.2.2 pAlertType . . . 855
8.894.2.3 pAlphaIDInfo . . . 855
8.894.2.4 pCallInfo . . . 855
8.894.2.5 pConnectNumInfo . . . 855
8.894.2.6 pDiagInfo . . . 855
8.894.2.7 pOTASPStatus . . . 855
8.894.2.8 pRemotePartyName . . . 855
8.894.2.9 pRemotePartyNum . . . 855
8.894.2.10 pSrvOpt . . . 855
8.894.2.11 pUUSInfo . . . 855
8.894.2.12 pVoicePrivacy . . . 855
8.895 unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . 856
8.895.1 Detailed Description . . . 856
8.895.2 Field Documentation . . . 857
8.895.2.1 pAlphaIDInfo . . . 857
8.895.2.2 pCallID . . . 857
8.895.2.3 pCCResType . . . 857
8.895.2.4 pCCSUPSType . . . 857

8.895.2.5 pFailCause . . . 857
8.895.2.6 pSvcClass . . . 857
8.896 unpack_voice_SLQSVoiceGetCLIP_t Struct Reference . . . 857
8.896.1 Detailed Description . . . 857
8.896.2 Field Documentation . . . 858
8.896.2.1 pAlphaIDInfo . . . 858
8.896.2.2 pCallID . . . 858
8.896.2.3 pCCResType . . . 858
8.896.2.4 pCCSUPSType . . . 858
8.896.2.5 pCLIPResp . . . 858
8.896.2.6 pFailCause . . . 858
8.897 unpack_voice_SLQSVoiceGetCLIR_t Struct Reference . . . 858
8.897.1 Detailed Description . . . 859
8.897.2 Field Documentation . . . 859
8.897.2.1 pAlphaIDInfo . . . 859
8.897.2.2 pCallID . . . 859
8.897.2.3 pCCResType . . . 860
8.897.2.4 pCCSUPSType . . . 860
8.897.2.5 pCLIRResp . . . 860
8.897.2.6 pFailCause . . . 860
8.898 unpack_voice_SLQSVoiceGetCNAP_t Struct Reference . . . 860
8.898.1 Detailed Description . . . 860
8.898.2 Field Documentation . . . 861
8.898.2.1 pAlphaIDInfo . . . 861
8.898.2.2 pCallID . . . 861
8.898.2.3 pCCResType . . . 861
8.898.2.4 pCCSUPSType . . . 861
8.898.2.5 pCNAPResp . . . 861
8.898.2.6 pFailCause . . . 861
8.899 unpack_voice_SLQSVoiceGetCOLP_t Struct Reference . . . 861
8.899.1 Detailed Description . . . 861
8.899.2 Field Documentation . . . 862
8.899.2.1 pAlphaIDInfo . . . 862
8.899.2.2 pCallID . . . 862
8.899.2.3 pCCResType . . . 862
8.899.2.4 pCCSUPSType . . . 862
8.899.2.5 pCOLPResp . . . 862
8.899.2.6 pFailCause . . . 862
8.900 unpack_voice_SLQSVoiceGetCOLR_t Struct Reference . . . 862
8.900.1 Detailed Description . . . 863

8.900.2 Field Documentation . . . 864
8.900.2.1 pAlphaIDInfo . . . 864
8.900.2.2 pCallID . . . 864
8.900.2.3 pCCResType . . . 864
8.900.2.4 pCCSUPSType . . . 864
8.900.2.5 pCOLRResp . . . 864
8.900.2.6 pFailCause . . . 864
8.901 unpack_voice_SLQSVoiceGetConfig_t Struct Reference . . . 864
8.901.1 Detailed Description . . . 864
8.901.2 Field Documentation . . . 866
8.901.2.1 pAirTimerCnt . . . 866
8.901.2.2 pAutoAnswerStat . . . 866
8.901.2.3 pCurAMRConfig . . . 866
8.901.2.4 pCurPrefVoiceSO . . . 866
8.901.2.5 pCurrTTYMode . . . 866
8.901.2.6 pCurVoiceDomainPref . . . 866
8.901.2.7 pCurVoicePrivacyPref . . . 866
8.901.2.8 pRoamTimerCnt . . . 866
8.902 unpack_voice_SLQSVoiceManageCalls_t Struct Reference . . . 866
8.902.1 Detailed Description . . . 866
8.902.2 Field Documentation . . . 866
8.902.2.1 pFailCause . . . 866
8.903 unpack_voice_SLQSVoiceSendFlash_t Struct Reference . . . 866
8.903.1 Detailed Description . . . 866
8.903.2 Field Documentation . . . 867
8.903.2.1 pCallID . . . 867
8.904 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . 867
8.904.1 Detailed Description . . . 867
8.904.2 Field Documentation . . . 868
8.904.2.1 pAlphaIDInfo . . . 868
8.904.2.2 pCallID . . . 868
8.904.2.3 pCCResType . . . 868
8.904.2.4 pCCSUPSType . . . 868
8.904.2.5 pFailCause . . . 868
8.905 unpack_voice_SLQSVoiceSetConfig_t Struct Reference . . . 868
8.905.1 Detailed Description . . . 868
8.905.2 Field Documentation . . . 869
8.905.2.1 pAirTimerStatus . . . 869
8.905.2.2 pAutoAnsStatus . . . 869
8.905.2.3 pPrefVoiceSOStatus . . . 870

8.905.2.4 pRoamTimerStatus . . . 870
8.905.2.5 pTTYConfigStatus . . . 870
8.905.2.6 pVoiceDomainPrefStatus . . . 870
8.906 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . 870
8.906.1 Detailed Description . . . 870
8.906.2 Field Documentation . . . 871
8.906.2.1 pAlphaIDInfo . . . 871
8.906.2.2 pCallID . . . 871
8.906.2.3 pCCResultType . . . 871
8.906.2.4 pCCSUPSType . . . 871
8.906.2.5 pFailCause . . . 871
8.907 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . 871
8.907.1 Detailed Description . . . 871
8.907.2 Field Documentation . . . 871
8.907.2.1 pCallID . . . 871
8.908 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . 871
8.908.1 Detailed Description . . . 872
8.908.2 Field Documentation . . . 872
8.908.2.1 callID . . . 872
8.909 unpack_voice_SLQSVoiceSUPSCallback_ind_t Struct Reference . . . 872
8.909.1 Detailed Description . . . 872
8.909.2 Field Documentation . . . 874
8.909.2.1 pAlphaIDInfo . . . 874
8.909.2.2 pCallBarPasswd . . . 874
8.909.2.3 pCallFwdInfo . . . 874
8.909.2.4 pCallFWNum . . . 874
8.909.2.5 pCallFWTimerVal . . . 874
8.909.2.6 pCallID . . . 874
8.909.2.7 pCLIPstatus . . . 874
8.909.2.8 pCLIRstatus . . . 874
8.909.2.9 pCNAPstatus . . . 874
8.909.2.10 pCOLPstatus . . . 874
8.909.2.11 pCOLRstatus . . . 874
8.909.2.12 pDataSrc . . . 874
8.909.2.13 pFailCause . . . 874
8.909.2.14 pNewPwdData . . . 874
8.909.2.15 pReason . . . 874
8.909.2.16 pSvcClass . . . 874
8.909.2.17 pUSSInfo . . . 874
8.909.2.18 SUPSInformation . . . 874

8.910 unpack_voice_SUPSNotificationCallback_ind_t Struct Reference . . . 874
8.910.1 Detailed Description . . . 875
8.910.2 Field Documentation . . . 876
8.910.2.1 callID . . . 876
8.910.2.2 notifType . . . 876
8.910.2.3 pCUGIndex . . . 876
8.910.2.4 pECTNum . . . 876
8.911 unpack_voice_USSDNotificationCallback_ind_t Struct Reference . . . 876
8.911.1 Detailed Description . . . 876
8.911.2 Field Documentation . . . 876
8.911.2.1 notification_Type . . . 876
8.911.2.2 USSDNotificationNetworkInfo . . . 876
8.912 unpack_voice_VoiceInfoRecCallback_ind_t Struct Reference . . . 876
8.912.1 Detailed Description . . . 877
8.912.2 Field Documentation . . . 878
8.912.2.1 callID . . . 878
8.912.2.2 pCalledPartyInfo . . . 878
8.912.2.3 pCallerIDInfo . . . 878
8.912.2.4 pCallerNameInfo . . . 878
8.912.2.5 pCallingPartyInfo . . . 878
8.912.2.6 pCallWaitInd . . . 878
8.912.2.7 pCLIRCause . . . 878
8.912.2.8 pConnectNumInfo . . . 878
8.912.2.9 pDispInfo . . . 878
8.912.2.10 pExtDispInfo . . . 878
8.912.2.11 pExtDispRecInfo . . . 878
8.912.2.12 pLineCtrlInfo . . . 878
8.912.2.13 pNSSAudioCtrl . . . 879
8.912.2.14 pNSSRelease . . . 879
8.912.2.15 pRedirNumInfo . . . 879
8.912.2.16 pSignalInfo . . . 879
8.913 unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference . . . 879
8.913.1 Detailed Description . . . 879
8.913.2 Field Documentation . . . 879
8.913.2.1 callID . . . 879
8.913.2.2 voicePrivacy . . . 879
8.914 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference . . . 879
8.914.1 Detailed Description . . . 879
8.914.2 Field Documentation . . . 880
8.914.2.1 DHCPv4LeaseOptTlv . . . 880

8.914.2.2 DHCPv4LeaseStateTlv . . . 880
8.914.2.3 IPv4AddrTlv . . . 880
8.914.2.4 ProfileIdTlv . . . 880
8.915 unpack_wds_GetAutoconnect_t Struct Reference . . . 880
8.915.1 Detailed Description . . . 880
8.915.2 Field Documentation . . . 880
8.915.2.1 psetting . . . 880
8.916 unpack_wds_GetByteTotals_t Struct Reference . . . 880
8.916.1 Detailed Description . . . 880
8.916.2 Field Documentation . . . 881
8.916.2.1 pRXTotalBytes . . . 881
8.916.2.2 pTXTotalBytes . . . 881
8.917 unpack_wds_GetConnectionRate_t Struct Reference . . . 881
8.917.1 Detailed Description . . . 881
8.917.2 Field Documentation . . . 881
8.917.2.1 currentChannelRXRate . . . 881
8.917.2.2 currentChannelTXRate . . . 881
8.917.2.3 maxChannelRXRate . . . 881
8.917.2.4 maxChannelTXRate . . . 882
8.918 unpack_wds_GetDataBearerTechnology_t Struct Reference . . . 882
8.918.1 Detailed Description . . . 882
8.918.2 Field Documentation . . . 882
8.918.2.1 pDataBearer . . . 882
8.919 unpack_wds_GetDefaultProfile_t Struct Reference . . . 882
8.919.1 Detailed Description . . . 883
8.919.2 Field Documentation . . . 884
8.919.2.1 apnname . . . 884
8.919.2.2 apnsize . . . 884
8.919.2.3 auth . . . 884
8.919.2.4 ipaddr . . . 884
8.919.2.5 ipaddrv6 . . . 884
8.919.2.6 name . . . 884
8.919.2.7 namesize . . . 885
8.919.2.8 pdptype . . . 885
8.919.2.9 pridns . . . 885
8.919.2.10 pridnsv6 . . . 885
8.919.2.11 secdns . . . 885
8.919.2.12 secdnsv6 . . . 885
8.919.2.13 username . . . 885
8.919.2.14 usersize . . . 885

8.920 unpack_wds_GetDefaultProfileNum_t Struct Reference . . . 885
8.920.1 Detailed Description . . . 885
8.920.2 Field Documentation . . . 885
8.920.2.1 index . . . 885
8.921 unpack_wds_GetDormancyState_t Struct Reference . . . 885
8.921.1 Detailed Description . . . 885
8.921.2 Field Documentation . . . 886
8.921.2.1 dormancyState . . . 886
8.922 unpack_wds_GetLastMobileIPError_t Struct Reference . . . 886
8.922.1 Detailed Description . . . 886
8.922.2 Field Documentation . . . 886
8.922.2.1 error . . . 886
8.923 unpack_wds_GetMobileIP_t Struct Reference . . . 886
8.923.1 Detailed Description . . . 886
8.923.2 Field Documentation . . . 886
8.923.2.1 mipMode . . . 886
8.924 unpack_wds_GetMobileIPProfile_t Struct Reference . . . 887
8.924.1 Detailed Description . . . 887
8.924.2 Field Documentation . . . 888
8.924.2.1 AAASPI . . . 888
8.924.2.2 AAAState . . . 888
8.924.2.3 address . . . 888
8.924.2.4 enabled . . . 888
8.924.2.5 HASPI . . . 888
8.924.2.6 HAState . . . 888
8.924.2.7 NAI . . . 888
8.924.2.8 naiSize . . . 888
8.924.2.9 primaryHA . . . 888
8.924.2.10 revTunneling . . . 888
8.924.2.11 secondaryHA . . . 888
8.925 unpack_wds_GetPacketStatistics_t Struct Reference . . . 888
8.925.1 Detailed Description . . . 889
8.925.2 Field Documentation . . . 890
8.925.2.1 pRXDroppedCount . . . 890
8.925.2.2 pRXOkBytesCount . . . 890
8.925.2.3 pRXOKBytesLastCall . . . 890
8.925.2.4 pRXPacketErrors . . . 890
8.925.2.5 pRXPacketOverflows . . . 890
8.925.2.6 pRXPacketSuccesses . . . 890
8.925.2.7 pTXDroppedCount . . . 890

8.925.2.8 pTXOkBytesCount . . . 890
8.925.2.9 pTXOKBytesLastCall . . . 890
8.925.2.10 pTXPacketErrors . . . 890
8.925.2.11 pTXPacketOverflows . . . 890
8.925.2.12 pTXPacketSuccesses . . . 890
8.926 unpack_wds_GetPacketStatus_t Struct Reference . . . 890
8.926.1 Detailed Description . . . 891
8.926.2 Field Documentation . . . 892
8.926.2.1 rXDroppedCount . . . 892
8.926.2.2 rXOkBytesCount . . . 892
8.926.2.3 rXOKBytesLastCall . . . 892
8.926.2.4 rXPacketErrors . . . 892
8.926.2.5 rXPacketOverflows . . . 892
8.926.2.6 rXPacketSuccesses . . . 892
8.926.2.7 tXDroppedCount . . . 892
8.926.2.8 tXOkBytesCount . . . 892
8.926.2.9 tXOKBytesLastCall . . . 892
8.926.2.10 tXPacketErrors . . . 892
8.926.2.11 tXPacketOverflows . . . 892
8.926.2.12 tXPacketSuccesses . . . 892
8.927 unpack_wds_GetSessionDuration_t Struct Reference . . . 892
8.927.1 Detailed Description . . . 892
8.927.2 Field Documentation . . . 893
8.927.2.1 callDuration . . . 893
8.928 unpack_wds_GetSessionState_t Struct Reference . . . 893
8.928.1 Detailed Description . . . 893
8.928.2 Field Documentation . . . 893
8.928.2.1 connectionStatus . . . 893
8.929 unpack_wds_RMSetTransferStatistics_t Struct Reference . . . 893
8.929.1 Detailed Description . . . 893
8.930 unpack_wds_SetMobileIPProfile_t Struct Reference . . . 893
8.930.1 Detailed Description . . . 893
8.931 unpack_wds_SLQSCreateProfile_t Struct Reference . . . 894
8.931.1 Detailed Description . . . 894
8.931.2 Field Documentation . . . 894
8.931.2.1 pCreateProfileOut . . . 894
8.931.2.2 pProfileID . . . 894
8.931.2.3 Tlvresult . . . 894
8.932 unpack_wds_SLQSDeleteProfile_t Struct Reference . . . 894
8.932.1 Detailed Description . . . 894

8.932.2 Field Documentation . . . 895
8.932.2.1 extendedErrorCode . . . 895
8.933 unpack_wds_SLQSDUNCallInfoCallBack_ind_t Struct Reference . . . 895
8.933.1 Detailed Description . . . 895
8.933.2 Field Documentation . . . 896
8.933.2.1 CRTlv . . . 896
8.933.2.2 CSTlv . . . 896
8.933.2.3 DBTTlv . . . 896
8.933.2.4 DSTlv . . . 896
8.933.2.5 LMCERTlv . . . 896
8.933.2.6 RXBOTlv . . . 896
8.933.2.7 TXBOTlv . . . 896
8.934 unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . 896
8.934.1 Detailed Description . . . 896
8.934.2 Field Documentation . . . 897
8.934.2.1 _3gppRelease . . . 897
8.934.2.2 defaultPDNEnabled . . . 897
8.934.2.3 LTEAttachProfile . . . 897
8.934.2.4 LTEAttachProfileList . . . 897
8.934.2.5 LTEAttachProfileListLen . . . 898
8.934.2.6 profileList . . . 898
8.935 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . 898
8.935.1 Detailed Description . . . 898
8.935.2 Field Documentation . . . 898
8.935.2.1 currNetworkInfo . . . 898
8.935.2.2 networkInfoLen . . . 898
8.935.2.3 prefNetwork . . . 898
8.936 unpack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . 898
8.936.1 Detailed Description . . . 899
8.936.2 Field Documentation . . . 899
8.936.2.1 current_channel_rx_rate . . . 899
8.936.2.2 current_channel_tx_rate . . . 899
8.936.2.3 max_channel_rx_rate . . . 899
8.936.2.4 max_channel_tx_rate . . . 899
8.937 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . 899
8.937.1 Detailed Description . . . 899
8.937.2 Field Documentation . . . 900
8.937.2.1 curDataBearerTechnology . . . 900
8.937.2.2 dataBearerMask . . . 900
8.937.2.3 lastCallDataBearerTechnology . . . 900

8.938 unpack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . 900
8.938.1 Detailed Description . . . 900
8.938.2 Field Documentation . . . 902
8.938.2.1 callEndReason . . . 902
8.938.2.2 channelRate . . . 902
8.938.2.3 connectionStatus . . . 902
8.938.2.4 dataBearerTech . . . 903
8.938.2.5 dormancyStatus . . . 903
8.938.2.6 lastCallDataBearerTech . . . 903
8.938.2.7 lastCallRXOKBytesCnt . . . 903
8.938.2.8 lastCallTXOKBytesCnt . . . 903
8.938.2.9 mdmCallDurationActive . . . 903
8.938.2.10 rxOKBytesCount . . . 903
8.938.2.11 txOKBytesCount . . . 903
8.939 unpack_wds_SLQSGetProfileSettings_t Struct Reference . . . 903
8.939.1 Detailed Description . . . 903
8.939.2 Field Documentation . . . 903
8.939.2.1 pProfileSettings . . . 903
8.939.2.2 ProfileType . . . 903
8.939.2.3 Tlvresult . . . 903
8.940 unpack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . 903
8.940.1 Detailed Description . . . 904
8.940.2 Field Documentation . . . 905
8.940.2.1 APNName . . . 905
8.940.2.2 Authentication . . . 905
8.940.2.3 DomainList . . . 905
8.940.2.4 GPRSGrantedQoS . . . 906
8.940.2.5 GWAddressV4 . . . 906
8.940.2.6 IMCNflag . . . 906
8.940.2.7 IPFamilyPreference . . . 906
8.940.2.8 IPv4 . . . 906
8.940.2.9 IPV6AddrInfo . . . 906
8.940.2.10 IPV6GWAddrInfo . . . 906
8.940.2.11 Mtu . . . 906
8.940.2.12 PCSCFAddrPCO . . . 906
8.940.2.13 PCSCFFQDNAddrList . . . 906
8.940.2.14 PDPType . . . 906
8.940.2.15 PrimaryDNSV4 . . . 906
8.940.2.16 PrimaryDNSV6 . . . 906
8.940.2.17 ProfileID . . . 906

8.940.2.18 ProfileName . . . 906
8.940.2.19 SecondaryDNSV4 . . . 906
8.940.2.20 SecondaryDNSV6 . . . 906
8.940.2.21 ServerAddrList . . . 906
8.940.2.22 SubnetMaskV4 . . . 906
8.940.2.23 Technology . . . 906
8.940.2.24 UMTSGrantedQoS . . . 906
8.940.2.25 Username . . . 906
8.941 unpack_wds_SLQSModifyProfile_t Struct Reference . . . 906
8.941.1 Detailed Description . . . 906
8.941.2 Field Documentation . . . 907
8.941.2.1 pExtErrorCode . . . 907
8.942 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . 907
8.942.1 Detailed Description . . . 907
8.942.2 Field Documentation . . . 907
8.942.2.1 Tlvresult . . . 907
8.943 unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . 907
8.943.1 Detailed Description . . . 907
8.943.2 Field Documentation . . . 909
8.943.2.1 bearerID . . . 909
8.943.2.2 conn_status . . . 909
8.943.2.3 ipFamily . . . 909
8.943.2.4 reconfigReqd . . . 909
8.943.2.5 sessionEndReason . . . 909
8.943.2.6 techName . . . 909
8.943.2.7 verboseSessnEndReason . . . 909
8.943.2.8 verboseSessnEndReasonType . . . 909
8.944 unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . 909
8.944.1 Detailed Description . . . 910
8.944.2 Field Documentation . . . 913
8.944.2.1 currDBTechAvail . . . 913
8.944.2.2 currNWInfo . . . 913
8.944.2.3 dataSysStatAvail . . . 913
8.944.2.4 dBTechAvail . . . 913
8.944.2.5 dBtechExtAvail . . . 913
8.944.2.6 dBTechExtRatValue . . . 913
8.944.2.7 dBTechExtSoMask . . . 914
8.944.2.8 dBTechnology . . . 914
8.944.2.9 dBtechnologyExt . . . 914
8.944.2.10 dormancyStatAvail . . . 914

8.944.2.11 dormancyStatus . . . 914
8.944.2.12 mipstatAvail . . . 914
8.944.2.13 mipStatus . . . 914
8.944.2.14 netInfoLen . . . 914
8.944.2.15 prefNetwork . . . 914
8.944.2.16 ratMask . . . 914
8.944.2.17 rx_bytes . . . 914
8.944.2.18 rx_pkts . . . 914
8.944.2.19 soMask . . . 914
8.944.2.20 tx_bytes . . . 914
8.944.2.21 tx_pkts . . . 914
8.944.2.22 xferStatAvail . . . 914
8.945 unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference . . . 914
8.945.1 Detailed Description . . . 914
8.945.2 Field Documentation . . . 915
8.945.2.1 pHwConfig . . . 915
8.945.2.2 pRequestOptionList . . . 915
8.946 unpack_wds_SLQSSGetLoopback_t Struct Reference . . . 915
8.946.1 Detailed Description . . . 915
8.946.2 Field Documentation . . . 915
8.946.2.1 ByteLoopbackMode . . . 915
8.946.2.2 ByteLoopbackMultiplier . . . 915
8.947 unpack_wds_SLQSStartDataSession_t Struct Reference . . . 915
8.947.1 Detailed Description . . . 915
8.947.2 Field Documentation . . . 916
8.947.2.1 pFailureReason . . . 916
8.947.2.2 psid . . . 916
8.947.2.3 pVerboseFailReasonType . . . 916
8.947.2.4 pVerboseFailureReason . . . 916
8.948 unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . 916
8.948.1 Detailed Description . . . 917
8.948.2 Field Documentation . . . 918
8.948.2.1 apnName . . . 918
8.948.2.2 bearerId . . . 918
8.948.2.3 contextId . . . 918
8.948.2.4 ipv4Address . . . 918
8.948.2.5 ipv4GWAddress . . . 918
8.948.2.6 ipv6Address . . . 918
8.948.2.7 ipv6GWAddress . . . 918
8.948.2.8 prDNSIPv4Address . . . 918

8.948.2.9 prDNSIPv6Address . . . 918
8.948.2.10 prPCSCFIPv4Address . . . 918
8.948.2.11 prPCSCFIPv6Address . . . 918
8.948.2.12 seDNSIPv4Address . . . 918
8.948.2.13 seDNSIPv6Address . . . 918
8.948.2.14 sePCSCFIPv4Address . . . 918
8.948.2.15 sePCSCFIPv6Address . . . 918
8.949 UnPackGetProfileSettingOut Struct Reference . . . 918
8.949.1 Detailed Description . . . 919
8.949.2 Field Documentation . . . 919
8.949.2.1 curProfile . . . 919
8.949.2.2 pExtErrCode . . . 919
8.950 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo Struct Reference . . . 919
8.950.1 Detailed Description . . . 919
8.950.2 Field Documentation . . . 920
8.950.2.1 bBinaryType . . . 920
8.950.2.2 bSerity . . . 920
8.950.2.3 bState . . . 920
8.950.2.4 bUserInputRequest . . . 920
8.950.2.5 szDescription . . . 920
8.950.2.6 szName . . . 921
8.950.2.7 szVersion . . . 921
8.950.2.8 TlvPresent . . . 921
8.950.2.9 ulPkgDownloadComplete . . . 921
8.950.2.10 ulPkgDownloadSize . . . 921
8.950.2.11 wDescriptionLength . . . 921
8.950.2.12 wNameLength . . . 921
8.950.2.13 wUpdateCompeteStatus . . . 921
8.950.2.14 wUserInputTimeout . . . 921
8.950.2.15 wVersionLength . . . 921
8.951 UnpackSwiAvmsEventReportConfig Struct Reference . . . 921
8.951.1 Detailed Description . . . 921
8.951.2 Field Documentation . . . 922
8.951.2.1 bState . . . 922
8.951.2.2 bUserInputRequest . . . 922
8.951.2.3 szAlertMsg . . . 922
8.951.2.4 TlvPresent . . . 922
8.951.2.5 wAlertMsgLength . . . 922
8.951.2.6 wUserInputTimeout . . . 922
8.952 UnpackSwiAvmsEventReportConnectionRequest Struct Reference . . . 922

8.952.1 Detailed Description . . . 922
8.952.2 Field Documentation . . . 922
8.952.2.1 bUserInputRequest . . . 922
8.952.2.2 TlvPresent . . . 922
8.952.2.3 wUserInputTimeout . . . 922
8.953 UnpackSwiAvmsEventReportDataSessionStatus Struct Reference . . . 922
8.953.1 Detailed Description . . . 923
8.953.2 Field Documentation . . . 923
8.953.2.1 bType . . . 923
8.953.2.2 TlvPresent . . . 923
8.953.2.3 wErrorCode . . . 923
8.954 UnpackSwiAvmsEventReportHTTPStatus Struct Reference . . . 923
8.954.1 Detailed Description . . . 923
8.954.2 Field Documentation . . . 924
8.954.2.1 TlvPresent . . . 924
8.954.2.2 wHTTPStatus . . . 925
8.955 UnpackSwiAvmsEventReportNotification Struct Reference . . . 925
8.955.1 Detailed Description . . . 925
8.955.2 Field Documentation . . . 925
8.955.2.1 bNotification . . . 926
8.955.2.2 TlvPresent . . . 926
8.955.2.3 wSessionStatus . . . 926
8.956 UnpackSwiAvmsEventReportPackageID Struct Reference . . . 926
8.956.1 Detailed Description . . . 926
8.956.2 Field Documentation . . . 926
8.956.2.1 bPackageID . . . 926
8.956.2.2 TlvPresent . . . 926
8.957 UnpackSwiAvmsEventReportRegStatus Struct Reference . . . 926
8.957.1 Detailed Description . . . 926
8.957.2 Field Documentation . . . 926
8.957.2.1 bRegStatus . . . 926
8.957.2.2 TlvPresent . . . 927
8.958 UnpackSwiAvmsEventReportSessionType Struct Reference . . . 927
8.958.1 Detailed Description . . . 927
8.958.2 Field Documentation . . . 927
8.958.2.1 bType . . . 927
8.958.2.2 TlvPresent . . . 927
8.959 UnpackSwiAvmsEventReportWAMSParamChange Struct Reference . . . 927
8.959.1 Detailed Description . . . 927
8.959.2 Field Documentation . . . 928

8.959.2.1 TlvPresent . . . 928
8.959.2.2 wWamsChangeMask . . . 928
8.960 unpackWdsProfileParam Union Reference . . . 928
8.960.1 Detailed Description . . . 928
8.960.2 Field Documentation . . . 928
8.960.2.1 SlqsProfile3GPP . . . 928
8.960.2.2 SlqsProfile3GPP2 . . . 928
8.961 voice_airTimer Struct Reference . . . 928
8.961.1 Detailed Description . . . 928
8.961.2 Field Documentation . . . 929
8.961.2.1 airTimerValue . . . 929
8.961.2.2 namID . . . 929
8.962 voice_allCallsAlphaIDInfo Struct Reference . . . 929
8.962.1 Detailed Description . . . 929
8.962.2 Field Documentation . . . 929
8.962.2.1 AlphaIDInfo . . . 929
8.962.2.2 callID . . . 929
8.963 voice_allCallsDiagInfo Struct Reference . . . 929
8.963.1 Detailed Description . . . 929
8.963.2 Field Documentation . . . 930
8.963.2.1 callID . . . 930
8.963.2.2 DiagInfo . . . 930
8.964 voice_allCallsUUSInfo Struct Reference . . . 930
8.964.1 Detailed Description . . . 930
8.964.2 Field Documentation . . . 930
8.964.2.1 callID . . . 930
8.964.2.2 uusInfo . . . 930
8.965 voice_alphaIDInfo Struct Reference . . . 930
8.965.1 Detailed Description . . . 930
8.965.2 Field Documentation . . . 931
8.965.2.1 alphaDcs . . . 931
8.965.2.2 alphaLen . . . 931
8.965.2.3 alphaText . . . 931
8.966 voice_arrAlertingPattern Struct Reference . . . 931
8.966.1 Detailed Description . . . 931
8.966.2 Field Documentation . . . 932
8.966.2.1 alertingPattern . . . 932
8.966.2.2 callID . . . 932
8.966.2.3 numInstances . . . 932
8.967 voice_arrAlertingType Struct Reference . . . 932

8.967.1 Detailed Description . . . 932
8.967.2 Field Documentation . . . 933
8.967.2.1 AlertingType . . . 933
8.967.2.2 callID . . . 933
8.967.2.3 numInstances . . . 933
8.968 voice_arrAlphaID Struct Reference . . . 933
8.968.1 Detailed Description . . . 933
8.968.2 Field Documentation . . . 933
8.968.2.1 allCallsAlphaIDInfoArr . . . 933
8.968.2.2 numInstances . . . 933
8.969 voice_arrCalledPartyNum Struct Reference . . . 933
8.969.1 Detailed Description . . . 934
8.969.2 Field Documentation . . . 934
8.969.2.1 CalledPartyNum . . . 934
8.969.2.2 numInstances . . . 934
8.970 voice_arrCallEndReason Struct Reference . . . 934
8.970.1 Detailed Description . . . 934
8.970.2 Field Documentation . . . 934
8.970.2.1 callEndReason . . . 934
8.970.2.2 callID . . . 935
8.970.2.3 numInstances . . . 935
8.971 voice_arrCallInfo Struct Reference . . . 935
8.971.1 Detailed Description . . . 935
8.971.2 Field Documentation . . . 935
8.971.2.1 getAllCallInfo . . . 935
8.971.2.2 numInstances . . . 935
8.972 voice_arrConnectPartyNum Struct Reference . . . 935
8.972.1 Detailed Description . . . 935
8.972.2 Field Documentation . . . 936
8.972.2.1 ConnectedPartyNum . . . 936
8.972.2.2 numInstances . . . 936
8.973 voice_arrDiagInfo Struct Reference . . . 936
8.973.1 Detailed Description . . . 936
8.973.2 Field Documentation . . . 936
8.973.2.1 DiagInfo . . . 936
8.973.2.2 numInstances . . . 936
8.974 voice_arrRedirPartyNum Struct Reference . . . 936
8.974.1 Detailed Description . . . 936
8.974.2 Field Documentation . . . 937
8.974.2.1 numInstances . . . 937

8.974.2.2 RedirPartyNum . . . 937
8.975 voice_arrRemotePartyName Struct Reference . . . 937
8.975.1 Detailed Description . . . 937
8.975.2 Field Documentation . . . 937
8.975.2.1 GetAllCallRmtPtyName . . . 937
8.975.2.2 numInstances . . . 937
8.976 voice_arrRemotePartyNum Struct Reference . . . 937
8.976.1 Detailed Description . . . 937
8.976.2 Field Documentation . . . 938
8.976.2.1 numInstances . . . 938
8.976.2.2 RmtPtyNum . . . 938
8.977 voice_arrSvcOption Struct Reference . . . 938
8.977.1 Detailed Description . . . 938
8.977.2 Field Documentation . . . 938
8.977.2.1 callID . . . 938
8.977.2.2 numInstances . . . 938
8.977.2.3 srvOption . . . 938
8.978 voice_arrUUSInfo Struct Reference . . . 938
8.978.1 Detailed Description . . . 939
8.978.2 Field Documentation . . . 939
8.978.2.1 AllCallsUUSInfo . . . 939
8.978.2.2 numInstances . . . 939
8.979 voice_burstDTMFInfo Struct Reference . . . 939
8.979.1 Detailed Description . . . 939
8.979.2 Field Documentation . . . 940
8.979.2.1 digitCnt . . . 940
8.979.2.2 pCallID . . . 940
8.979.2.3 pDigitBuff . . . 940
8.980 voice_calledPartyInfo Struct Reference . . . 940
8.980.1 Detailed Description . . . 940
8.980.2 Field Documentation . . . 941
8.980.2.1 number . . . 941
8.980.2.2 numLen . . . 941
8.980.2.3 numPlan . . . 941
8.980.2.4 numType . . . 941
8.980.2.5 PI . . . 941
8.980.2.6 SI . . . 941
8.981 voice_calledPartySubAdd Struct Reference . . . 941
8.981.1 Detailed Description . . . 942
8.981.2 Field Documentation . . . 942

8.981.2.1 extBit . . . 942
8.981.2.2 oddEvenInd . . . 942
8.981.2.3 subAddr . . . 942
8.981.2.4 subAddrLen . . . 942
8.981.2.5 subAddrType . . . 942
8.982 voice_callerIDInfo Struct Reference . . . 942
8.982.1 Detailed Description . . . 943
8.982.2 Field Documentation . . . 943
8.982.2.1 callerID . . . 943
8.982.2.2 callerIDLen . . . 943
8.982.2.3 PI . . . 943
8.983 voice_callFwdTypeAndPlan Struct Reference . . . 943
8.983.1 Detailed Description . . . 943
8.983.2 Field Documentation . . . 944
8.983.2.1 numberPlan . . . 944
8.983.2.2 numberType . . . 944
8.984 voice_callFWExtInfo Struct Reference . . . 944
8.984.1 Detailed Description . . . 944
8.984.2 Field Documentation . . . 945
8.984.2.1 noReplyTimer . . . 945
8.984.2.2 number . . . 945
8.984.2.3 numLen . . . 946
8.984.2.4 numPlan . . . 946
8.984.2.5 numType . . . 946
8.984.2.6 PI . . . 946
8.984.2.7 SI . . . 946
8.984.2.8 SvcClass . . . 946
8.984.2.9 SvcStatus . . . 946
8.985 voice_callFWInfo Struct Reference . . . 946
8.985.1 Detailed Description . . . 946
8.985.2 Field Documentation . . . 946
8.985.2.1 noReplyTimer . . . 946
8.985.2.2 number . . . 947
8.985.2.3 numLen . . . 947
8.985.2.4 SvcClass . . . 947
8.985.2.5 SvcStatus . . . 947
8.986 voice_callInfo Struct Reference . . . 947
8.986.1 Detailed Description . . . 947
8.986.2 Field Documentation . . . 948
8.986.2.1 callID . . . 948

8.986.2.2 callState . . . 948
8.986.2.3 callType . . . 948
8.986.2.4 direction . . . 948
8.986.2.5 mode . . . 948
8.987 voice_callingPartyInfo Struct Reference . . . 948
8.987.1 Detailed Description . . . 949
8.987.2 Field Documentation . . . 950
8.987.2.1 number . . . 950
8.987.2.2 numLen . . . 950
8.987.2.3 numPlan . . . 950
8.987.2.4 numType . . . 950
8.987.2.5 PI . . . 950
8.987.2.6 SI . . . 950
8.988 voice_ccSUPSType Struct Reference . . . 950
8.988.1 Detailed Description . . . 950
8.988.2 Field Documentation . . . 950
8.988.2.1 reason . . . 951
8.988.2.2 svcType . . . 951
8.989 voice_CLIPResp Struct Reference . . . 951
8.989.1 Detailed Description . . . 951
8.989.2 Field Documentation . . . 951
8.989.2.1 ActiveStatus . . . 951
8.989.2.2 ProvisionStatus . . . 951
8.990 voice_CLIRResp Struct Reference . . . 951
8.990.1 Detailed Description . . . 951
8.990.2 Field Documentation . . . 952
8.990.2.1 ActiveStatus . . . 952
8.990.2.2 ProvisionStatus . . . 952
8.991 voice_CNAPResp Struct Reference . . . 952
8.991.1 Detailed Description . . . 952
8.991.2 Field Documentation . . . 953
8.991.2.1 ActiveStatus . . . 953
8.991.2.2 ProvisionStatus . . . 953
8.992 voice_COLPResp Struct Reference . . . 953
8.992.1 Detailed Description . . . 953
8.992.2 Field Documentation . . . 953
8.992.2.1 ActiveStatus . . . 953
8.992.2.2 ProvisionStatus . . . 953
8.993 voice_COLRResp Struct Reference . . . 953
8.993.1 Detailed Description . . . 953

8.993.2 Field Documentation . . . 954
8.993.2.1 ActiveStatus . . . 954
8.993.2.2 ProvisionStatus . . . 954
8.994 voice_connectNumInfo Struct Reference . . . 954
8.994.1 Detailed Description . . . 954
8.994.2 Field Documentation . . . 955
8.994.2.1 callerID . . . 955
8.994.2.2 callerIDLen . . . 955
8.994.2.3 numPlan . . . 955
8.994.2.4 numPresInd . . . 956
8.994.2.5 numType . . . 956
8.994.2.6 screeningInd . . . 956
8.995 voice_CUGInfo Struct Reference . . . 956
8.995.1 Detailed Description . . . 956
8.995.2 Field Documentation . . . 956
8.995.2.1 CUGIndex . . . 956
8.995.2.2 SuppOA . . . 956
8.995.2.3 SuppPrefCUG . . . 956
8.996 voice_curAMRConfig Struct Reference . . . 956
8.996.1 Detailed Description . . . 956
8.996.2 Field Documentation . . . 957
8.996.2.1 gsmAmrStat . . . 957
8.996.2.2 wcdmaAmrStat . . . 957
8.997 voice_diagInfo Struct Reference . . . 957
8.997.1 Detailed Description . . . 957
8.997.2 Field Documentation . . . 957
8.997.2.1 diagInfoLen . . . 957
8.997.2.2 diagnosticInfo . . . 957
8.998 voice_DTMFInfo Struct Reference . . . 958
8.998.1 Detailed Description . . . 958
8.998.2 Field Documentation . . . 958
8.998.2.1 callID . . . 958
8.998.2.2 digitBuff . . . 958
8.998.2.3 digitCnt . . . 958
8.998.2.4 DTMFEvent . . . 958
8.999 voice_DTMFLengths Struct Reference . . . 958
8.999.1 Detailed Description . . . 959
8.999.2 Field Documentation . . . 959
8.999.2.1 DTMFInterdigitInterval . . . 959
8.999.2.2 DTMFPulseWidth . . . 959

8.1000 voice_ECTNum Struct Reference . . . 959
8.1000.1 Detailed Description . . . 959
8.1000.2 Field Documentation . . . 960
8.1000.2.1 ECTCallState . . . 960
8.1000.2.2 number . . . 960
8.1000.2.3 presentationInd . . . 960
8.1001 voice_extDispRecInfo Struct Reference . . . 960
8.1001.1 Detailed Description . . . 960
8.1001.2 Field Documentation . . . 960
8.1001.2.1 dispType . . . 960
8.1001.2.2 extDispInfo . . . 960
8.1001.2.3 extDispInfoLen . . . 961
8.1002 voice_getAllCallInformation Struct Reference . . . 961
8.1002.1 Detailed Description . . . 961
8.1002.2 Field Documentation . . . 961
8.1002.2.1 ALS . . . 961
8.1002.2.2 CallInfo . . . 961
8.1002.2.3 isEmpty . . . 961
8.1003 voice_getAllCallRmtPtyName Struct Reference . . . 961
8.1003.1 Detailed Description . . . 961
8.1003.2 Field Documentation . . . 962
8.1003.2.1 callID . . . 962
8.1003.2.2 RemotePartyName . . . 962
8.1004 voice_getAllCallRmtPtyNum Struct Reference . . . 962
8.1004.1 Detailed Description . . . 962
8.1004.2 Field Documentation . . . 962
8.1004.2.1 callID . . . 962
8.1004.2.2 RemotePartyNum . . . 962
8.1005 voice_getCallFWExtInfo Struct Reference . . . 962
8.1005.1 Detailed Description . . . 962
8.1005.2 Field Documentation . . . 963
8.1005.2.1 CallFWExtInfo . . . 963
8.1005.2.2 numInstances . . . 963
8.1006 voice_getCallFWInfo Struct Reference . . . 963
8.1006.1 Detailed Description . . . 963
8.1006.2 Field Documentation . . . 963
8.1006.2.1 CallFWInfo . . . 963
8.1006.2.2 numInstances . . . 963
8.1007 voice_lineCtrlInfo Struct Reference . . . 963
8.1007.1 Detailed Description . . . 964

8.1007.2 Field Documentation . . . 964
8.1007.2.1 polarityIncluded . . . 964
8.1007.2.2 pwrDenialTime . . . 964
8.1007.2.3 revPolarity . . . 964
8.1007.2.4 toggleMode . . . 964
8.1008 voice_newPwdData Struct Reference . . . 964
8.1008.1 Detailed Description . . . 964
8.1008.2 Field Documentation . . . 965
8.1008.2.1 newPwd . . . 965
8.1008.2.2 newPwdAgain . . . 965
8.1009 voice_NSSAudioCtrl Struct Reference . . . 965
8.1009.1 Detailed Description . . . 965
8.1009.2 Field Documentation . . . 965
8.1009.2.1 downLink . . . 965
8.1009.2.2 upLink . . . 965
8.1010 voice_peerNumberInfo Struct Reference . . . 965
8.1010.1 Detailed Description . . . 965
8.1010.2 Field Documentation . . . 966
8.1010.2.1 callID . . . 967
8.1010.2.2 number . . . 967
8.1010.2.3 numLen . . . 967
8.1010.2.4 numPI . . . 967
8.1010.2.5 numPlan . . . 967
8.1010.2.6 numSI . . . 967
8.1010.2.7 numType . . . 967
8.1011 voice_prefVoiceSO Struct Reference . . . 967
8.1011.1 Detailed Description . . . 967
8.1011.2 Field Documentation . . . 969
8.1011.2.1 evrcCapability . . . 969
8.1011.2.2 homeOrigVoiceSO . . . 969
8.1011.2.3 homePageVoiceSO . . . 969
8.1011.2.4 namID . . . 969
8.1011.2.5 roamOrigVoiceSO . . . 969
8.1012 voice_redirNumInfo Struct Reference . . . 969
8.1012.1 Detailed Description . . . 969
8.1012.2 Field Documentation . . . 970
8.1012.2.1 number . . . 970
8.1012.2.2 numLen . . . 970
8.1012.2.3 numPlan . . . 970
8.1012.2.4 numType . . . 970

8.1012.2.5 PI . . . 970
8.1012.2.6 reason . . . 970
8.1012.2.7 SI . . . 970
8.1013 voice_remotePartyName Struct Reference . . . 971
8.1013.1 Detailed Description . . . 971
8.1013.2 Field Documentation . . . 971
8.1013.2.1 callerName . . . 971
8.1013.2.2 codingScheme . . . 971
8.1013.2.3 nameLen . . . 971
8.1013.2.4 namePI . . . 971
8.1014 voice_remotePartyNum Struct Reference . . . 971
8.1014.1 Detailed Description . . . 972
8.1014.2 Field Documentation . . . 972
8.1014.2.1 numLen . . . 972
8.1014.2.2 presentationInd . . . 972
8.1014.2.3 remPartyNumber . . . 972
8.1015 voice_roamTimer Struct Reference . . . 972
8.1015.1 Detailed Description . . . 972
8.1015.2 Field Documentation . . . 973
8.1015.2.1 namID . . . 973
8.1015.2.2 roamTimerValue . . . 973
8.1016 voice_signalInfo Struct Reference . . . 973
8.1016.1 Detailed Description . . . 973
8.1016.2 Field Documentation . . . 973
8.1016.2.1 alertPitch . . . 973
8.1016.2.2 signal . . . 973
8.1016.2.3 signalType . . . 973
8.1017 voice_SUPSInfo Struct Reference . . . 974
8.1017.1 Detailed Description . . . 974
8.1017.2 Field Documentation . . . 974
8.1017.2.1 isModByCC . . . 974
8.1017.2.2 svcType . . . 974
8.1018 voice_USSDNotificationNetworkInfo Struct Reference . . . 974
8.1018.1 Detailed Description . . . 974
8.1018.2 Field Documentation . . . 975
8.1018.2.1 networkInfo . . . 975
8.1018.2.2 lvPresent . . . 975
8.1019 voice_USSInfo Struct Reference . . . 975
8.1019.1 Detailed Description . . . 975
8.1019.2 Field Documentation . . . 975

8.1019.2.1ussData . . . 975
8.1019.2.2ussDCS . . . 975
8.1019.2.3ussLen . . . 975
8.1020voice_UUSInfo Struct Reference . . . 975
8.1020.1Detailed Description . . . 976
8.1020.2Field Documentation . . . 976
8.1020.2.1UUSData . . . 976
8.1020.2.2UUSDatalen . . . 976
8.1020.2.3UUSDcs . . . 976
8.1020.2.4UUSType . . . 976
8.1021wds_channelRate Struct Reference . . . 977
8.1021.1Detailed Description . . . 977
8.1021.2Field Documentation . . . 977
8.1021.2.1CurrChanRxRate . . . 977
8.1021.2.2CurrChanTxRate . . . 977
8.1022wds_ChannelRateTlv Struct Reference . . . 977
8.1022.1Detailed Description . . . 977
8.1022.2Field Documentation . . . 977
8.1022.2.1ChannelRate . . . 977
8.1022.2.2TlvPresent . . . 977
8.1023wds_ConnStatusTlv Struct Reference . . . 978
8.1023.1Detailed Description . . . 978
8.1023.2Field Documentation . . . 978
8.1023.2.1MDMConnStatus . . . 978
8.1023.2.2TlvPresent . . . 978
8.1024wds_currNetworkInfo Struct Reference . . . 978
8.1024.1Detailed Description . . . 978
8.1024.2Field Documentation . . . 979
8.1024.2.1NetworkType . . . 979
8.1024.2.2RATMask . . . 979
8.1024.2.3SOMask . . . 979
8.1025wds_DataBearTechTlv Struct Reference . . . 979
8.1025.1Detailed Description . . . 979
8.1025.2Field Documentation . . . 979
8.1025.2.1DataBearerTech . . . 979
8.1025.2.2TlvPresent . . . 980
8.1026wds_DataULongLongTlv Struct Reference . . . 980
8.1026.1Detailed Description . . . 980
8.1026.2Field Documentation . . . 980
8.1026.2.1TlvPresent . . . 980

8.1026.2.2 ullData . . . 980
8.1027 wds_DataULongTlv Struct Reference . . . 980
8.1027.1 Detailed Description . . . 980
8.1027.2 Field Documentation . . . 980
8.1027.2.1 TlvPresent . . . 980
8.1027.2.2 ulData . . . 980
8.1028 wds_DHCPLeaseOptTlv Struct Reference . . . 980
8.1028.1 Detailed Description . . . 981
8.1028.2 Field Documentation . . . 981
8.1028.2.1 numOpt . . . 981
8.1028.2.2 optList . . . 981
8.1028.2.3 optListData . . . 981
8.1028.2.4 TlvPresent . . . 981
8.1029 wds_DHCPLeaseStateTlv Struct Reference . . . 981
8.1029.1 Detailed Description . . . 981
8.1029.2 Field Documentation . . . 981
8.1029.2.1 leaseState . . . 981
8.1029.2.2 TlvPresent . . . 982
8.1030 wds_DHCPOpt Struct Reference . . . 982
8.1030.1 Detailed Description . . . 982
8.1030.2 Field Documentation . . . 982
8.1030.2.1 optCode . . . 982
8.1030.2.2 optValLen . . . 982
8.1030.2.3 pOptVal . . . 982
8.1031 wds_DHCPProfileIdTlv Struct Reference . . . 982
8.1031.1 Detailed Description . . . 982
8.1031.2 Field Documentation . . . 983
8.1031.2.1 profileId . . . 983
8.1031.2.2 profileType . . . 983
8.1031.2.3 TlvPresent . . . 983
8.1032 wds_DHCPv4HWConfig Struct Reference . . . 983
8.1032.1 Detailed Description . . . 983
8.1032.2 Field Documentation . . . 983
8.1032.2.1 chaddr . . . 983
8.1032.2.2 chaddrLen . . . 983
8.1032.2.3 hwType . . . 983
8.1033 wds_DHCPv4Option Struct Reference . . . 983
8.1033.1 Detailed Description . . . 984
8.1033.2 Field Documentation . . . 984
8.1033.2.1 optCode . . . 984

8.1033.2.2 optVal . . . 984
8.1033.2.3 optValLen . . . 984
8.1034 wds_DHCPv4OptionList Struct Reference . . . 984
8.1034.1 Detailed Description . . . 984
8.1034.2 Field Documentation . . . 984
8.1034.2.1 numOpt . . . 984
8.1034.2.2 pOptList . . . 984
8.1035 wds_DHCPv4ProfileId Struct Reference . . . 985
8.1035.1 Detailed Description . . . 985
8.1035.2 Field Documentation . . . 985
8.1035.2.1 profileId . . . 985
8.1035.2.2 profileType . . . 985
8.1036 wds_Domain Struct Reference . . . 985
8.1036.1 Detailed Description . . . 985
8.1036.2 Field Documentation . . . 985
8.1036.2.1 domainLen . . . 985
8.1036.2.2 domainName . . . 985
8.1037 wds_DomainNameList Struct Reference . . . 986
8.1037.1 Detailed Description . . . 986
8.1037.2 Field Documentation . . . 986
8.1037.2.1 domain . . . 986
8.1037.2.2 numInstances . . . 986
8.1038 wds_DormStatTlv Struct Reference . . . 986
8.1038.1 Detailed Description . . . 986
8.1038.2 Field Documentation . . . 986
8.1038.2.1 DormancyStat . . . 987
8.1038.2.2 TlvPresent . . . 987
8.1039 wds_GPRSQoS Struct Reference . . . 987
8.1039.1 Detailed Description . . . 987
8.1039.2 Field Documentation . . . 987
8.1039.2.1 delayClass . . . 987
8.1039.2.2 meanThroughputClass . . . 987
8.1039.2.3 peakThroughputClass . . . 987
8.1039.2.4 precedenceClass . . . 987
8.1039.2.5 reliabilityClass . . . 987
8.1040 wds_IPv4AdTlv Struct Reference . . . 987
8.1040.1 Detailed Description . . . 988
8.1040.2 Field Documentation . . . 988
8.1040.2.1 IPv4Addr . . . 988
8.1040.2.2 TlvPresent . . . 988

8.1041 wds_IPV6AddressInfo Struct Reference . . . 988
8.1041.1 Detailed Description . . . 988
8.1041.2 Field Documentation . . . 988
8.1041.2.1 IPAddressV6 . . . 988
8.1041.2.2 IPV6PrefixLen . . . 988
8.1042 wds_IPV6GWAddressInfo Struct Reference . . . 988
8.1042.1 Detailed Description . . . 989
8.1042.2 Field Documentation . . . 989
8.1042.2.1 gwAddressV6 . . . 989
8.1042.2.2 gwV6PrefixLen . . . 989
8.1043 wds_LastMdmCallEndRsnTlv Struct Reference . . . 989
8.1043.1 Detailed Description . . . 989
8.1043.2 Field Documentation . . . 989
8.1043.2.1 CallEndReason . . . 989
8.1043.2.2 TlvPresent . . . 989
8.1044 wds_PCSCFFQDNAddress Struct Reference . . . 990
8.1044.1 Detailed Description . . . 990
8.1044.2 Field Documentation . . . 990
8.1044.2.1 fqdnAddr . . . 990
8.1044.2.2 fqdnLen . . . 990
8.1045 wds_PCSCFFQDNAddressList Struct Reference . . . 990
8.1045.1 Detailed Description . . . 990
8.1045.2 Field Documentation . . . 990
8.1045.2.1 numInstances . . . 990
8.1045.2.2 pcsfFQDNAddress . . . 991
8.1046 wds_PCSCFIPv4ServerAddressList Struct Reference . . . 991
8.1046.1 Detailed Description . . . 991
8.1046.2 Field Documentation . . . 991
8.1046.2.1 numInstances . . . 991
8.1046.2.2 pscsfIPv4Addr . . . 991
8.1047 wds_ProfileIdentifier Struct Reference . . . 991
8.1047.1 Detailed Description . . . 991
8.1047.2 Field Documentation . . . 992
8.1047.2.1 profileIndex . . . 992
8.1047.2.2 profileType . . . 992
8.1048 wds_profileInfo Union Reference . . . 992
8.1048.1 Detailed Description . . . 992
8.1048.2 Field Documentation . . . 992
8.1048.2.1 SlqsProfile3GPP . . . 992
8.1048.2.2 SlqsProfile3GPP2 . . . 992

8.1049 wds_RXBytesOKTlv Struct Reference . . . 992
8.1049.1 Detailed Description . . . 992
8.1049.2 Field Documentation . . . 992
8.1049.2.1 RxByteOKCnt . . . 993
8.1049.2.2 TlvPresent . . . 993
8.1050 wds_transferStatInd Struct Reference . . . 993
8.1050.1 Detailed Description . . . 993
8.1050.2 Field Documentation . . . 993
8.1050.2.1 StatsMask . . . 993
8.1050.2.2 StatsPeriod . . . 993
8.1051 wds_TrStatInd Struct Reference . . . 993
8.1051.1 Detailed Description . . . 993
8.1051.2 Field Documentation . . . 994
8.1051.2.1 statsMask . . . 994
8.1051.2.2 statsPeriod . . . 994
8.1052 wds_TXBytesOKTlv Struct Reference . . . 994
8.1052.1 Detailed Description . . . 994
8.1052.2 Field Documentation . . . 994
8.1052.2.1 TlvPresent . . . 994
8.1052.2.2 TxByteOKCnt . . . 994
8.1053 wds_UMTSMinQoS Struct Reference . . . 994
8.1053.1 Detailed Description . . . 995
8.1053.2 Field Documentation . . . 996
8.1053.2.1 deliveryErrSDU . . . 996
8.1053.2.2 grntDownlinkBitrate . . . 997
8.1053.2.3 grntUplinkBitrate . . . 997
8.1053.2.4 maxDownlinkBitrate . . . 997
8.1053.2.5 maxSDUSize . . . 997
8.1053.2.6 maxUplinkBitrate . . . 997
8.1053.2.7 qosDeliveryOrder . . . 997
8.1053.2.8 resBerRatio . . . 997
8.1053.2.9 sduErrorRatio . . . 997
8.1053.2.10 trafficClass . . . 997
8.1053.2.11 trafficPriority . . . 997
8.1053.2.12 transferDelay . . . 997
8.1054 wdsDhcpv4HwConfig Struct Reference . . . 997
8.1054.1 Detailed Description . . . 997
8.1054.2 Field Documentation . . . 997
8.1054.2.1 chaddr . . . 997
8.1054.2.2 chaddrLen . . . 997

8.1054.2.3 hwType . . . 997
8.1055 wdsDhcpv4Option Struct Reference . . . 998
8.1055.1 Detailed Description . . . 998
8.1055.2 Field Documentation . . . 998
8.1055.2.1 optCode . . . 998
8.1055.2.2 optVal . . . 998
8.1055.2.3 optValLen . . . 998
8.1056 wdsDhcpv4OptionList Struct Reference . . . 998
8.1056.1 Detailed Description . . . 998
8.1056.2 Field Documentation . . . 998
8.1056.2.1 numOpt . . . 999
8.1056.2.2 pOptList . . . 999
8.1057 wdsDhcpv4ProfileId Struct Reference . . . 999
8.1057.1 Detailed Description . . . 999
8.1057.2 Field Documentation . . . 999
8.1057.2.1 profileId . . . 999
8.1057.2.2 profileType . . . 999

**9 File Documentation** **1001**
9.1 apdoxypages.c File Reference . . . 1001
9.1.1 Detailed Description . . . 1001
9.2 audio.h File Reference . . . 1001
9.2.1 Function Documentation . . . 1002
9.2.1.1 pack_audio_SLQSGetAudioPathConfig . . . 1002
9.2.1.2 pack_audio_SLQSGetAudioProfile . . . 1002
9.2.1.3 pack_audio_SLQSGetAudioVolTLBConfig . . . 1003
9.2.1.4 pack_audio_SLQSSetAudioPathConfig . . . 1003
9.2.1.5 pack_audio_SLQSSetAudioProfile . . . 1004
9.2.1.6 pack_audio_SLQSSetAudioVolTLBConfig . . . 1004
9.2.1.7 unpack_audio_SLQSGetAudioPathConfig . . . 1005
9.2.1.8 unpack_audio_SLQSGetAudioProfile . . . 1005
9.2.1.9 unpack_audio_SLQSGetAudioVolTLBConfig . . . 1005
9.2.1.10 unpack_audio_SLQSSetAudioPathConfig . . . 1006
9.2.1.11 unpack_audio_SLQSSetAudioProfile . . . 1006
9.2.1.12 unpack_audio_SLQSSetAudioVolTLBConfig . . . 1006
9.3 cat.h File Reference . . . 1007
9.3.1 Macro Definition Documentation . . . 1008
9.3.1.1 CAN_COMMON_EVENT_TLV_NUMBER . . . 1008
9.3.1.2 CAT_EVENT_DATA_MAX_LENGTH . . . 1008
9.3.2 Typedef Documentation . . . 1008

9.3.2.1 unpack_cat_CATSendEnvelopeCommand_t . . . 1008
9.3.2.2 unpack_cat_CATSendTerminalResponse_t . . . 1008
9.3.3 Function Documentation . . . 1008
9.3.3.1 pack_cat_CATSendEnvelopeCommand . . . 1008
9.3.3.2 pack_cat_CATSendTerminalResponse . . . 1008
9.3.3.3 pack_cat_SetCATEventCallback . . . 1009
9.3.3.4 unpack_cat_CATSendEnvelopeCommand . . . 1009
9.3.3.5 unpack_cat_CATSendTerminalResponse . . . 1009
9.3.3.6 unpack_cat_SetCATEventCallback . . . 1010
9.3.3.7 unpack_cat_SetCatEventCallback_ind . . . 1010
9.4 common.h File Reference . . . 1010
9.4.1 Macro Definition Documentation . . . 1012
9.4.1.1 DEAULT_LOC_TIMEOUT_IN_SEC . . . 1012
9.4.1.2 MINREQBKLEN . . . 1012
9.4.1.3 MSGID_AND_LEN . . . 1012
9.4.1.4 MSGID_DONT_CARE . . . 1012
9.4.1.5 SDK_VALIDATE_INPUT_PACK_PARAM . . . 1012
9.4.1.6 SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID . . . 1012
9.4.1.7 SDK_VALIDATE_INPUT_UNPACK_PARAM . . . 1013
9.4.1.8 SDU_HDR_LEN . . . 1013
9.4.1.9 UNUSEDPARAM . . . 1013
9.4.2 Typedef Documentation . . . 1013
9.4.2.1 logger . . . 1013
9.4.3 Enumeration Type Documentation . . . 1013
9.4.3.1 eLOG_LEVEL . . . 1013
9.4.3.2 eQMI_SVC . . . 1013
9.4.3.3 eTimeout . . . 1014
9.4.3.4 msgtype . . . 1014
9.4.4 Function Documentation . . . 1015
9.4.4.1 fill_pack_ctx . . . 1015
9.4.4.2 fill_sdu_hdr . . . 1015
9.4.4.3 get_version . . . 1015
9.4.4.4 helper_get_error_code . . . 1015
9.4.4.5 helper_get_error_reason . . . 1015
9.4.4.6 helper_get_req_str . . . 1015
9.4.4.7 helper_get_resp_ctx . . . 1015
9.4.4.8 helper_get_xid . . . 1015
9.4.4.9 helper_set_log_func . . . 1015
9.4.4.10 helper_set_log_lvl . . . 1015
9.4.4.11 liteqmi_GetVersion . . . 1016

9.4.4.12 liteqmi_helper_decode7bitAsciiEncString . . . 1016
9.4.4.13 liteqmi_log . . . 1016
9.4.4.14 unpack_result_code_only . . . 1016
9.4.5 Variable Documentation . . . 1016
9.4.5.1 glog . . . 1016
9.4.5.2 gloglvl . . . 1016
9.5 dms.h File Reference . . . 1016
9.5.1 Macro Definition Documentation . . . 1023
9.5.1.1 ACT_CODE_MAX_SIZE . . . 1023
9.5.1.2 CK_MAX_SIZE . . . 1023
9.5.1.3 DMS_IMGDETAILS_LEN . . . 1023
9.5.1.4 DMS_MAX_CUST_ID_LEN . . . 1023
9.5.1.5 DMS_MAX_CUST_VALUE_LEN . . . 1023
9.5.1.6 DMS_MAX_FWUPDATE_LOG_STR_SZ . . . 1023
9.5.1.7 DMS_MAX_FWUPDATE_REF_STR_SZ . . . 1023
9.5.1.8 DMS_MAX_RADIO_IFCS_SIZE . . . 1023
9.5.1.9 DMS_PM_FACTORY . . . 1024
9.5.1.10 DMS_PM_LOW . . . 1024
9.5.1.11 DMS_PM_OFFLINE . . . 1024
9.5.1.12 DMS_PM_ONLINE . . . 1024
9.5.1.13 DMS_PM_PERSISTENT_LOW . . . 1024
9.5.1.14 DMS_PM_RESET . . . 1024
9.5.1.15 DMS_PM_SHUT_DOWN . . . 1024
9.5.1.16 DMS_SET_REG_IND_DISABLE . . . 1024
9.5.1.17 DMS_SET_REG_IND_ENABLE . . . 1024
9.5.1.18 DMS_SET_REG_IND_NO_CHANGE . . . 1024
9.5.1.19 DMS_SET_REPORT_DISABLE . . . 1024
9.5.1.20 DMS_SET_REPORT_ENABLE . . . 1024
9.5.1.21 DMS_SLQSFWINFO_APPVERSION_SZ . . . 1024
9.5.1.22 DMS_SLQSFWINFO_BOOTVERSION_SZ . . . 1024
9.5.1.23 DMS_SLQSFWINFO_CARRIER_SZ . . . 1024
9.5.1.24 DMS_SLQSFWINFO_CUR_CARR_NAME . . . 1024
9.5.1.25 DMS_SLQSFWINFO_CUR_CARR_REV . . . 1024
9.5.1.26 DMS_SLQSFWINFO_MODELID_SZ . . . 1024
9.5.1.27 DMS_SLQSFWINFO_PACKAGEID_SZ . . . 1024
9.5.1.28 DMS_SLQSFWINFO_PRIVERSION_SZ . . . 1024
9.5.1.29 DMS_SLQSFWINFO_SKU_SZ . . . 1024
9.5.1.30 DMS_SWI_SET_IND_DISABLE . . . 1024
9.5.1.31 DMS_SWI_SET_IND_ENABLE . . . 1024
9.5.1.32 DMS_UINT8_MAX_STRING_SZ . . . 1024

9.5.1.33 DMS_VALID_FSN_LEN . . . 1024
9.5.1.34 ERI_DATA_MAX_SIZE . . . 1024
9.5.1.35 MAX_BUILD_ID_LEN . . . 1024
9.5.1.36 MEID_MAX_SIZE . . . 1024
9.5.1.37 SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . 1025
9.5.1.38 SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . 1025
9.5.1.39 SPC_SIZE . . . 1025
9.5.1.40 UNIQUE_ID_LEN . . . 1025
9.5.2 Function Documentation . . . 1025
9.5.2.1 pack_dms_ActivateAutomatic . . . 1025
9.5.2.2 pack_dms_GetActivationState . . . 1025
9.5.2.3 pack_dms_GetBandCapability . . . 1025
9.5.2.4 pack_dms_GetCrashAction . . . 1026
9.5.2.5 pack_dms_GetCustFeature . . . 1026
9.5.2.6 pack_dms_GetCustFeaturesV2 . . . 1027
9.5.2.7 pack_dms_GetDeviceCap . . . 1027
9.5.2.8 pack_dms_GetDeviceCapabilities . . . 1027
9.5.2.9 pack_dms_GetDeviceHardwareRev . . . 1028
9.5.2.10 pack_dms_GetDeviceMfr . . . 1028
9.5.2.11 pack_dms_GetDeviceSerialNumbers . . . 1028
9.5.2.12 pack_dms_GetFirmwareInfo . . . 1029
9.5.2.13 pack_dms_GetFirmwareRevision . . . 1029
9.5.2.14 pack_dms_GetFirmwareRevisions . . . 1029
9.5.2.15 pack_dms_GetFSN . . . 1030
9.5.2.16 pack_dms_GetHardwareRevision . . . 1030
9.5.2.17 pack_dms_GetIMSI . . . 1030
9.5.2.18 pack_dms_GetManufacturer . . . 1031
9.5.2.19 pack_dms_GetModelID . . . 1031
9.5.2.20 pack_dms_GetNetworkTime . . . 1031
9.5.2.21 pack_dms_GetOfflineReason . . . 1032
9.5.2.22 pack_dms_GetPower . . . 1032
9.5.2.23 pack_dms_GetPRLVersion . . . 1032
9.5.2.24 pack_dms_GetSerialNumbers . . . 1033
9.5.2.25 pack_dms_GetUSBComp . . . 1033
9.5.2.26 pack_dms_GetVoiceNumber . . . 1034
9.5.2.27 pack_dms_ResetToFactoryDefaults . . . 1034
9.5.2.28 pack_dms_SetActivationStatusCallback . . . 1034
9.5.2.29 pack_dms_SetCrashAction . . . 1035
9.5.2.30 pack_dms_SetCustFeature . . . 1035
9.5.2.31 pack_dms_SetCustFeaturesV2 . . . 1035

9.5.2.32 pack_dms_SetEventReport . . . 1036
9.5.2.33 pack_dms_SetFirmwarePreference . . . 1036
9.5.2.34 pack_dms_SetIndicationRegister . . . 1036
9.5.2.35 pack_dms_SetPower . . . 1037
9.5.2.36 pack_dms_SetUSBComp . . . 1037
9.5.2.37 pack_dms_SLQSDmsSwiGetPCInfo . . . 1038
9.5.2.38 pack_dms_SLQSDmsSwiGetResetInfo . . . 1038
9.5.2.39 pack_dms_SLQSDmsSwiIndicationRegister . . . 1038
9.5.2.40 pack_dms_SLQSGetBandCapability . . . 1039
9.5.2.41 pack_dms_SLQSGetERIFile . . . 1039
9.5.2.42 pack_dms_SLQSGetPowerSaveModeConfig . . . 1039
9.5.2.43 pack_dms_SLQSSetPowerSaveModeConfig . . . 1040
9.5.2.44 pack_dms_SLQSSwiClearDyingGaspStatistics . . . 1040
9.5.2.45 pack_dms_SLQSSwiGetCrashInfo . . . 1041
9.5.2.46 pack_dms_SLQSSwiGetDyingGaspCfg . . . 1041
9.5.2.47 pack_dms_SLQSSwiGetDyingGaspStatistics . . . 1041
9.5.2.48 pack_dms_SLQSSwiGetFirmwareCurr . . . 1042
9.5.2.49 pack_dms_SLQSSwiGetFwUpdateStatus . . . 1042
9.5.2.50 pack_dms_SLQSSwiGetHostDevInfo . . . 1042
9.5.2.51 pack_dms_SLQSSwiGetOSInfo . . . 1043
9.5.2.52 pack_dms_SLQSSwiGetSerialNoExt . . . 1043
9.5.2.53 pack_dms_SLQSSwiSetDyingGaspCfg . . . 1043
9.5.2.54 pack_dms_SLQSSwiSetHostDevInfo . . . 1044
9.5.2.55 pack_dms_SLQSSwiSetOSInfo . . . 1044
9.5.2.56 pack_dms_SLQSUIMGetState . . . 1044
9.5.2.57 pack_dms_SwiSetEventReport . . . 1045
9.5.2.58 pack_dms_UIMChangePIN . . . 1045
9.5.2.59 pack_dms_UIMGetControlKeyStatus . . . 1045
9.5.2.60 pack_dms_UIMGetICCID . . . 1046
9.5.2.61 pack_dms_UIMGetPINStatus . . . 1046
9.5.2.62 pack_dms_UIMSetControlKeyProtection . . . 1047
9.5.2.63 pack_dms_UIMSetPINProtection . . . 1047
9.5.2.64 pack_dms_UIMUnblockControlKey . . . 1047
9.5.2.65 pack_dms_UIMUnblockPIN . . . 1048
9.5.2.66 pack_dms_UIMVerifyPIN . . . 1048
9.5.2.67 pack_dms_ValidateSPC . . . 1048
9.5.2.68 unpack_dms_ActivateAutomatic . . . 1049
9.5.2.69 unpack_dms_GetActivationState . . . 1049
9.5.2.70 unpack_dms_GetBandCapability . . . 1050
9.5.2.71 unpack_dms_GetCrashAction . . . 1050

9.5.2.72 unpack_dms_GetCustFeature . . . 1050
9.5.2.73 unpack_dms_GetCustFeaturesV2 . . . 1051
9.5.2.74 unpack_dms_GetDeviceCap . . . 1051
9.5.2.75 unpack_dms_GetDeviceCapabilities . . . 1051
9.5.2.76 unpack_dms_GetDeviceHardwareRev . . . 1052
9.5.2.77 unpack_dms_GetDeviceMfr . . . 1052
9.5.2.78 unpack_dms_GetDeviceSerialNumbers . . . 1052
9.5.2.79 unpack_dms_GetFirmwareInfo . . . 1053
9.5.2.80 unpack_dms_GetFirmwareRevision . . . 1053
9.5.2.81 unpack_dms_GetFirmwareRevisions . . . 1053
9.5.2.82 unpack_dms_GetFSN . . . 1054
9.5.2.83 unpack_dms_GetHardwareRevision . . . 1054
9.5.2.84 unpack_dms_GetIMSI . . . 1054
9.5.2.85 unpack_dms_GetManufacturer . . . 1055
9.5.2.86 unpack_dms_GetModelID . . . 1055
9.5.2.87 unpack_dms_GetNetworkTime . . . 1055
9.5.2.88 unpack_dms_GetOfflineReason . . . 1056
9.5.2.89 unpack_dms_GetPower . . . 1056
9.5.2.90 unpack_dms_GetPRLVersion . . . 1056
9.5.2.91 unpack_dms_GetSerialNumbers . . . 1057
9.5.2.92 unpack_dms_GetUSBComp . . . 1057
9.5.2.93 unpack_dms_GetVoiceNumber . . . 1057
9.5.2.94 unpack_dms_PSMCfgChange_ind . . . 1058
9.5.2.95 unpack_dms_ResetToFactoryDefaults . . . 1058
9.5.2.96 unpack_dms_SetActivationStatusCallback . . . 1058
9.5.2.97 unpack_dms_SetCrashAction . . . 1059
9.5.2.98 unpack_dms_SetCustFeature . . . 1059
9.5.2.99 unpack_dms_SetCustFeaturesV2 . . . 1059
9.5.2.100 unpack_dms_SetEventReport . . . 1060
9.5.2.101 unpack_dms_SetEventReport_ind . . . 1060
9.5.2.102 unpack_dms_SetFirmwarePreference . . . 1061
9.5.2.103 unpack_dms_SetIndicationRegister . . . 1061
9.5.2.104 unpack_dms_SetPower . . . 1061
9.5.2.105 unpack_dms_SetUSBComp . . . 1062
9.5.2.106 unpack_dms_SLQSDmsSwiGetPCInfo . . . 1062
9.5.2.107 unpack_dms_SLQSDmsSwiGetResetInfo . . . 1062
9.5.2.108 unpack_dms_SLQSDmsSwiGetResetInfo_Ind . . . 1063
9.5.2.109 unpack_dms_SLQSDmsSwiIndicationRegister . . . 1063
9.5.2.110 unpack_dms_SLQSGetBandCapability . . . 1063
9.5.2.111 unpack_dms_SLQSGetERIFile . . . 1064

9.5.2.112 unpack_dms_SLQSGetPowerSaveModeConfig . . . 1064
9.5.2.113 unpack_dms_SLQSSetPowerSaveModeConfig . . . 1065
9.5.2.114 unpack_dms_SLQSSwiClearDyingGaspStatistics . . . 1065
9.5.2.115 unpack_dms_SLQSSwiGetCrashInfo . . . 1065
9.5.2.116 unpack_dms_SLQSSwiGetDyingGaspCfg . . . 1066
9.5.2.117 unpack_dms_SLQSSwiGetDyingGaspStatistics . . . 1066
9.5.2.118 unpack_dms_SLQSSwiGetFirmwareCurr . . . 1066
9.5.2.119 unpack_dms_SLQSSwiGetFwUpdateStatus . . . 1067
9.5.2.120 unpack_dms_SLQSSwiGetHostDevInfo . . . 1067
9.5.2.121 unpack_dms_SLQSSwiGetOSInfo . . . 1067
9.5.2.122 unpack_dms_SLQSSwiGetSerialNoExt . . . 1068
9.5.2.123 unpack_dms_SLQSSwiSetDyingGaspCfg . . . 1068
9.5.2.124 unpack_dms_SLQSSwiSetHostDevInfo . . . 1068
9.5.2.125 unpack_dms_SLQSSwiSetOSInfo . . . 1069
9.5.2.126 unpack_dms_SLQSUIMGetState . . . 1069
9.5.2.127 unpack_dms_SwiEventReportCallBack_ind . . . 1069
9.5.2.128 unpack_dms_SwiSetEventReport . . . 1070
9.5.2.129 unpack_dms_UIMChangePIN . . . 1070
9.5.2.130 unpack_dms_UIMGetControlKeyStatus . . . 1071
9.5.2.131 unpack_dms_UIMGetICCID . . . 1071
9.5.2.132 unpack_dms_UIMGetPINStatus . . . 1071
9.5.2.133 unpack_dms_UIMSetControlKeyProtection . . . 1072
9.5.2.134 unpack_dms_UIMSetPINProtection . . . 1072
9.5.2.135 unpack_dms_UIMUnblockControlKey . . . 1072
9.5.2.136 unpack_dms_UIMUnblockPIN . . . 1073
9.5.2.137 unpack_dms_UIMVerifyPIN . . . 1073
9.5.2.138 unpack_dms_ValidateSPC . . . 1073
9.6 fms.h File Reference . . . 1074
9.6.1 Macro Definition Documentation . . . 1075
9.6.1.1 FMS_FW_PRI_BUILD_MATCH_LEN . . . 1075
9.6.1.2 FMS_GOBI_LISTENTRIES_MAX . . . 1075
9.6.1.3 FMS_GOBI_MBN_BUILD_ID_STR_LEN . . . 1075
9.6.1.4 FMS_GOBI_MBN_IMG_ID_STR_LEN . . . 1075
9.6.1.5 FMS_IMAGE_ID_BUILD_ID_LEN . . . 1075
9.6.1.6 FMS_IMAGE_ID_IMG_ID_LEN . . . 1075
9.6.1.7 FMS_IMAGE_ID_MAX_ENTRIES . . . 1075
9.6.1.8 FMS_IMAGE_ID_PRI_IMGTYPE . . . 1075
9.6.1.9 FMS_MAX_IMAGE_ID_ELEMENT . . . 1075
9.6.1.10 FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE . . . 1075
9.6.2 Function Documentation . . . 1075

9.6.2.1 GetValidFwPriCombinations . . . 1075
9.6.2.2 pack_fms_GetImagesPreference . . . 1075
9.6.2.3 pack_fms_GetStoredImages . . . 1076
9.6.2.4 pack_fms_SetImagesPreference . . . 1076
9.6.2.5 unpack_fms_GetImagesPreference . . . 1076
9.6.2.6 unpack_fms_GetStoredImages . . . 1076
9.6.2.7 unpack_fms_SetImagesPreference . . . 1077
9.7 ims.h File Reference . . . 1077
9.7.1 Macro Definition Documentation . . . 1079
9.7.1.1 MAX_NAME_LEN . . . 1079
9.7.2 Typedef Documentation . . . 1079
9.7.2.1 unpack_ims_SLQSImsConfigIndicationRegister_t . . . 1079
9.7.3 Function Documentation . . . 1079
9.7.3.1 pack_ims_SLQSGetIMSSMSConfig . . . 1079
9.7.3.2 pack_ims_SLQSGetIMSUserConfig . . . 1079
9.7.3.3 pack_ims_SLQSGetIMSVoIPConfig . . . 1080
9.7.3.4 pack_ims_SLQSGetRegMgrConfig . . . 1080
9.7.3.5 pack_ims_SLQSGetSIPConfig . . . 1081
9.7.3.6 pack_ims_SLQSImsConfigIndicationRegister . . . 1081
9.7.3.7 pack_ims_SLQSSetIMSSMSConfig . . . 1081
9.7.3.8 pack_ims_SLQSSetIMSUserConfig . . . 1082
9.7.3.9 pack_ims_SLQSSetIMSVoIPConfig . . . 1082
9.7.3.10 pack_ims_SLQSSetRegMgrConfig . . . 1083
9.7.3.11 pack_ims_SLQSSetSIPConfig . . . 1083
9.7.3.12 unpack_ims_SLQSGetIMSSMSConfig . . . 1084
9.7.3.13 unpack_ims_SLQSGetIMSUserConfig . . . 1084
9.7.3.14 unpack_ims_SLQSGetIMSVoIPConfig . . . 1084
9.7.3.15 unpack_ims_SLQSGetRegMgrConfig . . . 1085
9.7.3.16 unpack_ims_SLQSGetSIPConfig . . . 1085
9.7.3.17 unpack_ims_SLQSImsConfigIndicationRegister . . . 1085
9.7.3.18 unpack_ims_SLQSRegMgrCfgCallBack_ind . . . 1086
9.7.3.19 unpack_ims_SLQSSetIMSSMSConfig . . . 1086
9.7.3.20 unpack_ims_SLQSSetIMSUserConfig . . . 1086
9.7.3.21 unpack_ims_SLQSSetIMSVoIPConfig . . . 1087
9.7.3.22 unpack_ims_SLQSSetRegMgrConfig . . . 1087
9.7.3.23 unpack_ims_SLQSSetSIPConfig . . . 1087
9.7.3.24 unpack_ims_SLQSSIPCfgCallBack_ind . . . 1088
9.7.3.25 unpack_ims_SLQSSMSCfgCallBack_ind . . . 1088
9.7.3.26 unpack_ims_SLQSUserCfgCallBack_ind . . . 1088
9.7.3.27 unpack_ims_SLQSVoIPCfgCallBack_ind . . . 1089

9.8 imsa.h File Reference . . . 1089
9.8.1 Macro Definition Documentation . . . 1090
9.8.1.1 MAX_ERROR_CODE_LEN . . . 1090
9.8.2 Typedef Documentation . . . 1090
9.8.2.1 unpack_imsa_SLQSRegisterIMSAIndication_t . . . 1090
9.8.3 Function Documentation . . . 1090
9.8.3.1 pack_imsa_SLQSGetIMSARegStatus . . . 1090
9.8.3.2 pack_imsa_SLQSGetIMSAServiceStatus . . . 1091
9.8.3.3 pack_imsa_SLQSRegisterIMSAIndication . . . 1091
9.8.3.4 unpack_imsa_SLQSGetIMSARegStatus . . . 1092
9.8.3.5 unpack_imsa_SLQSGetIMSAServiceStatus . . . 1092
9.8.3.6 unpack_imsa_SLQSImsaPdpStatusCallBack_ind . . . 1093
9.8.3.7 unpack_imsa_SLQSImsaRatStatusCallBack_ind . . . 1093
9.8.3.8 unpack_imsa_SLQSImsaRegStatusCallBack_ind . . . 1093
9.8.3.9 unpack_imsa_SLQSImsaSvcStatusCallBack_ind . . . 1094
9.8.3.10 unpack_imsa_SLQSRegisterIMSAIndication . . . 1094
9.9 lite-fw.h File Reference . . . 1094
9.9.1 Detailed Description . . . 1096
9.9.2 Macro Definition Documentation . . . 1097
9.9.2.1 FIRMWARE_INFO_STRING_SIZE . . . 1097
9.9.2.2 IMG_MASK_CLEAR . . . 1097
9.9.2.3 IMG_MASK_GENERIC . . . 1097
9.9.2.4 IMG_MASK_MDM . . . 1097
9.9.2.5 IMG_MASK_PRI . . . 1097
9.9.2.6 libSDP_BuildImagesPreferenceRequest . . . 1097
9.9.2.7 libSDP_CalculateImageMask . . . 1097
9.9.2.8 LIBSDP_CARRIER_PACKAGE_SKU . . . 1097
9.9.2.9 libSDP_CheckValidFirmwareInfo . . . 1097
9.9.2.10 libSDP_DownloadFW . . . 1097
9.9.2.11 libSDP_ExtractFirmwareParametersByPath . . . 1097
9.9.2.12 libSDP_getFileType . . . 1097
9.9.2.13 libSDP_GetModelFamily . . . 1097
9.9.2.14 libSDP_GetVersion . . . 1097
9.9.2.15 libsdp_set_log_func . . . 1097
9.9.2.16 libsdp_SetReadBlockSize . . . 1097
9.9.2.17 LIBSDP_SKU_STRING_LENGTH . . . 1097
9.9.2.18 LITEFW_CARRIER_PACKAGE_SKU . . . 1097
9.9.2.19 LITEFW_SKU_STRING_LENGTH . . . 1097
9.9.3 Typedef Documentation . . . 1097
9.9.3.1 libSDP_FirmwareInfo . . . 1097

9.9.3.2 libsdplogger . . . 1097
9.9.3.3 litefw_FirmwareInfo . . . 1097
9.9.3.4 litefwlogger . . . 1098
9.9.4 Enumeration Type Documentation . . . 1098
9.9.4.1 litefw_Fw_Type . . . 1098
9.9.4.2 litefw_fwdwl_error_codes . . . 1098
9.9.4.3 litefw_Models . . . 1099
9.9.4.4 litefw_QDL_FLOW_CONTROLs . . . 1099
9.9.4.5 litefw_QDL_MODEs . . . 1099
9.9.5 Function Documentation . . . 1100
9.9.5.1 litefw_BuildImagesPreferenceRequest . . . 1100
9.9.5.2 litefw_CalculateImageMask . . . 1100
9.9.5.3 litefw_CheckValidFirmwareInfo . . . 1100
9.9.5.4 litefw_DownloadFW . . . 1101
9.9.5.5 litefw_ExtractFirmwareParametersByPath . . . 1101
9.9.5.6 litefw_getFileType . . . 1102
9.9.5.7 litefw_GetModelFamily . . . 1102
9.9.5.8 litefw_GetQTLDownloadMode . . . 1102
9.9.5.9 litefw_GetQTLHWFlowControl . . . 1102
9.9.5.10 litefw_GetVersion . . . 1103
9.9.5.11 litefw_logsenable . . . 1103
9.9.5.12 litefw_set_log_func . . . 1103
9.9.5.13 litefw_SetQTLDownloadMode . . . 1103
9.9.5.14 litefw_SetQTLHWFlowControl . . . 1104
9.9.5.15 litefw_SetReadBlockSize . . . 1104
9.9.5.16 litefw_switch_9x07_to_downloadmode . . . 1104
9.9.5.17 litefw_switch_to_BootHoldMode . . . 1104
9.10 loc.h File Reference . . . 1105
9.10.1 Macro Definition Documentation . . . 1108
9.10.1.1 LOC_UINT8_MAX_STRING_SZ . . . 1108
9.10.1.2 LOCEVENTMASKBATCHFULLNOTIFICATION . . . 1108
9.10.1.3 LOCEVENTMASKENGINESTATE . . . 1108
9.10.1.4 LOCEVENTMASKFIXSESSIONSTATE . . . 1109
9.10.1.5 LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION . . . 1109
9.10.1.6 LOCEVENTMASKGEOFENCEBREACHNOTIFICATION . . . 1109
9.10.1.7 LOCEVENTMASKGEOFENCEGENALERT . . . 1109
9.10.1.8 LOCEVENTMASKGNSSMEASUREMENTREPORT . . . 1109
9.10.1.9 LOCEVENTMASKGNSSSVINFO . . . 1109
9.10.1.10 LOCEVENTMASKINJECTPOSITIONREQ . . . 1109
9.10.1.11 LOCEVENTMASKINJECTPREDICTEDORBITSREQ . . . 1109

9.10.1.12 LOCEVENTMASKINJECTTIMEREQ . . . 1109
9.10.1.13 LOCEVENTMASKINJECTWIFIAPDATAREQ . . . 1109
9.10.1.14 LOCEVENTMASKINVALIDVALUE . . . 1109
9.10.1.15 LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT . . . 1110
9.10.1.16 LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ . . . 1110
9.10.1.17 LOCEVENTMASKMOTIONDATACONTROL . . . 1110
9.10.1.18 LOCEVENTMASKNIGEOFENCENOTIFICATION . . . 1110
9.10.1.19 LOCEVENTMASKNINOTIFYVERIFYREQ . . . 1110
9.10.1.20 LOCEVENTMASKNMEA . . . 1110
9.10.1.21 LOCEVENTMASKPEDOMETERCONTROL . . . 1110
9.10.1.22 LOCEVENTMASKPOSITIONREPORT . . . 1110
9.10.1.23 LOCEVENTMASKSENSORSTREAMINGREADYSTATUS . . . 1110
9.10.1.24 LOCEVENTMASKSETSPISTREAMINGREPORT . . . 1110
9.10.1.25 LOCEVENTMASKTIMESYNCREQ . . . 1111
9.10.1.26 LOCEVENTMASKVEHICLEDATAREADYSTATUS . . . 1111
9.10.1.27 LOCEVENTMASKWIFIREQ . . . 1111
9.10.1.28 MAX_LOC_NMEA_STR_LEN . . . 1111
9.10.1.29 MAX_SENSOR_DATA_LEN . . . 1111
9.10.1.30 MAX_TEMP_DATA_LEN . . . 1111
9.10.2 Typedef Documentation . . . 1111
9.10.2.1 unpack_loc_SLQSLOCGetServer_t . . . 1111
9.10.2.2 unpack_loc_SLQSLOCInjectPosition_t . . . 1111
9.10.2.3 unpack_loc_SLQSLOCInjectSensorData_t . . . 1111
9.10.2.4 unpack_loc_SLQSLOCInjectUTCTime_t . . . 1111
9.10.2.5 unpack_loc_SLQSLOCSetCradleMountConfig_t . . . 1111
9.10.3 Enumeration Type Documentation . . . 1111
9.10.3.1 anonymous enum . . . 1111
9.10.4 Function Documentation . . . 1111
9.10.4.1 pack_loc_DeleteAssistData . . . 1111
9.10.4.2 pack_loc_EventRegister . . . 1112
9.10.4.3 pack_loc_SetExtPowerState . . . 1112
9.10.4.4 pack_loc_SetOperationMode . . . 1113
9.10.4.5 pack_loc_SLQSLOCGetBestAvailPos . . . 1113
9.10.4.6 pack_loc_SLQSLOCGetServer . . . 1113
9.10.4.7 pack_loc_SLQSLOCInjectPosition . . . 1114
9.10.4.8 pack_loc_SLQSLOCInjectSensorData . . . 1114
9.10.4.9 pack_loc_SLQSLOCInjectUTCTime . . . 1115
9.10.4.10 pack_loc_SLQSLOCSetCradleMountConfig . . . 1115
9.10.4.11 pack_loc_SLQSLOCSetServer . . . 1115
9.10.4.12 pack_loc_Start . . . 1116

9.10.4.13 pack_loc_Stop . . . 1116
9.10.4.14 unpack_loc_BestAvailPos_Ind . . . 1117
9.10.4.15 unpack_loc_CradleMountCallback_Ind . . . 1117
9.10.4.16 unpack_loc_DeleteAssistData . . . 1117
9.10.4.17 unpack_loc_DeleteAssistData_Ind . . . 1118
9.10.4.18 unpack_loc_EngineState_Ind . . . 1118
9.10.4.19 unpack_loc_EventNMEA_Ind . . . 1118
9.10.4.20 unpack_loc_EventRegister . . . 1119
9.10.4.21 unpack_loc_EventTimeSyncCallback_Ind . . . 1119
9.10.4.22 unpack_loc_GetServer_Ind . . . 1119
9.10.4.23 unpack_loc_GnssSvInfo_Ind . . . 1120
9.10.4.24 unpack_loc_InjectPositionCallback_Ind . . . 1120
9.10.4.25 unpack_loc_InjectSensorDataCallback_Ind . . . 1120
9.10.4.26 unpack_loc_InjectTimeSyncDataCallback_Ind . . . 1121
9.10.4.27 unpack_loc_InjectUTCTimeCallback_Ind . . . 1121
9.10.4.28 unpack_loc_PositionRpt_Ind . . . 1121
9.10.4.29 unpack_loc_SensorStreamingCallback_Ind . . . 1122
9.10.4.30 unpack_loc_SetExtPowerConfig_Ind . . . 1122
9.10.4.31 unpack_loc_SetExtPowerState . . . 1122
9.10.4.32 unpack_loc_SetOperationMode . . . 1123
9.10.4.33 unpack_loc_SetOperationMode_Ind . . . 1123
9.10.4.34 unpack_loc_SetServer_Ind . . . 1123
9.10.4.35 unpack_loc_SLQSLOCGetBestAvailPos . . . 1124
9.10.4.36 unpack_loc_SLQSLOCGetServer . . . 1124
9.10.4.37 unpack_loc_SLQSLOCInjectPosition . . . 1124
9.10.4.38 unpack_loc_SLQSLOCInjectSensorData . . . 1125
9.10.4.39 unpack_loc_SLQSLOCInjectUTCTime . . . 1125
9.10.4.40 unpack_loc_SLQSLOCSetCradleMountConfig . . . 1125
9.10.4.41 unpack_loc_SLQSLOCSetServer . . . 1126
9.10.4.42 unpack_loc_Start . . . 1126
9.10.4.43 unpack_loc_Stop . . . 1126
9.11 nas.h File Reference . . . 1127
9.11.1 Macro Definition Documentation . . . 1136
9.11.1.1 NAS_IMSI_M_S1_LENGTH . . . 1136
9.11.1.2 NAS_IMSI_M_S2_LENGTH . . . 1136
9.11.1.3 NAS_MAX_DESCRIPTION_LENGTH . . . 1136
9.11.1.4 NAS_MAX_NUM_NETWORKS . . . 1136
9.11.1.5 NAS_MAX_SCC_RX_INFO_INSTANCES . . . 1136
9.11.1.6 NAS_NAM_NAME_LENGTH . . . 1136
9.11.1.7 NAS_OTA_MESSAGE_MAX_BUF_SIZE . . . 1136

9.11.1.8 NAS_PLMN_LENGTH . . . 1136
9.11.1.9 NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST . . . 1137
9.11.2 Typedef Documentation . . . 1137
9.11.2.1 unpack_nas_InitiateDomainAttach_t . . . 1137
9.11.2.2 unpack_nas_SetACCOLC_t . . . 1137
9.11.2.3 unpack_nas_SetCDMANetworkParameters_t . . . 1137
9.11.2.4 unpack_nas_SetLURejectCallback_t . . . 1137
9.11.2.5 unpack_nas_SetRFInfoCallback_t . . . 1137
9.11.2.6 unpack_nas_SLQSConfigSigInfo_t . . . 1137
9.11.2.7 unpack_nas_SLQSInitiateNetworkRegistration_t . . . 1137
9.11.2.8 unpack_nas_SLQSNasConfigSigInfo2_t . . . 1137
9.11.2.9 unpack_nas_SLQSNasIndicationRegisterExt_t . . . 1137
9.11.2.10 unpack_nas_SLQSNASSeteDRXParams_t . . . 1137
9.11.2.11 unpack_nas_SLQSNasSwiIndicationRegister_t . . . 1137
9.11.2.12 unpack_nas_SLQSNASSwiSetChannelLock_t . . . 1137
9.11.2.13 unpack_nas_SLQSSetBandPreference_t . . . 1137
9.11.2.14 unpack_nas_SLQSSetSignalStrengthsCallback_t . . . 1137
9.11.2.15 unpack_nas_SLQSSetSysSelectionPref_t . . . 1137
9.11.2.16 unpack_nas_SLQSSetSysSelectionPrefExt_t . . . 1137
9.11.2.17 unpack_nas_SLQSSwiPSDetach_t . . . 1137
9.11.3 Enumeration Type Documentation . . . 1137
9.11.3.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB . . . 1137
9.11.3.2 LITEQMI_NAS_LTE_CPHY_SCELL_STATE . . . 1137
9.11.3.3 NAS_LTE_CPHY_CA_BW_NRB_LITE . . . 1138
9.11.3.4 NAS_LTE_CPHY_SCELL_STATE_LITE . . . 1138
9.11.4 Function Documentation . . . 1138
9.11.4.1 pack_nas_GetACCOLC . . . 1138
9.11.4.2 pack_nas_GetANAAAAuthenticationStatus . . . 1138
9.11.4.3 pack_nas_GetCDMANetworkParameters . . . 1139
9.11.4.4 pack_nas_GetHomeNetwork . . . 1139
9.11.4.5 pack_nas_GetHomeNetwork3GPP2 . . . 1139
9.11.4.6 pack_nas_GetNetworkPreference . . . 1140
9.11.4.7 pack_nas_GetRFInfo . . . 1140
9.11.4.8 pack_nas_GetServingNetwork . . . 1140
9.11.4.9 pack_nas_GetServingNetworkCapabilities . . . 1141
9.11.4.10 pack_nas_GetSignalStrengths . . . 1141
9.11.4.11 pack_nas_InitiateDomainAttach . . . 1141
9.11.4.12 pack_nas_PerformNetworkScan . . . 1142
9.11.4.13 pack_nas_SetACCOLC . . . 1142
9.11.4.14 pack_nas_SetCDMANetworkParameters . . . 1142

9.11.4.15 pack_nas_SetLURejectCallback . . . 1143
9.11.4.16 pack_nas_SetNetworkPreference . . . 1143
9.11.4.17 pack_nas_SetRFInfoCallback . . . 1143
9.11.4.18 pack_nas_SLQSConfigSigInfo . . . 1144
9.11.4.19 pack_nas_SLQSGetErrorRate . . . 1144
9.11.4.20 pack_nas_SlqsGetLTECphyCAInfo . . . 1144
9.11.4.21 pack_nas_SLQSGetNetworkTime . . . 1145
9.11.4.22 pack_nas_SLQSGetOperatorNameData . . . 1145
9.11.4.23 pack_nas_SLQSGetPLMNName . . . 1145
9.11.4.24 pack_nas_SLQSGetServingSystem . . . 1146
9.11.4.25 pack_nas_SLQSGetSignalStrength . . . 1146
9.11.4.26 pack_nas_SLQSGetSysInfo . . . 1146
9.11.4.27 pack_nas_SLQSGetSysSelectionPref . . . 1147
9.11.4.28 pack_nas_SLQSGetSysSelectionPrefExt . . . 1147
9.11.4.29 pack_nas_SLQSInitiateNetworkRegistration . . . 1147
9.11.4.30 pack_nas_SLQSNasConfigSigInfo2 . . . 1148
9.11.4.31 pack_nas_SLQSNasGet3GPP2Subscription . . . 1148
9.11.4.32 pack_nas_SLQSNasGetCellLocationInfo . . . 1148
9.11.4.33 pack_nas_SLQSNASGeteDRXParams . . . 1149
9.11.4.34 pack_nas_SLQSNasGetHDRColorCode . . . 1149
9.11.4.35 pack_nas_SLQSNasGetRFInfo . . . 1149
9.11.4.36 pack_nas_SLQSNasGetSigInfo . . . 1150
9.11.4.37 pack_nas_SLQSNasGetTxRxInfo . . . 1150
9.11.4.38 pack_nas_SLQSNasIndicationRegisterExt . . . 1151
9.11.4.39 pack_nas_SLQSNASSeteDRXParams . . . 1151
9.11.4.40 pack_nas_SLQSNASSwiGetChannelLock . . . 1151
9.11.4.41 pack_nas_SLQSNasSwiIndicationRegister . . . 1152
9.11.4.42 pack_nas_SLQSNasSwiModemStatus . . . 1152
9.11.4.43 pack_nas_SLQSNASSwiSetChannelLock . . . 1152
9.11.4.44 pack_nas_SLQSSetBandPreference . . . 1153
9.11.4.45 pack_nas_SLQSSetSignalStrengthsCallback . . . 1153
9.11.4.46 pack_nas_SLQSSetSysSelectionPref . . . 1154
9.11.4.47 pack_nas_SLQSSetSysSelectionPrefExt . . . 1154
9.11.4.48 pack_nas_SLQSSwiGetHDRPersonality . . . 1154
9.11.4.49 pack_nas_SLQSSwiGetHDRProtSubtype . . . 1155
9.11.4.50 pack_nas_SLQSSwiGetHRPDStats . . . 1155
9.11.4.51 pack_nas_SLQSSwiGetLteCQI . . . 1155
9.11.4.52 pack_nas_SLQSSwiGetLteSccRxInfo . . . 1156
9.11.4.53 pack_nas_SLQSSwiNetworkDebug . . . 1156
9.11.4.54 pack_nas_SLQSSwiPSDetach . . . 1156

9.11.4.55 unpack_nas_GetACCOLC . . . 1157
9.11.4.56 unpack_nas_GetANAAAAuthenticationStatus . . . 1157
9.11.4.57 unpack_nas_GetCDMANetworkParameters . . . 1157
9.11.4.58 unpack_nas_GetHomeNetwork . . . 1158
9.11.4.59 unpack_nas_GetHomeNetwork3GPP2 . . . 1158
9.11.4.60 unpack_nas_GetNetworkPreference . . . 1158
9.11.4.61 unpack_nas_GetRFInfo . . . 1159
9.11.4.62 unpack_nas_GetServingNetwork . . . 1159
9.11.4.63 unpack_nas_GetServingNetworkCapabilities . . . 1159
9.11.4.64 unpack_nas_GetSignalStrengths . . . 1160
9.11.4.65 unpack_nas_InitiateDomainAttach . . . 1160
9.11.4.66 unpack_nas_PerformNetworkScan . . . 1161
9.11.4.67 unpack_nas_SetACCOLC . . . 1161
9.11.4.68 unpack_nas_SetCDMANetworkParameters . . . 1161
9.11.4.69 unpack_nas_SetDataCapabilitiesCallback_ind . . . 1162
9.11.4.70 unpack_nas_SetEventReportInd . . . 1162
9.11.4.71 unpack_nas_SetLURejectCallback . . . 1162
9.11.4.72 unpack_nas_SetNasLTECphyCaIndCallback_ind . . . 1163
9.11.4.73 unpack_nas_SetNetworkPreference . . . 1163
9.11.4.74 unpack_nas_SetRFInfoCallback . . . 1163
9.11.4.75 unpack_nas_SetRoamingIndicatorCallback_ind . . . 1164
9.11.4.76 unpack_nas_SetServingSystemCallback_ind . . . 1164
9.11.4.77 unpack_nas_SLQSConfigSigInfo . . . 1164
9.11.4.78 unpack_nas_SLQSGetErrorRate . . . 1165
9.11.4.79 unpack_nas_SlqsGetLTECphyCAInfo . . . 1165
9.11.4.80 unpack_nas_SLQSGetNetworkTime . . . 1165
9.11.4.81 unpack_nas_SLQSGetOperatorNameData . . . 1166
9.11.4.82 unpack_nas_SLQSGetPLMNName . . . 1166
9.11.4.83 unpack_nas_SLQSGetServingSystem . . . 1166
9.11.4.84 unpack_nas_SLQSGetSignalStrength . . . 1167
9.11.4.85 unpack_nas_SLQSGetSysInfo . . . 1167
9.11.4.86 unpack_nas_SLQSGetSysSelectionPref . . . 1168
9.11.4.87 unpack_nas_SLQSGetSysSelectionPrefExt . . . 1168
9.11.4.88 unpack_nas_SLQSInitiateNetworkRegistration . . . 1168
9.11.4.89 unpack_nas_SLQSNasConfigSigInfo2 . . . 1169
9.11.4.90 unpack_nas_SLQSNasGet3GPP2Subscription . . . 1169
9.11.4.91 unpack_nas_SLQSNasGetCellLocationInfo . . . 1169
9.11.4.92 unpack_nas_SLQSNASGeteDRXParams . . . 1170
9.11.4.93 unpack_nas_SLQSNasGetHDRColorCode . . . 1170
9.11.4.94 unpack_nas_SLQSNasGetRFInfo . . . 1170

9.11.4.95 unpack_nas_SLQSNasGetSigInfo . . . 1171
9.11.4.96 unpack_nas_SLQSNasGetTxRxInfo . . . 1171
9.11.4.97 unpack_nas_SLQSNasIndicationRegisterExt . . . 1172
9.11.4.98 unpack_nas_SLQSNasNetworkTimeCallBack_ind . . . 1172
9.11.4.99 unpack_nas_SLQSNASSeteDRXParams . . . 1172
9.11.4.100 unpack_nas_SLQSNasSigInfoCallback_ind . . . 1173
9.11.4.101 unpack_nas_SLQSNASSwiGetChannelLock . . . 1173
9.11.4.102 unpack_nas_SLQSNasSwiIndicationRegister . . . 1173
9.11.4.103 unpack_nas_SLQSNasSwiModemStatus . . . 1174
9.11.4.104 unpack_nas_SLQSNasSwiOTAMessageCallback_ind . . . 1174
9.11.4.105 unpack_nas_SLQSNASSwiSetChannelLock . . . 1174
9.11.4.106 unpack_nas_SLQSNasSysInfoCallback_ind . . . 1175
9.11.4.107 unpack_nas_SLQSNasTimerCallback_ind . . . 1175
9.11.4.108 unpack_nas_SLQSSetBandPreference . . . 1175
9.11.4.109 unpack_nas_SLQSSetSignalStrengthsCallback . . . 1176
9.11.4.110 unpack_nas_SLQSSetSysSelectionPref . . . 1176
9.11.4.111 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind . . . 1176
9.11.4.112 unpack_nas_SLQSSetSysSelectionPrefExt . . . 1177
9.11.4.113 unpack_nas_SLQSSwiGetHDRPersonality . . . 1177
9.11.4.114 unpack_nas_SLQSSwiGetHDRProtSubtype . . . 1177
9.11.4.115 unpack_nas_SLQSSwiGetHRPDStats . . . 1178
9.11.4.116 unpack_nas_SLQSSwiGetLteCQI . . . 1178
9.11.4.117 unpack_nas_SLQSSwiGetLteSccRxInfo . . . 1178
9.11.4.118 unpack_nas_SLQSSwiHDRPersonalityCallback_Ind . . . 1179
9.11.4.119 unpack_nas_SLQSSwiNetworkDebug . . . 1179
9.11.4.120 unpack_nas_SLQSSwiPSDetach . . . 1179
9.11.4.121 unpack_nas_SLQSSwiRandIndicatorCallback_Ind . . . 1180
9.11.4.122 unpack_valid_nas_GetCDMANetworkParameters . . . 1180
9.11.4.123 unpack_valid_nas_SLQSGetServingSystem . . . 1180
9.11.4.124 unpack_valid_nas_SLQSGetSignalStrength . . . 1181
9.11.4.125 unpack_valid_nas_SLQSNasGetSigInfo . . . 1182
9.12 pds.h File Reference . . . 1182
9.12.1 Typedef Documentation . . . 1185
9.12.1.1 unpack_pds_ForceXTRADownload_t . . . 1185
9.12.1.2 unpack_pds_PDSInjectTimeReference_t . . . 1185
9.12.1.3 unpack_pds_ResetPDSData_t . . . 1185
9.12.1.4 unpack_pds_SetEventReportCallback_t . . . 1185
9.12.1.5 unpack_pds_SetPDSDefaults_t . . . 1185
9.12.1.6 unpack_pds_SetPDSState_t . . . 1185
9.12.1.7 unpack_pds_SetPortAutomaticTracking_t . . . 1185

9.12.1.8 unpack_pds_SetServiceAutomaticTracking_t . . . 1185
9.12.1.9 unpack_pds_SetXTRAAutomaticDownload_t . . . 1185
9.12.1.10 unpack_pds_SetXTRANetwork_t . . . 1185
9.12.1.11 unpack_pds_SLQSPDSDeterminePosition_t . . . 1185
9.12.1.12 unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t . . . 1185
9.12.1.13 unpack_pds_SLQSPDSInjectPositionData_t . . . 1185
9.12.1.14 unpack_pds_SLQSSetAGPSConfig_t . . . 1185
9.12.1.15 unpack_pds_SLQSSetPositionMethodState_t . . . 1185
9.12.1.16 unpack_pds_StartPDSTrackingSessionExt_t . . . 1185
9.12.1.17 unpack_pds_StopPDSTrackingSession_t . . . 1185
9.12.2 Function Documentation . . . 1185
9.12.2.1 pack_pds_ForceXTRADownload . . . 1186
9.12.2.2 pack_pds_GetPDSDefaults . . . 1186
9.12.2.3 pack_pds_GetPDSState . . . 1186
9.12.2.4 pack_pds_GetPortAutomaticTracking . . . 1187
9.12.2.5 pack_pds_GetServiceAutomaticTracking . . . 1187
9.12.2.6 pack_pds_GetXTRAAutomaticDownload . . . 1187
9.12.2.7 pack_pds_GetXTRANetwork . . . 1188
9.12.2.8 pack_pds_GetXTRAValidity . . . 1188
9.12.2.9 pack_pds_PDSInjectTimeReference . . . 1188
9.12.2.10 pack_pds_ResetPDSData . . . 1189
9.12.2.11 pack_pds_SetEventReportCallback . . . 1189
9.12.2.12 pack_pds_SetPDSDefaults . . . 1189
9.12.2.13 pack_pds_SetPDSState . . . 1190
9.12.2.14 pack_pds_SetPortAutomaticTracking . . . 1190
9.12.2.15 pack_pds_SetServiceAutomaticTracking . . . 1190
9.12.2.16 pack_pds_SetXTRAAutomaticDownload . . . 1191
9.12.2.17 pack_pds_SetXTRANetwork . . . 1191
9.12.2.18 pack_pds_SLQSGetAGPSConfig . . . 1192
9.12.2.19 pack_pds_SLQSGetGPSStateInfo . . . 1192
9.12.2.20 pack_pds_SLQSPDSDeterminePosition . . . 1192
9.12.2.21 pack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1193
9.12.2.22 pack_pds_SLQSPDSInjectPositionData . . . 1193
9.12.2.23 pack_pds_SLQSSetAGPSConfig . . . 1193
9.12.2.24 pack_pds_SLQSSetPositionMethodState . . . 1194
9.12.2.25 pack_pds_StartPDSTrackingSessionExt . . . 1194
9.12.2.26 pack_pds_StopPDSTrackingSession . . . 1194
9.12.2.27 unpack_pds_ForceXTRADownload . . . 1195
9.12.2.28 unpack_pds_GetPDSDefaults . . . 1195
9.12.2.29 unpack_pds_GetPDSState . . . 1195

9.12.2.30 unpack_pds_GetPortAutomaticTracking . . . 1196
9.12.2.31 unpack_pds_GetServiceAutomaticTracking . . . 1196
9.12.2.32 unpack_pds_GetXTRAAutomaticDownload . . . 1197
9.12.2.33 unpack_pds_GetXTRANetwork . . . 1197
9.12.2.34 unpack_pds_GetXTRAValidity . . . 1197
9.12.2.35 unpack_pds_PDSInjectTimeReference . . . 1198
9.12.2.36 unpack_pds_ResetPDSData . . . 1198
9.12.2.37 unpack_pds_SetEventReport_Ind . . . 1198
9.12.2.38 unpack_pds_SetEventReportCallback . . . 1199
9.12.2.39 unpack_pds_SetPDSDefaults . . . 1199
9.12.2.40 unpack_pds_SetPDSState . . . 1199
9.12.2.41 unpack_pds_SetPdsState_Ind . . . 1200
9.12.2.42 unpack_pds_SetPortAutomaticTracking . . . 1200
9.12.2.43 unpack_pds_SetServiceAutomaticTracking . . . 1200
9.12.2.44 unpack_pds_SetXTRAAutomaticDownload . . . 1201
9.12.2.45 unpack_pds_SetXTRANetwork . . . 1201
9.12.2.46 unpack_pds_SLQSGetAGPSConfig . . . 1201
9.12.2.47 unpack_pds_SLQSGetGPSStateInfo . . . 1202
9.12.2.48 unpack_pds_SLQSPDSDeterminePosition . . . 1202
9.12.2.49 unpack_pds_SLQSPDSInjectAbsoluteTimeReference . . . 1202
9.12.2.50 unpack_pds_SLQSPDSInjectPositionData . . . 1203
9.12.2.51 unpack_pds_SLQSSetAGPSConfig . . . 1203
9.12.2.52 unpack_pds_SLQSSetPositionMethodState . . . 1203
9.12.2.53 unpack_pds_StartPDSTrackingSessionExt . . . 1204
9.12.2.54 unpack_pds_StopPDSTrackingSession . . . 1204
9.13 qaGobiApiTableBandClasses.h File Reference . . . 1204
9.13.1 Detailed Description . . . 1204
9.13.2 Band Classes (Value - Description) . . . 1205
9.13.2.1 LTE Bands . . . 1206
9.14 qaGobiApiTableCallControlReturnReasons.h File Reference . . . 1207
9.14.1 Detailed Description . . . 1207
9.14.2 S1 . . . 1208
9.15 qaGobiApiTableCallEndReasons.h File Reference . . . 1208
9.15.1 Detailed Description . . . 1208
9.15.2 Call end reason codes (Code - Reason) . . . 1209
9.15.2.1 Technology-agnostic call end reasons . . . 1209
9.15.2.2 EVDO CDMA 1xEV-DO . . . 1209
9.15.2.3 WCDMA/GSM call end reasons . . . 1210
9.15.2.4 EVDO CDMA 1xEV-DO . . . 1212
9.15.2.5 call end reason type . . . 1212

9.15.2.6 Mobile IP call end reasons (Type=1) . . . 1212
9.15.2.7 Internal call end reasons (Type=2) . . . 1214
9.15.2.8 Call Manager defined call end reasons (Type=3) . . . 1217
9.15.2.9 3GPP specification defined call end reasons (Type=6) . . . 1221
9.15.2.10 PPP call end reasons (Type=7) . . . 1223
9.15.2.11 EHRPD call end reasons (Type=8) . . . 1224
9.15.2.12 IPV6 call end reasons (Type=9) . . . 1225
9.16 qaGobiApiTableCarrierCodes.h File Reference . . . 1225
9.16.1 Detailed Description . . . 1225
9.16.2 Carrier Codes (Number - Carrier) . . . 1225
9.17 qaGobiApiTableCodingScheme.h File Reference . . . 1227
9.17.1 Detailed Description . . . 1227
9.17.2 S1 . . . 1227
9.17.2.1 Use of bits 3..0\n\n . . . 1227
9.17.3 Coding Group Bits 7..4(0001) . . . 1227
9.17.3.1 use of bits 3..0 . . . 1227
9.17.4 Coding Group Bits 7..4(0010) . . . 1228
9.17.4.1 use of bits 3..0 . . . 1228
9.17.5 Coding Group Bits 7..4(0011) . . . 1228
9.17.5.1 use of bits 3..0 . . . 1228
9.17.6 Coding Group Bits 7..4(01xx) . . . 1228
9.17.6.1 use of bits 3..0 . . . 1228
9.17.7 Coding Group Bits 7..4(1001) . . . 1229
9.17.7.1 Reserved coding groups . . . 1229
9.17.8 Coding Group Bits 7..4(1010..1101) . . . 1229
9.17.8.1 Reserved coding groups . . . 1229
9.17.9 Coding Group Bits 7..4(1110) . . . 1229
9.17.9.1 Defined by the WAP Forum . . . 1229
9.17.10 Coding Group Bits 7..4 (1111) . . . 1229
9.17.10.1 Data coding / message handling . . . 1229
9.17.11 Macro Definition Documentation . . . 1229
9.17.11.1 __GOBI_API_CODING_SCHEME_H__ . . . 1229
9.18 qaGobiApiTableGpsCapabilityCodes.h File Reference . . . 1229
9.18.1 Detailed Description . . . 1230
9.18.2 GPS capability (Value - Capability) . . . 1230
9.19 qaGobiApiTablePowerModes.h File Reference . . . 1230
9.19.1 Detailed Description . . . 1230
9.19.2 Power Modes (Value - Description) . . . 1230
9.20 qaGobiApiTableRadioInterfaces.h File Reference . . . 1231
9.20.1 Detailed Description . . . 1231

9.20.2 Radio interface . . . 1231
9.20.2.1 Technology (Value - Radio Interface Technology) . . . 1231
9.21 qaGobiApiTableRegionCodes.h File Reference . . . 1231
9.21.1 Detailed Description . . . 1231
9.21.2 Region Codes (Code - Region) . . . 1231
9.22 qaGobiApiTableServiceOptions.h File Reference . . . 1232
9.22.1 Detailed Description . . . 1232
9.22.2 Service Option codes (Code - Reason) . . . 1232
9.22.2.1 Description . . . 1232
9.23 qaGobiApiTableSupServiceInfoClasses.h File Reference . . . 1234
9.23.1 Detailed Description . . . 1234
9.23.2 Supplementary Service Information Classes (Value - Service Class) . . . 1234
9.24 qaGobiApiTableSwiAudio.h File Reference . . . 1234
9.24.1 Detailed Description . . . 1234
9.24.2 ACDB Device (Device ID - description) . . . 1235
9.24.3 Physical Interface (Device ID - description - Interface parameters) . . . 1235
9.25 qaGobiApiTableSwiOMADMSessionStatus.h File Reference . . . 1235
9.25.1 Detailed Description . . . 1235
9.25.2 OMA DM Session Status (Session Status - Meaning - Usage) . . . 1235
9.26 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference . . . 1236
9.26.1 Detailed Description . . . 1236
9.26.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage) . . . 1236
9.27 qaGobiApiTableVoiceCallEndReasons.h File Reference . . . 1237
9.27.1 Detailed Description . . . 1237
9.27.2 Voice Call and supplementary services end reason codes (Code - Reason) . . . 1238
9.27.2.1 General . . . 1238
9.27.2.2 service Errors . . . 1239
9.27.2.3 control cause values . . . 1240
9.27.2.4 reject causes . . . 1242
9.27.2.5 reject causes . . . 1242
9.27.2.6 reject causes . . . 1243
9.27.2.7 stratum reject causes . . . 1243
9.27.2.8 reject causes . . . 1243
9.27.2.9 IP end reasons . . . 1243
9.28 qmerrno.h File Reference . . . 1244
9.28.1 Enumeration Type Documentation . . . 1246
9.28.1.1 eQCWWANError . . . 1246
9.28.1.2 qm_wds_ds_profile_extended_err_codes . . . 1251
9.29 qos.h File Reference . . . 1251
9.29.1 Macro Definition Documentation . . . 1253

9.29.1.1 LITEQMI_MAX_QOS_FILTERS . . . 1253
9.29.1.2 LITEQMI_MAX_QOS_FLOW_PER_APN_STATS . . . 1253
9.29.1.3 LITEQMI_MAX_QOS_FLOWS . . . 1253
9.29.2 Function Documentation . . . 1253
9.29.2.1 pack_qos_BindDataPort . . . 1253
9.29.2.2 pack_qos_SLQSQosGetNetworkStatus . . . 1253
9.29.2.3 pack_qos_SLQSQosSwiReadApnExtraParams . . . 1254
9.29.2.4 pack_qos_SLQSQosSwiReadDataStats . . . 1255
9.29.2.5 pack_qos_SLQSSetQosEventCallback . . . 1255
9.29.2.6 unpack_qos_BindDataPort . . . 1256
9.29.2.7 unpack_qos_SLQSQosGetNetworkStatus . . . 1256
9.29.2.8 unpack_qos_SLQSQosSwiReadApnExtraParams . . . 1257
9.29.2.9 unpack_qos_SLQSQosSwiReadDataStats . . . 1257
9.29.2.10 unpack_qos_SLQSSetQosEventCallback . . . 1258
9.29.2.11 unpack_qos_SLQSSetQosEventCallback_ind . . . 1258
9.29.2.12 unpack_qos_SLQSSetQosNWStatusCallback_ind . . . 1259
9.29.2.13 unpack_qos_SLQSSetQosPriEventCallback_ind . . . 1259
9.29.2.14 unpack_qos_SLQSSetQosStatusCallback_ind . . . 1260
9.30 rms.h File Reference . . . 1261
9.30.1 Macro Definition Documentation . . . 1261
9.30.1.1 __LITEQMI_RMS_H__ . . . 1261
9.30.2 Function Documentation . . . 1261
9.30.2.1 pack_rms_GetSMSWake . . . 1261
9.30.2.2 pack_rms_SetSMSWake . . . 1262
9.30.2.3 unpack_rms_GetSMSWake . . . 1262
9.30.2.4 unpack_rms_SetSMSWake . . . 1262
9.31 sar.h File Reference . . . 1263
9.31.1 Typedef Documentation . . . 1263
9.31.1.1 unpack_sar_SLQSSetRfSarState_t . . . 1263
9.31.2 Function Documentation . . . 1263
9.31.2.1 pack_sar_SLQSGetRfSarState . . . 1263
9.31.2.2 pack_sar_SLQSSetRfSarState . . . 1263
9.31.2.3 unpack_sar_SLQSGetRfSarState . . . 1264
9.31.2.4 unpack_sar_SLQSSetRfSarState . . . 1264
9.32 sms.h File Reference . . . 1264
9.32.1 Macro Definition Documentation . . . 1268
9.32.1.1 MAX_CDMA_ENC_MO_TXT_MSG_SIZE . . . 1268
9.32.1.2 MAX_MS_TRANSFER_ROUTE_MSG . . . 1268
9.32.1.3 MAX_MSC_ADDRESS_SIZE . . . 1268
9.32.1.4 MAX_MSE_TWS_MSG . . . 1268

9.32.1.5 MAX_SMS_LIST_SIZE . . . 1268
9.32.1.6 MAX_SMS_MESSAGE_SIZE . . . 1268
9.32.1.7 SMS_CONFIG_LEN . . . 1268
9.32.1.8 SMS_MAX_SMS_ROUTES . . . 1268
9.32.1.9 SMS_NUM_OF_SET . . . 1268
9.32.1.10 SMSC_TYPE_LEN . . . 1268
9.32.2 Enumeration Type Documentation . . . 1268
9.32.2.1 eqmiCbkSetStatus . . . 1269
9.32.3 Function Documentation . . . 1269
9.32.3.1 pack_sms_GetSMSCAddress . . . 1269
9.32.3.2 pack_sms_SaveSMS . . . 1269
9.32.3.3 pack_sms_SendSMS . . . 1269
9.32.3.4 pack_sms_SetNewSMSCallback . . . 1270
9.32.3.5 pack_sms_SetSMSCAddress . . . 1270
9.32.3.6 pack_sms_SLQSDeleteSMS . . . 1271
9.32.3.7 pack_sms_SLQSGetIndicationRegister . . . 1271
9.32.3.8 pack_sms_SLQSGetMessageWaiting . . . 1271
9.32.3.9 pack_sms_SLQSGetSMS . . . 1272
9.32.3.10 pack_sms_SLQSGetSmsBroadcastConfig . . . 1272
9.32.3.11 pack_sms_SLQSGetSMSList . . . 1272
9.32.3.12 pack_sms_SLQSGetTransLayerInfo . . . 1273
9.32.3.13 pack_sms_SLQSGetTransNWRegInfo . . . 1273
9.32.3.14 pack_sms_SLQSModifySMSStatus . . . 1273
9.32.3.15 pack_sms_SLQSSendAsyncSMS . . . 1274
9.32.3.16 pack_sms_SLQSSetIndicationRegister . . . 1274
9.32.3.17 pack_sms_SLQSSetSmsBroadcastActivation . . . 1274
9.32.3.18 pack_sms_SLQSSetSmsBroadcastConfig . . . 1275
9.32.3.19 pack_sms_SLQSSetSmsStorage . . . 1275
9.32.3.20 pack_sms_SLQSSmsGetMaxStorageSize . . . 1276
9.32.3.21 pack_sms_SLQSSmsGetMessageProtocol . . . 1276
9.32.3.22 pack_sms_SLQSSmsSetRoutes . . . 1276
9.32.3.23 pack_sms_SLQSSwiGetSMSStorage . . . 1277
9.32.3.24 unpack_sms_GetSMSCAddress . . . 1277
9.32.3.25 unpack_sms_SaveSMS . . . 1277
9.32.3.26 unpack_sms_SendSMS . . . 1278
9.32.3.27 unpack_sms_SetNewSMSCallback . . . 1278
9.32.3.28 unpack_sms_SetNewSMSCallback_ind . . . 1278
9.32.3.29 unpack_sms_SetSMSCAddress . . . 1279
9.32.3.30 unpack_sms_SLQSDeleteSMS . . . 1279
9.32.3.31 unpack_sms_SLQSGetIndicationRegister . . . 1279

9.32.3.32 unpack_sms_SLQSGetMessageWaiting . . . 1280
9.32.3.33 unpack_sms_SLQSGetSMS . . . 1280
9.32.3.34 unpack_sms_SLQSGetSmsBroadcastConfig . . . 1280
9.32.3.35 unpack_sms_SLQSGetSMSList . . . 1281
9.32.3.36 unpack_sms_SLQSGetTransLayerInfo . . . 1281
9.32.3.37 unpack_sms_SLQSGetTransNWRegInfo . . . 1281
9.32.3.38 unpack_sms_SLQSModifySMSStatus . . . 1282
9.32.3.39 unpack_sms_SLQSNWRegInfoCallback_ind . . . 1282
9.32.3.40 unpack_sms_SLQSSendAsyncSMS . . . 1282
9.32.3.41 unpack_sms_SLQSSetIndicationRegister . . . 1283
9.32.3.42 unpack_sms_SLQSSetSmsBroadcastActivation . . . 1283
9.32.3.43 unpack_sms_SLQSSetSmsBroadcastConfig . . . 1283
9.32.3.44 unpack_sms_SLQSSetSmsStorage . . . 1284
9.32.3.45 unpack_sms_SLQSSmsGetMaxStorageSize . . . 1284
9.32.3.46 unpack_sms_SLQSSmsGetMessageProtocol . . . 1284
9.32.3.47 unpack_sms_SLQSSmsSetRoutes . . . 1285
9.32.3.48 unpack_sms_SLQSSwiGetSMSStorage . . . 1285
9.32.3.49 unpack_sms_SLQSTransLayerInfoCallback_ind . . . 1286
9.32.3.50 unpack_sms_SLQSWmsAsyncRawSendCallBack_ind . . . 1286
9.32.3.51 unpack_sms_SLQSWmsMemoryFullCallBack_ind . . . 1286
9.32.3.52 unpack_sms_SLQSWmsMessageWaitingCallBack_ind . . . 1287
9.33 swiaudio.h File Reference . . . 1287
9.33.1 Macro Definition Documentation . . . 1288
9.33.1.1 SWIAUDIO_MAX_LEN_IFACE_TABLE . . . 1288
9.33.2 Typedef Documentation . . . 1288
9.33.2.1 unpack_swiaudio_SLQSSetM2MAudioAVCFG_t . . . 1288
9.33.2.2 unpack_swiaudio_SLQSSetM2MAudioLPBK_t . . . 1288
9.33.2.3 unpack_swiaudio_SLQSSetM2MAudioNVDef_t . . . 1289
9.33.2.4 unpack_swiaudio_SLQSSetM2MAudioProfile_t . . . 1289
9.33.2.5 unpack_swiaudio_SLQSSetM2MAudioVolume_t . . . 1289
9.33.2.6 unpack_swiaudio_SLQSSetM2MAVMute_t . . . 1289
9.33.2.7 unpack_swiaudio_SLQSSetM2MSpkrGain_t . . . 1289
9.33.3 Function Documentation . . . 1289
9.33.3.1 pack_swiaudio_SLQSGetM2MAudioProfile . . . 1289
9.33.3.2 pack_swiaudio_SLQSGetM2MAudioVolume . . . 1289
9.33.3.3 pack_swiaudio_SLQSGetM2MAVMute . . . 1289
9.33.3.4 pack_swiaudio_SLQSGetM2MSpkrGain . . . 1290
9.33.3.5 pack_swiaudio_SLQSSetM2MAudioAVCFG . . . 1290
9.33.3.6 pack_swiaudio_SLQSSetM2MAudioLPBK . . . 1291
9.33.3.7 pack_swiaudio_SLQSSetM2MAudioNVDef . . . 1291

9.33.3.8 pack_swiaudio_SLQSSetM2MAudioProfile . . . 1291
9.33.3.9 pack_swiaudio_SLQSSetM2MAudioVolume . . . 1292
9.33.3.10 pack_swiaudio_SLQSSetM2MAVMute . . . 1292
9.33.3.11 pack_swiaudio_SLQSSetM2MSpkrGain . . . 1292
9.33.3.12 unpack_swiaudio_SLQSGetM2MAudioProfile . . . 1293
9.33.3.13 unpack_swiaudio_SLQSGetM2MAudioVolume . . . 1293
9.33.3.14 unpack_swiaudio_SLQSGetM2MAVMute . . . 1293
9.33.3.15 unpack_swiaudio_SLQSGetM2MSpkrGain . . . 1294
9.33.3.16 unpack_swiaudio_SLQSSetM2MAudioAVCFG . . . 1294
9.33.3.17 unpack_swiaudio_SLQSSetM2MAudioLPBK . . . 1294
9.33.3.18 unpack_swiaudio_SLQSSetM2MAudioNVDef . . . 1295
9.33.3.19 unpack_swiaudio_SLQSSetM2MAudioProfile . . . 1295
9.33.3.20 unpack_swiaudio_SLQSSetM2MAudioVolume . . . 1295
9.33.3.21 unpack_swiaudio_SLQSSetM2MAVMute . . . 1296
9.33.3.22 unpack_swiaudio_SLQSSetM2MSpkrGain . . . 1296
9.34 swiavms.h File Reference . . . 1296
9.34.1 Macro Definition Documentation . . . 1298
9.34.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 1298
9.34.1.2 MAX_AVMS_SETTINGS_RETRY_TIMER_NUMBER . . . 1298
9.34.1.3 MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINF-O_DESC_LENGTH . . . 1298
9.34.1.4 MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINF-O_NAME_LENGTH . . . 1298
9.34.1.5 MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINF-O_VERSION_LENGTH . . . 1298
9.34.1.6 MAX_PACK_SWI_AVMS_SESSIONGETINFO_CONFIG_ALERT_MSG_LENG-TH . . . 1298
9.34.1.7 MAX_PACK_SWI_AVMS_SET_SETTING_APN_LENGTH . . . 1298
9.34.1.8 MAX_PACK_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS . 1298
9.34.1.9 MAX_PACK_SWI_AVMS_SET_SETTING_PWD_LENGTH . . . 1298
9.34.1.10 MAX_PACK_SWI_AVMS_SET_SETTING_UNAME_LENGTH . . . 1298
9.34.2 Function Documentation . . . 1299
9.34.2.1 pack_swiavms_SLQSAVMSGetSettings . . . 1299
9.34.2.2 pack_swiavms_SLQSAVMSGetSettings_v2 . . . 1299
9.34.2.3 pack_swiavms_SLQSAVMSSendSelection . . . 1300
9.34.2.4 pack_swiavms_SLQSAVMSSessionGetInfo . . . 1300
9.34.2.5 pack_swiavms_SLQSAvmsSetEventReport . . . 1301
9.34.2.6 pack_swiavms_SLQSAVMSSetSettings . . . 1301
9.34.2.7 pack_swiavms_SLQSAVMSSetSettings_v2 . . . 1302
9.34.2.8 pack_swiavms_SLQSAVMSSetSettingsNoAutoRebootField . . . 1303
9.34.2.9 pack_swiavms_SLQSAVMSStartSession . . . 1303

9.34.2.10 pack_swiavms_SLQSAVMSStopSession . . . 1304
9.34.2.11 unpack_swiavms_SLQSAVMSEventReportInd . . . 1305
9.34.2.12 unpack_swiavms_SLQSAVMSGetSettings . . . 1305
9.34.2.13 unpack_swiavms_SLQSAVMSGetSettings_v2 . . . 1306
9.34.2.14 unpack_swiavms_SLQSAVMSSendSelection . . . 1306
9.34.2.15 unpack_swiavms_SLQSAVMSSessionGetInfo . . . 1306
9.34.2.16 unpack_swiavms_SLQSAvmsSetEventReport . . . 1307
9.34.2.17 unpack_swiavms_SLQSAVMSSetSettings . . . 1307
9.34.2.18 unpack_swiavms_SLQSAVMSSetSettings_v2 . . . 1308
9.34.2.19 unpack_swiavms_SLQSAVMSStartSession . . . 1308
9.34.2.20 unpack_swiavms_SLQSAVMSStopSession . . . 1309
9.34.2.21 unpack_swiavms_SLQSAVMSStopSession_avc2 . . . 1309
9.35 SwiDataTypes.h File Reference . . . 1310
9.35.1 Detailed Description . . . 1310
9.35.2 Macro Definition Documentation . . . 1310
9.35.2.1 QMI_NO_LTE_FW_SUPPORT . . . 1310
9.35.2.2 QMI_TLV_PLACEHOLDER . . . 1310
9.35.2.3 SWI_API . . . 1310
9.35.2.4 UNUSEDPARAM . . . 1310
9.35.3 Typedef Documentation . . . 1311
9.35.3.1 BOOL . . . 1311
9.35.3.2 BYTE . . . 1311
9.35.3.3 CHAR . . . 1311
9.35.3.4 FLOAT . . . 1311
9.35.3.5 INT32 . . . 1311
9.35.3.6 INT8 . . . 1311
9.35.3.7 LPCSTR . . . 1311
9.35.3.8 SHORT . . . 1311
9.35.3.9 ULONG . . . 1311
9.35.3.10 ULONGLONG . . . 1311
9.35.3.11 USHORT . . . 1311
9.35.3.12 WORD . . . 1311
9.36 swidms.h File Reference . . . 1311
9.36.1 Typedef Documentation . . . 1312
9.36.1.1 unpack_swidms_SLQSSwiDmsSetUsbNetNum_t . . . 1312
9.36.2 Function Documentation . . . 1312
9.36.2.1 pack_swidms_SLQSSwiDmsGetMTU . . . 1312
9.36.2.2 pack_swidms_SLQSSwiDmsGetUsbNetNum . . . 1312
9.36.2.3 pack_swidms_SLQSSwiDmsSetMTU . . . 1312
9.36.2.4 pack_swidms_SLQSSwiDmsSetUsbNetNum . . . 1313

9.36.2.5 unpack_swidms_SLQSSwiDmsGetMTU . . . 1313
9.36.2.6 unpack_swidms_SLQSSwiDmsGetUsbNetNum . . . 1314
9.36.2.7 unpack_swidms_SLQSSwiDmsSetMTU . . . 1314
9.36.2.8 unpack_swidms_SLQSSwiDmsSetUsbNetNum . . . 1314
9.37 swiloc.h File Reference . . . 1315
9.37.1 Typedef Documentation . . . 1315
9.37.1.1 unpack_swiloc_SwiLocSetAutoStart_t . . . 1315
9.37.2 Function Documentation . . . 1315
9.37.2.1 pack_swiloc_SwiLocGetAutoStart . . . 1315
9.37.2.2 pack_swiloc_SwiLocSetAutoStart . . . 1315
9.37.2.3 unpack_swiloc_SwiLocGetAutoStart . . . 1316
9.37.2.4 unpack_swiloc_SwiLocSetAutoStart . . . 1316
9.38 swioma.h File Reference . . . 1316
9.38.1 Macro Definition Documentation . . . 1317
9.38.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 1317
9.38.2 Function Documentation . . . 1317
9.38.2.1 pack_swioma_SLQSOMADMAlertCallback . . . 1317
9.38.2.2 pack_swioma_SLQSOMADMCancelSession . . . 1318
9.38.2.3 pack_swioma_SLQSOMADMGetSessionInfo . . . 1319
9.38.2.4 pack_swioma_SLQSOMADMGetSettings . . . 1319
9.38.2.5 pack_swioma_SLQSOMADMSendSelection . . . 1320
9.38.2.6 pack_swioma_SLQSOMADMSetSettings . . . 1321
9.38.2.7 pack_swioma_SLQSOMADMStartSession . . . 1321
9.38.2.8 unpack_swioma_SLQSOMADMAlertCallback . . . 1322
9.38.2.9 unpack_swioma_SLQSOMADMAlertCallback_ind . . . 1322
9.38.2.10 unpack_swioma_SLQSOMADMCancelSession . . . 1323
9.38.2.11 unpack_swioma_SLQSOMADMGetSessionInfo . . . 1323
9.38.2.12 unpack_swioma_SLQSOMADMGetSettings . . . 1324
9.38.2.13 unpack_swioma_SLQSOMADMSendSelection . . . 1324
9.38.2.14 unpack_swioma_SLQSOMADMSetSettings . . . 1324
9.38.2.15 unpack_swioma_SLQSOMADMStartSession . . . 1325
9.39 swiomaext.h File Reference . . . 1325
9.39.1 Macro Definition Documentation . . . 1326
9.39.1.1 LITEQMI_MAX_SWIOMA_STR_LEN . . . 1326
9.39.2 Typedef Documentation . . . 1326
9.39.2.1 unpack_swioma_SLQSOMADMCancelSessionExt_t . . . 1326
9.39.2.2 unpack_swioma_SLQSOMADMSendSelectionExt_t . . . 1326
9.39.2.3 unpack_swioma_SLQSOMADMSetSettingsExt_t . . . 1326
9.39.2.4 unpack_swioma_SLQSOMADMStartSessionExt_t . . . 1326
9.39.3 Function Documentation . . . 1326

9.39.3.1 pack_swioma_SLQSOMADMCancelSessionExt . . . 1326
9.39.3.2 pack_swioma_SLQSOMADMSendSelectionExt . . . 1327
9.39.3.3 pack_swioma_SLQSOMADMSetSettingsExt . . . 1328
9.39.3.4 pack_swioma_SLQSOMADMStartSessionExt . . . 1328
9.39.3.5 unpack_swioma_SLQSOMADMCancelSessionExt . . . 1329
9.39.3.6 unpack_swioma_SLQSOMADMSendSelectionExt . . . 1329
9.39.3.7 unpack_swioma_SLQSOMADMSetSettingsExt . . . 1330
9.39.3.8 unpack_swioma_SLQSOMADMStartSessionExt . . . 1330
9.40 switype_256bit.h File Reference . . . 1331
9.40.1 Macro Definition Documentation . . . 1331
9.40.1.1 SWI_UINT256_BIT_OFFSET . . . 1331
9.40.1.2 SWI_UINT256_BITS_PER_WORD . . . 1331
9.40.1.3 SWI_UINT256_INT_VALUE . . . 1331
9.40.1.4 SWI_UINT256_WORD_COUNT . . . 1331
9.40.1.5 SWI_UINT256_WORD_OFFSET . . . 1331
9.40.2 Function Documentation . . . 1331
9.40.2.1 swi_uint256_clear_bit . . . 1331
9.40.2.2 swi_uint256_get_bit . . . 1331
9.40.2.3 swi_uint256_print_mask . . . 1332
9.40.2.4 swi_uint256_set_bit . . . 1332
9.41 SWIWWANCMAPI.h File Reference . . . 1332
9.42 tmd.h File Reference . . . 1332
9.42.1 Macro Definition Documentation . . . 1333
9.42.1.1 MAX_MITIGATION_DEV_ID_LEN . . . 1333
9.42.1.2 MAX_MITIGATION_DEV_LIST_LEN . . . 1333
9.42.1.3 TMD_MAX_DEV_LIST . . . 1333
9.42.2 Function Documentation . . . 1333
9.42.2.1 pack_tmd_SLQSTmdDeRegNotMitigationLvl . . . 1333
9.42.2.2 pack_tmd_SLQSTmdGetMitigationDevList . . . 1333
9.42.2.3 pack_tmd_SLQSTmdGetMitigationLvl . . . 1334
9.42.2.4 pack_tmd_SLQSTmdRegNotMitigationLvl . . . 1334
9.42.2.5 unpack_tmd_SLQSTmdDeRegNotMitigationLvl . . . 1335
9.42.2.6 unpack_tmd_SLQSTmdGetMitigationDevList . . . 1335
9.42.2.7 unpack_tmd_SLQSTmdGetMitigationLvl . . . 1335
9.42.2.8 unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind . . . 1336
9.42.2.9 unpack_tmd_SLQSTmdRegNotMitigationLvl . . . 1336
9.43 uim.h File Reference . . . 1336
9.43.1 Macro Definition Documentation . . . 1339
9.43.1.1 MAX_DESCRIPTION_LENGTH . . . 1339
9.43.1.2 MAX_ICCID_LENGTH . . . 1339

9.43.1.3 MAX_NO_OF_APPLICATIONS . . . 1339
9.43.1.4 MAX_NO_OF_SLOTS . . . 1339
9.43.1.5 MAX_SLOTS_STATUS . . . 1339
9.43.1.6 UIM_MAX_ACTIVE_PERS_FEATURES . . . 1339
9.43.1.7 UIM_MAX_CONTENT_LENGTH . . . 1339
9.43.1.8 UIM_MAX_DESCRIPTION_LENGTH . . . 1339
9.43.1.9 UIM_MAX_NO_OF_APPLICATIONS . . . 1339
9.43.1.10 UIM_MAX_NO_OF_SLOTS . . . 1339
9.43.1.11 UIM_UINT8_MAX_STRING_SZ . . . 1340
9.43.2 Function Documentation . . . 1340
9.43.2.1 pack_uim_ChangePin . . . 1340
9.43.2.2 pack_uim_GetCardStatus . . . 1340
9.43.2.3 pack_uim_ReadTransparent . . . 1340
9.43.2.4 pack_uim_SetPinProtection . . . 1341
9.43.2.5 pack_uim_SLQSUIMAuthenticate . . . 1341
9.43.2.6 pack_uim_SLQSUIMDepersonalization . . . 1341
9.43.2.7 pack_uim_SLQSUIMEventRegister . . . 1342
9.43.2.8 pack_uim_SLQSUIMGetConfiguration . . . 1342
9.43.2.9 pack_uim_SLQSUIMGetFileAttributes . . . 1343
9.43.2.10 pack_uim_SLQSUIMGetSlotsStatus . . . 1343
9.43.2.11 pack_uim_SLQSUIMPowerDown . . . 1343
9.43.2.12 pack_uim_SLQSUIMPowerUp . . . 1344
9.43.2.13 pack_uim_SLQSUIMRefreshComplete . . . 1344
9.43.2.14 pack_uim_SLQSUIMRefreshGetLastEvent . . . 1344
9.43.2.15 pack_uim_SLQSUIMRefreshOK . . . 1345
9.43.2.16 pack_uim_SLQSUIMRefreshRegister . . . 1345
9.43.2.17 pack_uim_SLQSUIMReset . . . 1345
9.43.2.18 pack_uim_SLQSUIMSwitchSlot . . . 1346
9.43.2.19 pack_uim_UnblockPin . . . 1346
9.43.2.20 pack_uim_VerifyPin . . . 1347
9.43.2.21 unpack_uim_ChangePin . . . 1347
9.43.2.22 unpack_uim_GetCardStatus . . . 1347
9.43.2.23 unpack_uim_ReadTransparent . . . 1348
9.43.2.24 unpack_uim_SetPinProtection . . . 1348
9.43.2.25 unpack_uim_SetUimSlotStatusChangeCallback_ind . . . 1348
9.43.2.26 unpack_uim_SLQSUIMAuthenticate . . . 1349
9.43.2.27 unpack_uim_SLQSUIMDepersonalization . . . 1349
9.43.2.28 unpack_uim_SLQSUIMEventRegister . . . 1349
9.43.2.29 unpack_uim_SLQSUIMGetConfiguration . . . 1350
9.43.2.30 unpack_uim_SLQSUIMGetFileAttributes . . . 1350

9.43.2.31 unpack_uim_SLQSUIMGetSlotsStatus . . . 1351
9.43.2.32 unpack_uim_SLQSUIMPowerDown . . . 1351
9.43.2.33 unpack_uim_SLQSUIMPowerUp . . . 1351
9.43.2.34 unpack_uim_SLQSUIMRefreshCallback_Ind . . . 1352
9.43.2.35 unpack_uim_SLQSUIMRefreshComplete . . . 1352
9.43.2.36 unpack_uim_SLQSUIMRefreshGetLastEvent . . . 1352
9.43.2.37 unpack_uim_SLQSUIMRefreshOK . . . 1353
9.43.2.38 unpack_uim_SLQSUIMRefreshRegister . . . 1353
9.43.2.39 unpack_uim_SLQSUIMReset . . . 1353
9.43.2.40 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind . . . 1353
9.43.2.41 unpack_uim_SLQSUIMSwitchSlot . . . 1354
9.43.2.42 unpack_uim_UnblockPin . . . 1354
9.43.2.43 unpack_uim_VerifyPin . . . 1355
9.44 voice.h File Reference . . . 1355
9.44.1 Detailed Description . . . 1360
9.44.2 S1 . . . 1360
9.44.3 Macro Definition Documentation . . . 1360
9.44.3.1 BARRING_PASSWORD_LENGTH . . . 1360
9.44.3.2 MAX_VOICE_CALL_NO_LEN . . . 1360
9.44.3.3 MAX_VOICE_DESCRIPTION_LENGTH . . . 1360
9.44.3.4 MAXVOICEUSSDLENGTH . . . 1360
9.44.3.5 VOICE_MAX_NO_OF_CALLS . . . 1360
9.44.4 Enumeration Type Documentation . . . 1360
9.44.4.1 liteServiceClassInformation . . . 1360
9.44.5 Function Documentation . . . 1361
9.44.5.1 pack_voice_AnswerUSSD . . . 1361
9.44.5.2 pack_voice_CancelUSSD . . . 1361
9.44.5.3 pack_voice_OriginateUSSD . . . 1361
9.44.5.4 pack_voice_SLQSOriginateUSSD . . . 1362
9.44.5.5 pack_voice_SLQSVoiceALSSelectLine . . . 1362
9.44.5.6 pack_voice_SLQSVoiceALSSetLineSwitching . . . 1362
9.44.5.7 pack_voice_SLQSVoiceAnswerCall . . . 1363
9.44.5.8 pack_voice_SLQSVoiceBindSubscription . . . 1363
9.44.5.9 pack_voice_SLQSVoiceBurstDTMF . . . 1364
9.44.5.10 pack_voice_SLQSVoiceDialCall . . . 1364
9.44.5.11 pack_voice_SLQSVoiceEndCall . . . 1364
9.44.5.12 pack_voice_SLQSVoiceGetAllCallInfo . . . 1365
9.44.5.13 pack_voice_SLQSVoiceGetCallBarring . . . 1365
9.44.5.14 pack_voice_SLQSVoiceGetCallForwardingStatus . . . 1365
9.44.5.15 pack_voice_SLQSVoiceGetCallInfo . . . 1366

9.44.5.16 pack_voice_SLQSVoiceGetCallWaiting . . . 1366
9.44.5.17 pack_voice_SLQSVoiceGetCLIP . . . 1367
9.44.5.18 pack_voice_SLQSVoiceGetCLIR . . . 1367
9.44.5.19 pack_voice_SLQSVoiceGetCNAP . . . 1367
9.44.5.20 pack_voice_SLQSVoiceGetCOLP . . . 1368
9.44.5.21 pack_voice_SLQSVoiceGetCOLR . . . 1368
9.44.5.22 pack_voice_SLQSVoiceGetConfig . . . 1368
9.44.5.23 pack_voice_SLQSVoiceIndicationRegister . . . 1369
9.44.5.24 pack_voice_SLQSVoiceManageCalls . . . 1369
9.44.5.25 pack_voice_SLQSVoiceOrigUSSDNoWait . . . 1369
9.44.5.26 pack_voice_SLQSVoiceSendFlash . . . 1370
9.44.5.27 pack_voice_SLQSVoiceSetCallBarringPassword . . . 1370
9.44.5.28 pack_voice_SLQSVoiceSetConfig . . . 1370
9.44.5.29 pack_voice_SLQSVoiceSetPreferredPrivacy . . . 1371
9.44.5.30 pack_voice_SLQSVoiceSetSUPSService . . . 1371
9.44.5.31 pack_voice_SLQSVoiceStartContDTMF . . . 1372
9.44.5.32 pack_voice_SLQSVoiceStopContDTMF . . . 1372
9.44.5.33 unpack_voice_allCallStatusCallback_ind . . . 1372
9.44.5.34 unpack_voice_AnswerUSSD . . . 1373
9.44.5.35 unpack_voice_CancelUSSD . . . 1373
9.44.5.36 unpack_voice_DTMFEventCallback_ind . . . 1373
9.44.5.37 unpack_voice_OriginateUSSD . . . 1374
9.44.5.38 unpack_voice_OTASPStatusCallback_ind . . . 1374
9.44.5.39 unpack_voice_SLQSOriginateUSSD . . . 1374
9.44.5.40 unpack_voice_SLQSVoiceALSSelectLine . . . 1375
9.44.5.41 unpack_voice_SLQSVoiceALSSetLineSwitching . . . 1375
9.44.5.42 unpack_voice_SLQSVoiceAnswerCall . . . 1375
9.44.5.43 unpack_voice_SLQSVoiceBindSubscription . . . 1376
9.44.5.44 unpack_voice_SLQSVoiceBurstDTMF . . . 1376
9.44.5.45 unpack_voice_SLQSVoiceDialCall . . . 1376
9.44.5.46 unpack_voice_SLQSVoiceEndCall . . . 1377
9.44.5.47 unpack_voice_SLQSVoiceGetAllCallInfo . . . 1377
9.44.5.48 unpack_voice_SLQSVoiceGetCallBarring . . . 1377
9.44.5.49 unpack_voice_SLQSVoiceGetCallForwardingStatus . . . 1378
9.44.5.50 unpack_voice_SLQSVoiceGetCallInfo . . . 1378
9.44.5.51 unpack_voice_SLQSVoiceGetCallWaiting . . . 1378
9.44.5.52 unpack_voice_SLQSVoiceGetCLIP . . . 1379
9.44.5.53 unpack_voice_SLQSVoiceGetCLIR . . . 1379
9.44.5.54 unpack_voice_SLQSVoiceGetCNAP . . . 1379
9.44.5.55 unpack_voice_SLQSVoiceGetCOLP . . . 1380

9.44.5.56 unpack_voice_SLQSVoiceGetCOLR . . . 1380
9.44.5.57 unpack_voice_SLQSVoiceGetConfig . . . 1380
9.44.5.58 unpack_voice_SLQSVoiceIndicationRegister . . . 1381
9.44.5.59 unpack_voice_SLQSVoiceManageCalls . . . 1381
9.44.5.60 unpack_voice_SLQSVoiceOrigUSSDNoWait . . . 1381
9.44.5.61 unpack_voice_SLQSVoiceSendFlash . . . 1382
9.44.5.62 unpack_voice_SLQSVoiceSetCallBarringPassword . . . 1382
9.44.5.63 unpack_voice_SLQSVoiceSetConfig . . . 1382
9.44.5.64 unpack_voice_SLQSVoiceSetPreferredPrivacy . . . 1383
9.44.5.65 unpack_voice_SLQSVoiceSetSUPSService . . . 1383
9.44.5.66 unpack_voice_SLQSVoiceStartContDTMF . . . 1383
9.44.5.67 unpack_voice_SLQSVoiceStopContDTMF . . . 1384
9.44.5.68 unpack_voice_SLQSVoiceSUPSCallback_ind . . . 1384
9.44.5.69 unpack_voice_SUPSNotificationCallback_ind . . . 1384
9.44.5.70 unpack_voice_USSDNotificationCallback_ind . . . 1385
9.44.5.71 unpack_voice_VoiceInfoRecCallback_ind . . . 1385
9.44.5.72 unpack_voice_voicePrivacyChangeCallback_ind . . . 1386
9.45 wds.h File Reference . . . 1386
9.45.1 Detailed Description . . . 1392
9.45.2 SO Mask . . . 1392
9.45.3 RAT Mask . . . 1392
9.45.4 Macro Definition Documentation . . . 1393
9.45.4.1 BYT_STAT_STAT_MASK . . . 1393
9.45.4.2 IPV6_ADDRESS_ARRAY_SIZE . . . 1393
9.45.4.3 MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE . . . 1393
9.45.4.4 PACK_WDS_IPV4 . . . 1393
9.45.4.5 PACK_WDS_IPV6 . . . 1393
9.45.4.6 WDS_DHCP_MAX_NUM_OPTIONS . . . 1393
9.45.4.7 WDS_DHCP_OPTION_DATA_BUF_SIZE . . . 1393
9.45.5 Typedef Documentation . . . 1393
9.45.5.1 UnpackQmiProfileInfo . . . 1393
9.45.6 Enumeration Type Documentation . . . 1394
9.45.6.1 liteQmiDataBearerMasks . . . 1394
9.45.7 Function Documentation . . . 1394
9.45.7.1 pack_wds_DHCPv4ClientLeaseChange . . . 1394
9.45.7.2 pack_wds_GetAutoconnect . . . 1394
9.45.7.3 pack_wds_GetByteTotals . . . 1395
9.45.7.4 pack_wds_GetConnectionRate . . . 1395
9.45.7.5 pack_wds_GetDataBearerTechnology . . . 1395
9.45.7.6 pack_wds_GetDefaultProfile . . . 1396

9.45.7.7 pack_wds_GetDefaultProfileNum . . . 1396
9.45.7.8 pack_wds_GetDormancyState . . . 1397
9.45.7.9 pack_wds_GetLastMobileIPError . . . 1397
9.45.7.10 pack_wds_GetMobileIP . . . 1398
9.45.7.11 pack_wds_GetMobileIPProfile . . . 1398
9.45.7.12 pack_wds_GetPacketStatistics . . . 1398
9.45.7.13 pack_wds_GetPacketStatus . . . 1399
9.45.7.14 pack_wds_GetSessionDuration . . . 1399
9.45.7.15 pack_wds_GetSessionState . . . 1400
9.45.7.16 pack_wds_RMSetTransferStatistics . . . 1400
9.45.7.17 pack_wds_SetAutoconnect . . . 1400
9.45.7.18 pack_wds_SetDefaultProfile . . . 1401
9.45.7.19 pack_wds_SetDefaultProfileNum . . . 1401
9.45.7.20 pack_wds_SetMobileIP . . . 1402
9.45.7.21 pack_wds_SetMobileIPParameters . . . 1402
9.45.7.22 pack_wds_SetMobileIPProfile . . . 1403
9.45.7.23 pack_wds_SetMuxID . . . 1403
9.45.7.24 pack_wds_SLQSCreateProfile . . . 1403
9.45.7.25 pack_wds_SLQSDeleteProfile . . . 1404
9.45.7.26 pack_wds_SLQSGet3GPPConfigItem . . . 1404
9.45.7.27 pack_wds_SLQSGetCurrDataSystemStat . . . 1405
9.45.7.28 pack_wds_SLQSGetCurrentChannelRate . . . 1405
9.45.7.29 pack_wds_SLQSGetDataBearerTechnology . . . 1405
9.45.7.30 pack_wds_SLQSGetDUNCallInfo . . . 1406
9.45.7.31 pack_wds_SLQSGetProfileSettings . . . 1406
9.45.7.32 pack_wds_SLQSGetRuntimeSettings . . . 1407
9.45.7.33 pack_wds_SLQSModifyProfile . . . 1407
9.45.7.34 pack_wds_SLQSResetPacketStatics . . . 1408
9.45.7.35 pack_wds_SLQSSet3GPPConfigItem . . . 1408
9.45.7.36 pack_wds_SLQSSetIPFamilyPreference . . . 1409
9.45.7.37 pack_wds_SLQSSetWdsEventCallback . . . 1409
9.45.7.38 pack_wds_SLQSSGetDHCPv4ClientConfig . . . 1409
9.45.7.39 pack_wds_SLQSSGetLoopback . . . 1410
9.45.7.40 pack_wds_SLQSSSetDHCPv4ClientConfig . . . 1410
9.45.7.41 pack_wds_SLQSSSetLoopback . . . 1410
9.45.7.42 pack_wds_SLQSStartDataSession . . . 1411
9.45.7.43 pack_wds_SLQSStopDataSession . . . 1411
9.45.7.44 pack_wds_SLQSWdsGoActive . . . 1412
9.45.7.45 pack_wds_SLQSWdsGoDormant . . . 1412
9.45.7.46 pack_wds_SLQSWdsSetEventReport . . . 1413

9.45.7.47 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . 1413
9.45.7.48 unpack_wds_DHCPv4ClientLease_ind . . . 1413
9.45.7.49 unpack_wds_DHCPv4ClientLeaseChange . . . 1414
9.45.7.50 unpack_wds_GetAutoconnect . . . 1414
9.45.7.51 unpack_wds_GetByteTotals . . . 1414
9.45.7.52 unpack_wds_GetConnectionRate . . . 1415
9.45.7.53 unpack_wds_GetDataBearerTechnology . . . 1415
9.45.7.54 unpack_wds_GetDefaultProfile . . . 1416
9.45.7.55 unpack_wds_GetDefaultProfileNum . . . 1416
9.45.7.56 unpack_wds_GetDormancyState . . . 1416
9.45.7.57 unpack_wds_GetLastMobileIPError . . . 1417
9.45.7.58 unpack_wds_GetMobileIP . . . 1417
9.45.7.59 unpack_wds_GetMobileIPProfile . . . 1417
9.45.7.60 unpack_wds_GetPacketStatistics . . . 1418
9.45.7.61 unpack_wds_GetPacketStatus . . . 1418
9.45.7.62 unpack_wds_GetSessionDuration . . . 1418
9.45.7.63 unpack_wds_GetSessionState . . . 1419
9.45.7.64 unpack_wds_RMSetTransferStatistics . . . 1419
9.45.7.65 unpack_wds_RMTransferStatistics_ind . . . 1419
9.45.7.66 unpack_wds_SetAutoconnect . . . 1420
9.45.7.67 unpack_wds_SetDefaultProfile . . . 1420
9.45.7.68 unpack_wds_SetDefaultProfileNum . . . 1420
9.45.7.69 unpack_wds_SetMobileIP . . . 1421
9.45.7.70 unpack_wds_SetMobileIPParameters . . . 1421
9.45.7.71 unpack_wds_SetMobileIPProfile . . . 1421
9.45.7.72 unpack_wds_SetMuxID . . . 1422
9.45.7.73 unpack_wds_SLQSCreateProfile . . . 1422
9.45.7.74 unpack_wds_SLQSDeleteProfile . . . 1422
9.45.7.75 unpack_wds_SLQSDUNCallInfoCallBack_ind . . . 1423
9.45.7.76 unpack_wds_SLQSGet3GPPConfigItem . . . 1423
9.45.7.77 unpack_wds_SLQSGetCurrDataSystemStat . . . 1423
9.45.7.78 unpack_wds_SLQSGetCurrentChannelRate . . . 1424
9.45.7.79 unpack_wds_SLQSGetDataBearerTechnology . . . 1424
9.45.7.80 unpack_wds_SLQSGetDUNCallInfo . . . 1424
9.45.7.81 unpack_wds_SLQSGetProfileSettings . . . 1425
9.45.7.82 unpack_wds_SLQSGetRuntimeSettings . . . 1425
9.45.7.83 unpack_wds_SLQSModifyProfile . . . 1425
9.45.7.84 unpack_wds_SLQSResetPacketStatics . . . 1426
9.45.7.85 unpack_wds_SLQSSet3GPPConfigItem . . . 1426
9.45.7.86 unpack_wds_SLQSSetIPFamilyPreference . . . 1426

9.45.7.87 unpack_wds_SLQSSetPacketSrvStatusCallback . . . . . 1427
9.45.7.88 unpack_wds_SLQSSetWdsEventCallback . . . . . 1427
9.45.7.89 unpack_wds_SLQSSetWdsEventCallback_ind . . . . . 1427
9.45.7.90 unpack_wds_SLQSSGetDHCPv4ClientConfig . . . . . 1428
9.45.7.91 unpack_wds_SLQSSGetLoopback . . . . . 1428
9.45.7.92 unpack_wds_SLQSSSetDHCPv4ClientConfig . . . . . 1428
9.45.7.93 unpack_wds_SLQSSSetLoopback . . . . . 1429
9.45.7.94 unpack_wds_SLQSStartDataSession . . . . . 1429
9.45.7.95 unpack_wds_SLQSStopDataSession . . . . . 1429
9.45.7.96 unpack_wds_SLQSWdsGoActive . . . . . 1430
9.45.7.97 unpack_wds_SLQSWdsGoDormant . . . . . 1430
9.45.7.98 unpack_wds_SLQSWdsSetEventReport . . . . . 1430
9.45.7.99 unpack_wds_SLQSWdsSwiPDPRuntimeSettings . . . . . 1430

**Index** **1432**

# Chapter 1

# Welcome to the Sierra Wireless Linux QMI SDK API Reference Guide

This API reference guide contains information about all the modules, in the Sierra Wireless Linux QMI SDK (SLQS). Use the tabs at the top of the page to navigate the reference guide.

- Modules tab – lists all the service modules and provides a link to the API header file in each module.
- References tab – links to reference material.

## 1.1 Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

## 1.2 Limitation of Liability

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

## 1.3 Patents

Patents This product may contain technology developed by or for Sierra Wireless Inc. This product includes technology licensed from QUALCOMM® 3G. This product is manufactured or sold by Sierra Wireless Inc. or its affiliates

under one or more patents licensed from InterDigital Group.

## 1.4 Copyright

© 2011-2015 Sierra Wireless. All rights reserved.

## 1.5 Trademarks

AirCard® and Heart of the Wireless Machine® are registered trademarks of Sierra Wireless. Watcher® is a trademark of Sierra Wireless, registered in the European Community. Sierra Wireless, the Sierra Wireless logo, the red wave design, and the red-tipped antenna are trademarks of Sierra Wireless. Windows® is a registered trademark of Microsoft Corporation. QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license. Linux is a registered trademark of Linus Torvalds. Other trademarks are the property of the respective owners.

## 1.6 Contact Information

If you have any questions about the Sierra Wireless Linux SDK, contact your Sierra Wireless account manager.

Consult our website for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, and press releases: http://www.sierrawireless.com.

# Chapter 2

# Module Index

## 2.1 Modules

Here is a list of all modules:

QMI pack/unpack (lite-qmi) . . . . . 29
Streaming Download Protocol (lite-fw) . . . . . 30

# Chapter 3

# Namespace Index

## 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Tables . . . . . 31

# Chapter 4

# Data Structure Index

## 4.1 Data Structures

Here are the data structures with brief descriptions:


_litefw_FirmwareInfo_ . . . 33
altSrcInfo_t . . . 34
appStats . . . 35
audio_RXAGCList . . . 38
audio_RXAVCList . . . 39
audio_RXPCMIIRFltr . . . 39
audio_TXAGCList . . . 41
audio_TXPCMIIRFltr . . . 42
CarrierImage_t . . . 43
cat_AlPhaIdentifierTlv . . . 45
cat_commonEventTlv . . . 45
cat_currentCatEvent . . . 46
cat_EndProactiveSessionTlv . . . 47
cat_EventIDDataTlv . . . 47
cat_EventListTlv . . . 47
cat_RefreshTlv . . . 48
cdmaSSInfo . . . 48
connectionStatus . . . 49
crashInfoParams . . . 50
crashInformation . . . 50
currNetworkInfo . . . 51
dms_ActivationStatusTlv . . . 52
dms_OperatingModeTlv . . . 52
dms_PSMActiveTimerIndTlv . . . 53
dms_PSMActiveTimerTlv . . . 54
dms_PSMDurationDueToOOSTlv . . . 54
dms_PSMDurationThresholdTlv . . . 55
dms_PSMEarlyWakeupTimeTlv . . . 55
dms_PSMEnableStateIndTlv . . . 56
dms_PSMEnableStateTlv . . . 56
dms_PSMPeriodicUpdateTimerIndTlv . . . 57
dms_PSMPeriodicUpdateTimerTlv . . . 57
dms_PSMRandomizationWindowTlv . . . 58
dms_TemperatureTlv . . . 58
dms_VoltageTlv . . . 59
DMScustSettingInfo . . . 60
DMScustSettingList . . . 61
DMSgetCustomFeatureV2 . . . 61

DMSgetCustomInput . . . 62
dunchannelRate . . . 62
eriDataparams . . . 63
eTWSPLMNInfoTlv . . . 64
FMSImageElement . . . 64
FMSImageIdElement . . . 65
FMSImageIDEntries . . . 66
FMSImageList . . . 67
FMSPrefImageList . . . 67
hdrSSInfo . . . 68
image_info_t . . . 69
ims_AMRModeInfo . . . 69
ims_AMROctAlgnInfo . . . 70
ims_AMRWBModeInfo . . . 71
ims_AMRWBOctAlgnInfo . . . 71
ims_CSCFPortNameInfo . . . 72
ims_EnabAMRWBInfo . . . 72
ims_EnabSCRAMRInfo . . . 73
ims_EnabSCRAMRWBInfo . . . 73
ims_IMSDomainInfo . . . 74
ims_IMSTestModeInfo . . . 74
ims_MinSessExpInfo . . . 74
ims_PCSCFPortInfo . . . 75
ims_PhCtxtURIInfo . . . 75
ims_RngBkTmrInfo . . . 76
ims_RngTmrInfo . . . 76
ims_RTPRTCPInactTmrDurInfo . . . 77
ims_SessDurInfo . . . 77
ims_SigCompEnInfo . . . 78
ims_SIPPortInfo . . . 78
ims_SIPRegnTmrInfo . . . 79
ims_SMSFmtInfo . . . 79
ims_SMSoIPNwInfo . . . 80
ims_SubscrTmrInfo . . . 80
ims_TmrT1Info . . . 81
ims_TmrT2Info . . . 81
ims_TmrTfInfo . . . 82
imsa_IMSFailErrCodeTlv . . . 82
imsa_IMSRegStatusErrorCodeInfo . . . 83
imsa_IMSRegStatusInfo . . . 83
imsa_NewIMSRegStatusInfo . . . 84
imsa_RatHandoverStatusInfo . . . 84
imsa_SmsRatInfo . . . 85
imsa_SmsSvcStatusInfo . . . 86
imsa_UtRatInfo . . . 86
imsa_UtSvcStatusInfo . . . 87
imsa_VoipRatInfo . . . 88
imsa_VoipSvcStatusInfo . . . 88
imsa_VtRatInfo . . . 89
imsa_VtSvcStatusInfo . . . 89
ipv6AddressInfo . . . 90
LibPackGPRSRequestedQoS . . . 90
LibpackProfile3GPP . . . 91
LibpackProfile3GPP2 . . . 98
LibPackprofile_3GPP . . . 103
LibPackprofile_3GPP2 . . . 109
LibPackQosClassID . . . 114
LibPackTFTIDParams . . . 115

LibPackUMTSQoS . . . 117
LibPackUMTSReqQoSSigInd . . . 120
loc_accelAcceptReady . . . 120
loc_accelTempAcceptReady . . . 121
loc_BdsSV . . . 122
loc_BdsSVInfo . . . 122
loc_CellDb . . . 123
loc_ClkInfo . . . 123
loc_GnssData . . . 125
loc_gpsTime . . . 126
loc_gyroAcceptReady . . . 127
loc_gyroTempAcceptReady . . . 128
loc_IPv4Config . . . 129
loc_IPv4Info . . . 129
loc_IPv6Config . . . 130
loc_IPv6Info . . . 130
loc_LocApplicationInfo . . . 131
loc_precisionDilution . . . 132
loc_satelliteInfo . . . 133
loc_sensorDataUsage . . . 135
loc_SV . . . 136
loc_SVInfo . . . 137
loc_svUsedforFix . . . 137
loc_urlAddr . . . 138
loc_URLAddrInfo . . . 138
lteSSInfo . . . 139
messageModeTlv . . . 140
nas_acqOrderPref . . . 140
nas_AcqOrderPrefTlv . . . 141
nas_ActPilotPNElement . . . 142
nas_AddCDMASysInfo . . . 142
nas_AddSysInfo . . . 143
nas_BandPrefTlv . . . 143
nas_CallBarringSysInfo . . . 145
nas_callBarStatus . . . 145
nas_CDMAChannel . . . 146
nas_CDMAECIOThresh . . . 147
nas_CDMAInfo . . . 148
nas_CDMARSSIThresh . . . 149
nas_CDMASysInfo . . . 149
nas_CDMASysInfoExt . . . 153
nas_cellParams . . . 153
nas_ciotAcqOrderPref . . . 154
nas_CiotAcqOrderPrefTlv . . . 155
nas_CiotLteOpModePrefTlv . . . 156
nas_CommInfo . . . 156
nas_CSGID . . . 158
nas_currentPLMN . . . 159
nas_dataSrvCapabilities . . . 160
nas_DataStatusDetail . . . 161
nas_detailSvcInfo . . . 163
nas_DeviceConfigDetail . . . 164
nas_dirNum . . . 165
nas_DRCParams . . . 165
nas_ecioListElement . . . 166
nas_ECIOThresh . . . 167
nas_EmerModeTlv . . . 167
nas_errorRateListElement . . . 168

nas_GERANInfo . . . 169
nas_geranInstInfo . . . 171
nas_gsmCellInfo . . . 171
nas_GSMRSSIThresh . . . 172
nas_GSMSrvStatusInfo . . . 173
nas_GSMSysInfo . . . 174
nas_GWAcqOrderPrefTlv . . . 177
nas_HDRECIOThresh . . . 177
nas_HDRIOThresh . . . 178
nas_HDRPersonality_Ind_Data . . . 178
nas_HDRRSSIThresh . . . 179
nas_HDRSINRThresh . . . 180
nas_HDRSINRThreshold . . . 180
nas_HDRSysInfo . . . 181
nas_homeSIDNID . . . 183
nas_infoInterFreq . . . 183
nas_IOThresh . . . 185
nas_lteBandPrefExt . . . 185
nas_LTEBandPrefTlv . . . 186
nas_lteEARFCN . . . 188
nas_lteGsmCellInfo . . . 188
nas_LTEInfo . . . 189
nas_LTEInfoInterfreq . . . 192
nas_LTEInfoIntrafreq . . . 192
nas_LTEInfoNeighboringGSM . . . 194
nas_LTEInfoNeighboringWCDMA . . . 195
nas_LteM1BandPrefTlv . . . 196
nas_LteNb1BandPrefTlv . . . 198
nas_LTEOperationMode . . . 200
nas_ltePCI . . . 200
nas_lteRsrpinformation . . . 201
nas_LTERSRPThresh . . . 201
nas_LTERSRQThresh . . . 202
nas_LTERSSIThresh . . . 202
nas_LTESigRptCfg . . . 203
nas_LTESigRptConfig . . . 204
nas_lteSnrinformation . . . 204
nas_LTESNRThresh . . . 205
nas_LTESNRThreshold . . . 205
nas_LTESysInfo . . . 206
nas_lteWcdmaCellInfo . . . 208
nas_minBasedIMSI . . . 209
nas_MNRInfo . . . 210
nas_ModePrefTlv . . . 211
nas_namName . . . 212
nas_netSelectionPref . . . 212
nas_NetSelPrefTlv . . . 213
nas_NetworkStat1x . . . 213
nas_NetworkStatEVDO . . . 216
nas_nmrCellInfo . . . 217
nas_nr5gBandPref . . . 218
nas_NR5gBandPrefTlv . . . 219
nas_operatorNameString . . . 220
nas_OperatorPLMNData . . . 220
nas_operatorPLMNList . . . 221
nas_PhyCaAggPcellInfo . . . 222
nas_PhyCaAggScellDlBw . . . 223
nas_PhyCaAggScellIndex . . . 223

nas_PhyCaAggScellIndType . . . 224
nas_PhyCaAggScellInfo . . . 225
nas_PilotSetData . . . 227
nas_PilotSetParams . . . 228
nas_PLMNNetworkName . . . 228
nas_PLMNNetworkNameData . . . 229
nas_PRLPrefTlv . . . 230
nas_protocolSubtypeElement . . . 231
nas_qaQmi3Gpp2TimeZone . . . 233
nas_QmiNas3GppNetworkInfo . . . 233
nas_QmiNas3GppNetworkRAT . . . 235
nas_QmisNasPcsDigit . . . 235
nas_RankIndicatorTlv . . . 236
nas_RatDisabledMaskTlv . . . 236
nas_RejectReasonTlv . . . 237
nas_RFBandInfoElements . . . 238
nas_RfBandInfoExtFormat . . . 238
nas_RfBandInfoExtFormatElements . . . 239
nas_RfBandInfoList . . . 240
nas_RfBandwidthInfo . . . 240
nas_RfBandwidthInfoElements . . . 241
nas_RfDedicatedBandInfo . . . 242
nas_RfDedicatedBandInfoElements . . . 242
nas_RFInfoTlv . . . 243
nas_roamIndList . . . 244
nas_RoamPrefTlv . . . 245
nas_RSRPThresh . . . 245
nas_rsrqInformation . . . 246
nas_RSRQThresh . . . 246
nas_RSSIThresh . . . 247
nas_rxInfo . . . 248
nas_RxSigInfo . . . 249
nas_rxSignalStrengthListElement . . . 250
nas_SccRxInfo . . . 251
nas_serviceProviderName . . . 252
nas_servSystem . . . 253
nas_sidNid . . . 254
nas_SignalStrengthTlv . . . 255
nas_SLQSSignalStrengthsIndReq . . . 255
nas_SLQSSignalStrengthsInformation . . . 257
nas_SLQSSignalStrengthsTlv . . . 258
nas_SrvDomainPrefTlv . . . 259
nas_SrvStatusInfo . . . 259
nas_sysInfoCommon . . . 260
nas_TDSCDMAECIOThresh . . . 263
nas_TDSCDMARSCPThresh . . . 263
nas_TDSCDMARSSIThresh . . . 264
nas_TDSCDMASINRCONFThresh . . . 264
nas_TDSCDMASINRThresh . . . 265
nas_timeInfo . . . 265
nas_trueIMSI . . . 267
nas_txInfo . . . 268
nas_UMTSInfo . . . 268
nas_UMTSinstInfo . . . 270
nas_umtsLTENbrCell . . . 271
nas_UniversalTime . . . 272
nas_wcdmaCellInfo . . . 273
nas_WCDMAECIOThresh . . . 274

nas_WCDMAInfoLTENeighborCell . . . 274
nas_WCDMARSSIThresh . . . 275
nas_WCDMASysInfo . . . 276
nas_wcdmaUARFCN . . . 279
NASBandPreferenceTlv . . . 280
NASEmergencyModeTlv . . . 280
NasGetLTECphyCaInfo . . . 280
NASGWAcqOrderPrefTlv . . . 281
NASLTEBandPreferenceTlv . . . 282
NASLteNasReleaseInfoTlv . . . 282
NASModePreferenceTlv . . . 283
NASNetSelPreferenceTlv . . . 283
NASOTAMessageTlv . . . 284
NASPhyCaAggPcellInfo . . . 285
NASPhyCaAggScellDlBw . . . 286
NASPhyCaAggScellIndex . . . 286
NASPhyCaAggScellIndType . . . 287
NASPhyCaAggScellInfo . . . 287
NASPRLPreferenceTlv . . . 288
NASQmiCbkNasSwiOTAMessageInd . . . 289
NASQmiCbkNasSystemSelPrefInd . . . 289
NASRoamPreferenceTlv . . . 291
NASServDomainPrefTlv . . . 291
NASServingSystemInfo . . . 292
NASTimeInfoTlv . . . 293
newMTMessageTlv . . . 294
pack_audio_SLQSGetAudioPathConfig_t . . . 294
pack_audio_SLQSGetAudioProfile_t . . . 295
pack_audio_SLQSGetAudioVolTLBConfig_t . . . 296
pack_audio_SLQSSetAudioPathConfig_t . . . 296
pack_audio_SLQSSetAudioProfile_t . . . 298
pack_audio_SLQSSetAudioVolTLBConfig_t . . . 300
pack_cat_CATSendEnvelopeCommand_t . . . 300
pack_cat_CATSendTerminalResponse_t . . . 301
pack_cat_SetCATEventCallback_t . . . 302
pack_dms_ActivateAutomatic_t . . . 303
pack_dms_GetCustFeaturesV2_t . . . 303
pack_dms_ResetToFactoryDefaults_t . . . 304
pack_dms_SetActivationStatusCallback_t . . . 304
pack_dms_SetCrashAction_t . . . 304
pack_dms_SetCustFeature_t . . . 305
pack_dms_SetCustFeaturesV2_t . . . 307
pack_dms_SetEventReport_t . . . 308
pack_dms_SetIndicationRegister_t . . . 308
pack_dms_SetPower_t . . . 309
pack_dms_SetUSBComp_t . . . 310
pack_dms_SLQSDmsSwiIndicationRegister_t . . . 311
pack_dms_SLQSSetPowerSaveModeConfig_t . . . 311
pack_dms_SLQSSwiGetCrashInfo_t . . . 312
pack_dms_SLQSSwiSetDyingGaspCfg_t . . . 312
pack_dms_SLQSSwiSetHostDevInfo_t . . . 313
pack_dms_SLQSSwiSetOSInfo_t . . . 314
pack_dms_SwiSetEventReport_t . . . 314
pack_dms_UIMChangePIN_t . . . 315
pack_dms_UIMGetControlKeyStatus_t . . . 315
pack_dms_UIMGetICCID_t . . . 316
pack_dms_UIMSetControlKeyProtection_t . . . 316
pack_dms_UIMSetPINProtection_t . . . 317

pack_dms_UIMUnblockControlKey_t . . . 318
pack_dms_UIMUnblockPIN_t . . . 318
pack_dms_UIMVerifyPIN_t . . . 319
pack_fms_GetImagesPreference_t . . . 320
pack_fms_GetStoredImages_t . . . 320
pack_fms_SetImagesPreference_t . . . 320
pack_ims_SLQSImsConfigIndicationRegister_t . . . 321
pack_ims_SLQSSetIMSSMSConfig_t . . . 323
pack_ims_SLQSSetIMSUserConfig_t . . . 323
pack_ims_SLQSSetIMSVoIPConfig_t . . . 324
pack_ims_SLQSSetRegMgrConfig_t . . . 326
pack_ims_SLQSSetSIPConfig_t . . . 327
pack_imsa_SLQSRegisterIMSAIndication_t . . . 328
pack_loc_Delete_Assist_Data_t . . . 329
pack_loc_EventRegister_t . . . 330
pack_loc_SetExtPowerState_t . . . 333
pack_loc_SetOperationMode_t . . . 333
pack_loc_SLQSLOCGetBestAvailPos_t . . . 334
pack_loc_SLQSLOCGetServer_t . . . 335
pack_loc_SLQSLOCInjectPosition_t . . . 335
pack_loc_SLQSLOCInjectSensorData_t . . . 340
pack_loc_SLQSLOCInjectUTCTime_t . . . 342
pack_loc_SLQSLOCSetCradleMountConfig_t . . . 342
pack_loc_SLQSLOCSetServer_t . . . 343
pack_loc_Start_t . . . 344
pack_loc_Stop_t . . . 346
pack_nas_InitiateDomainAttach_t . . . 346
pack_nas_SetACCOLC_t . . . 347
pack_nas_SetCDMANetworkParameters_t . . . 347
pack_nas_SetNetworkPreference_t . . . 349
pack_nas_SLQSConfigSigInfo_t . . . 349
pack_nas_SLQSGetPLMNName_t . . . 351
pack_nas_SLQSInitiateNetworkRegistration_t . . . 352
pack_nas_SLQSNasConfigSigInfo2_t . . . 353
pack_nas_SLQSNasGet3GPP2Subscription_t . . . 357
pack_nas_SLQSNasGetTxRxInfo_t . . . 358
pack_nas_SLQSNasIndicationRegisterExt_t . . . 358
pack_nas_SLQSNASSeteDRXParams_t . . . 361
pack_nas_SLQSNasSwiIndicationRegister_t . . . 362
pack_nas_SLQSNASSwiSetChannelLock_t . . . 363
pack_nas_SLQSSetBandPreference_t . . . 364
pack_nas_SLQSSetSignalStrengthsCallback_t . . . 366
pack_nas_SLQSSetSysSelectionPref_t . . . 366
pack_nas_SLQSSetSysSelectionPrefExt_t . . . 372
pack_nas_SLQSSwiPSDetach_t . . . 382
pack_pds_PDSInjectTimeReference_t . . . 382
pack_pds_ResetPDSData_t . . . 383
pack_pds_SetEventReportCallback_t . . . 384
pack_pds_SetPDSDefaults_t . . . 384
pack_pds_SetPDSState_t . . . 385
pack_pds_SetPortAutomaticTracking_t . . . 385
pack_pds_SetServiceAutomaticTracking_t . . . 386
pack_pds_SetXTRAAutomaticDownload_t . . . 386
pack_pds_SetXTRANetwork_t . . . 387
pack_pds_SLQSGetAGPSConfig_t . . . 387
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t . . . 388
pack_pds_SLQSPDSInjectPositionData_t . . . 389
pack_pds_SLQSSetAGPSConfig_t . . . 391

pack_pds_SLQSSetPositionMethodState_t . . . 391
pack_pds_StartPDSTrackingSessionExt_t . . . 392
pack_qmi_t . . . 393
pack_qos_BindDataPort_t . . . 394
pack_qos_SLQSQosSwiReadApnExtraParams_t . . . 395
pack_qos_SLQSQosSwiReadDataStats_t . . . 395
pack_qos_SLQSSetQosEventCallback_t . . . 395
pack_rms_SetSMSWake_t . . . 396
pack_sar_SLQSSetRfSarState_t . . . 396
pack_sms_SaveSMS_t . . . 397
pack_sms_SendSMS_t . . . 398
pack_sms_SetNewSMSCallback_t . . . 399
pack_sms_SetSMSCAddress_t . . . 399
pack_sms_SLQSDeleteSMS_t . . . 399
pack_sms_SLQSGetSMS_t . . . 400
pack_sms_SLQSGetSmsBroadcastConfig_t . . . 401
pack_sms_SLQSGetSMSList_t . . . 401
pack_sms_SLQSModifySMSStatus_t . . . 402
pack_sms_SLQSSendAsyncSMS_t . . . 403
pack_sms_SLQSSetIndicationRegister_t . . . 403
pack_sms_SLQSSetSmsBroadcastActivation_t . . . 404
pack_sms_SLQSSetSmsBroadcastConfig_t . . . 404
pack_sms_SLQSSetSmsStorage_t . . . 405
pack_sms_SLQSSmsGetMaxStorageSize_t . . . 406
pack_sms_SLQSSmsSetRoutes_t . . . 406
pack_swiaudio_SLQSGetM2MAudioProfile_t . . . 406
pack_swiaudio_SLQSGetM2MAudioVolume_t . . . 407
pack_swiaudio_SLQSGetM2MAVMute_t . . . 407
pack_swiaudio_SLQSGetM2MSpkrGain_t . . . 408
pack_swiaudio_SLQSSetM2MAudioAVCFG_t . . . 408
pack_swiaudio_SLQSSetM2MAudioLPBK_t . . . 409
pack_swiaudio_SLQSSetM2MAudioProfile_t . . . 409
pack_swiaudio_SLQSSetM2MAudioVolume_t . . . 411
pack_swiaudio_SLQSSetM2MAVMute_t . . . 411
pack_swiaudio_SLQSSetM2MSpkrGain_t . . . 412
pack_swiavms_SLQSAVMSSendSelection_t . . . 413
pack_swiavms_SLQSAVMSSetSettings_t . . . 413
pack_swiavms_SLQSAVMSSetSettings_v2_t . . . 415
pack_swiavms_SLQSAVMSStartSession_t . . . 417
pack_swiavms_SLQSAVMSStopSession_t . . . 417
pack_swidms_SLQSSwiDmsSetMTU_t . . . 418
pack_swidms_SLQSSwiDmsSetUsbNetNum_t . . . 418
pack_swiloc_SwiLocSetAutoStart_t . . . 419
pack_swioma_SLQSOMADMCancelSession_t . . . 420
pack_swioma_SLQSOMADMCancelSessionExt_t . . . 421
pack_swioma_SLQSOMADMGetSessionInfo_t . . . 421
pack_swioma_SLQSOMADMSendSelection_t . . . 422
pack_swioma_SLQSOMADMSendSelectionExt_t . . . 422
pack_swioma_SLQSOMADMSetSettings_t . . . 423
pack_swioma_SLQSOMADMSetSettingsExt_t . . . 424
pack_swioma_SLQSOMADMStartSession_t . . . 425
pack_swioma_SLQSOMADMStartSessionExt_t . . . 426
pack_tmd_SLQSTmdDeRegNotMitigationLvl_t . . . 426
pack_tmd_SLQSTmdGetMitigationLvl_t . . . 427
pack_tmd_SLQSTmdRegNotMitigationLvl_t . . . 427
pack_uim_ChangePin_t . . . 428
pack_uim_ReadTransparent_t . . . 429
pack_uim_SetPinProtection_t . . . 430

pack_uim_SLQSUIMAuthenticate_t . . . 431
pack_uim_SLQSUIMDepersonalization_t . . . 431
pack_uim_SLQSUIMEventRegister_t . . . 432
pack_uim_SLQSUIMGetConfiguration_t . . . 432
pack_uim_SLQSUIMGetFileAttributes_t . . . 433
pack_uim_SLQSUIMPowerDown_t . . . 434
pack_uim_SLQSUIMPowerUp_t . . . 434
pack_uim_SLQSUIMRefreshComplete_t . . . 435
pack_uim_SLQSUIMRefreshGetLastEvent_t . . . 435
pack_uim_SLQSUIMRefreshOK_t . . . 436
pack_uim_SLQSUIMRefreshRegister_t . . . 436
pack_uim_SLQSUIMSwitchSlot_t . . . 437
pack_uim_UnblockPin_t . . . 437
pack_uim_VerifyPin_t . . . 438
pack_voice_AnswerUSSD_t . . . 439
pack_voice_OriginateUSSD_t . . . 440
pack_voice_SLQSOriginateUSSD_t . . . 440
pack_voice_SLQSVoiceALSSelectLine_t . . . 441
pack_voice_SLQSVoiceALSSetLineSwitching_t . . . 441
pack_voice_SLQSVoiceAnswerCall_t . . . 442
pack_voice_SLQSVoiceBindSubscription_t . . . 442
pack_voice_SLQSVoiceBurstDTMF_t . . . 443
pack_voice_SLQSVoiceDialCall_t . . . 443
pack_voice_SLQSVoiceEndCall_t . . . 445
pack_voice_SLQSVoiceGetCallBarring_t . . . 445
pack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 446
pack_voice_SLQSVoiceGetCallInfo_t . . . 447
pack_voice_SLQSVoiceGetCallWaiting_t . . . 447
pack_voice_SLQSVoiceGetConfig_t . . . 448
pack_voice_SLQSVoiceIndicationRegister_t . . . 450
pack_voice_SLQSVoiceManageCalls_t . . . 450
pack_voice_SLQSVoiceOrigUSSDNoWait_t . . . 451
pack_voice_SLQSVoiceSendFlash_t . . . 452
pack_voice_SLQSVoiceSetCallBarringPassword_t . . . 452
pack_voice_SLQSVoiceSetConfig_t . . . 453
pack_voice_SLQSVoiceSetPreferredPrivacy_t . . . 455
pack_voice_SLQSVoiceSetSUPSService_t . . . 455
pack_voice_SLQSVoiceStartContDTMF_t . . . 457
pack_voice_SLQSVoiceStopContDTMF_t . . . 458
pack_wds_DHCPv4ClientLeaseChange_t . . . 458
pack_wds_GetDefaultProfile_t . . . 459
pack_wds_GetDefaultProfileNum_t . . . 459
pack_wds_GetDormancyState_t . . . 460
pack_wds_GetLastMobileIPError_t . . . 460
pack_wds_GetMobileIP_t . . . 460
pack_wds_GetMobileIPProfile_t . . . 460
pack_wds_GetPacketStatistics_t . . . 461
pack_wds_GetPacketStatus_t . . . 461
pack_wds_GetSessionDuration_t . . . 462
pack_wds_RMSetTransferStatistics_t . . . 462
pack_wds_SetAutoconnect_t . . . 463
pack_wds_SetDefaultProfile_t . . . 463
pack_wds_SetDefaultProfileNum_t . . . 465
pack_wds_SetMobileIP_t . . . 466
pack_wds_SetMobileIPParameters_t . . . 466
pack_wds_SetMobileIPProfile_t . . . 467
pack_wds_SLQSCreateProfile_t . . . 469
pack_wds_SLQSDeleteProfile_t . . . 470

pack_wds_SLQSGetCurrDataSystemStat_t . . . 470
pack_wds_SLQSGetDataBearerTechnology_t . . . 471
pack_wds_SLQSGetDUNCallInfo_t . . . 471
pack_wds_SLQSGetProfileSettings_t . . . 472
pack_wds_SLQSGetRuntimeSettings_t . . . 473
pack_wds_SLQSModifyProfile_t . . . 474
pack_wds_SLQSSet3GPPConfigItem_t . . . 475
pack_wds_SLQSSetIPFamilyPreference_t . . . 477
pack_wds_SLQSSetWdsEventCallback_t . . . 477
pack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 478
pack_wds_SLQSSSetDHCPv4ClientConfig_t . . . 478
pack_wds_SLQSSSetLoopback_t . . . 479
pack_wds_SLQSStartDataSession_t . . . 479
pack_wds_SLQSStopDataSession_t . . . 481
pack_wds_SLQSWdsSetEventReport_t . . . 481
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 483
PackCreateProfileOut . . . 483
packgetDyingGaspCfg . . . 484
packgetDyingGaspStatistics . . . 484
PackSwiAvmsSetSettingsAPNInfo . . . 485
PackSwiAvmsSetSettingsConnectionRetryTimers . . . 486
PackSwiAvmsSetSettingsPeriodInfo . . . 487
PackSwiAVMSSettingsAPNInfo . . . 487
PackSwiAVMSSettingsConnectionRetryTimers . . . 488
PackSwiAVMSSettingsPeriodsInfo . . . 488
qmiSmsMessageList . . . 489
qmiWDSDataBearerTechnology . . . 489
qos_BindDataPortMuxID_t . . . 490
qos_BindDataPortPeripheralEndPointID_t . . . 490
qos_BindDataPortSIODataPort_t . . . 491
RFBandInfoElements . . . 491
rmTrasnferStaticsReq . . . 492
sensorData_t . . . 493
slot_t . . . 494
slotInf . . . 495
slots_t . . . 497
sms_BroadcastConfig . . . 497
sms_CDMABroadcastConfig . . . 498
sms_getIndicationReg . . . 498
sms_getMsgWaitingInfo . . . 499
sms_getTransLayerInfo . . . 500
sms_getTransNWRegInfo . . . 500
sms_maxStorageSizeReq . . . 501
sms_maxStorageSizeResp . . . 502
sms_messageWaitingInfoContent . . . 502
sms_msgProtocolResp . . . 503
sms_qaQmi3GPP2BroadcastCfgInfo . . . 503
sms_qaQmi3GPPBroadcastCfgInfo . . . 504
sms_routeEntry . . . 505
sms_sendAsyncsmsParams . . . 506
sms_setIndicationReg . . . 508
sms_setRoutesReq . . . 509
sms_transLayerInfo . . . 510
sMSCAddressInfo . . . 511
sMSCAddressTlv . . . 511
sMSEtwsMessageInfo . . . 512
sMSEtwsMessageTlv . . . 512
sMSEtwsPlmnInfo . . . 513

sMSMessageModeInfo . . . 513
sMSMTMessageInfo . . . 513
sMSOnIMSInfo . . . 514
sMSOnIMSTlv . . . 514
sMSTransferRouteMTMessageInfo . . . 515
swi_uint256_t . . . 516
swiaudio_PCMparams . . . 516
swidms_ehrpdMTUSizeTlv . . . 516
swidms_hrpdMTUSizeTlv . . . 517
swidms_mtuSize3gppTlv . . . 517
swidms_usbMTUSizeTlv . . . 518
tdscdmaSigInfoExt . . . 518
tempData_t . . . 519
tmd_mitigationDevList . . . 520
transferRouteMessageTlv . . . 521
uim_appStatus . . . 521
uim_authenticateResult . . . 524
uim_authenticationData . . . 525
uim_cardResult . . . 526
uim_cardStatus . . . 527
uim_changeUIMPIN . . . 528
uim_depersonalizationInformation . . . 529
uim_encryptedPIN1 . . . 530
uim_fileAttributes . . . 531
uim_fileInfo . . . 534
uim_hotSwapStatus . . . 535
uim_personalizationStatus . . . 535
uim_readResult . . . 536
uim_readTransparentInfo . . . 537
uim_refreshevent . . . 537
uim_registerRefresh . . . 539
uim_remainingRetries . . . 540
uim_sessionInformation . . . 541
uim_setPINProtection . . . 541
uim_slotInfo . . . 542
uim_UIMSessionInformation . . . 544
uim_unblockUIMPIN . . . 545
uim_verifyUIMPIN . . . 546
unpack_audio_SLQSGetAudioPathConfig_t . . . 546
unpack_audio_SLQSGetAudioProfile_t . . . 549
unpack_audio_SLQSGetAudioVolTLBConfig_t . . . 550
unpack_audio_SLQSSetAudioVolTLBConfig_t . . . 550
unpack_cat_SetCatEventCallback_ind_t . . . 551
unpack_cat_SetCATEventCallback_t . . . 551
unpack_dms_GetActivationState_t . . . 552
unpack_dms_GetBandCapability_t . . . 553
unpack_dms_GetCrashAction_t . . . 555
unpack_dms_GetCustFeature_t . . . 556
unpack_dms_GetCustFeaturesV2_t . . . 558
unpack_dms_GetDeviceCap_t . . . 559
unpack_dms_GetDeviceCapabilities_t . . . 560
unpack_dms_GetDeviceHardwareRev_t . . . 562
unpack_dms_GetDeviceMfr_t . . . 563
unpack_dms_GetDeviceSerialNumbers_t . . . 563
unpack_dms_GetFirmwareInfo_t . . . 565
unpack_dms_GetFirmwareRevision_t . . . 567
unpack_dms_GetFirmwareRevisions_t . . . 568
unpack_dms_GetFSN_t . . . 569

unpack_dms_GetHardwareRevision_t . . . 570
unpack_dms_GetIMSI_t . . . 570
unpack_dms_GetManufacturer_t . . . 571
unpack_dms_GetModelID_t . . . 572
unpack_dms_GetNetworkTime_t . . . 572
unpack_dms_GetOfflineReason_t . . . 573
unpack_dms_GetPower_t . . . 574
unpack_dms_GetPRLVersion_t . . . 575
unpack_dms_GetSerialNumbers_t . . . 576
unpack_dms_GetUSBComp_t . . . 577
unpack_dms_GetVoiceNumber_t . . . 579
unpack_dms_PSMCfgChange_ind_t . . . 581
unpack_dms_ResetToFactoryDefaults_t . . . 581
unpack_dms_SetActivationStatusCallback_t . . . 582
unpack_dms_SetCrashAction_t . . . 582
unpack_dms_SetCustFeature_t . . . 583
unpack_dms_SetCustFeaturesV2_t . . . 583
unpack_dms_SetEventReport_ind_t . . . 584
unpack_dms_SetEventReport_t . . . 585
unpack_dms_SetFirmwarePreference_t . . . 585
unpack_dms_SetIndicationRegister_t . . . 586
unpack_dms_SetPower_t . . . 586
unpack_dms_SetUSBComp_t . . . 587
unpack_dms_SLQSDmsSwiGetPCInfo_t . . . 587
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t . . . 590
unpack_dms_SLQSDmsSwiGetResetInfo_t . . . 591
unpack_dms_SLQSDmsSwiIndicationRegister_t . . . 592
unpack_dms_SLQSGetBandCapability_t . . . 593
unpack_dms_SLQSGetERIFile_t . . . 597
unpack_dms_SLQSGetPowerSaveModeConfig_t . . . 598
unpack_dms_SLQSSetPowerSaveModeConfig_t . . . 599
unpack_dms_SLQSSwiClearDyingGaspStatistics_t . . . 600
unpack_dms_SLQSSwiGetCrashInfo_t . . . 600
unpack_dms_SLQSSwiGetDyingGaspCfg_t . . . 601
unpack_dms_SLQSSwiGetDyingGaspStatistics_t . . . 602
unpack_dms_SLQSSwiGetFirmwareCurr_t . . . 602
unpack_dms_SLQSSwiGetFwUpdateStatus_t . . . 603
unpack_dms_SLQSSwiGetHostDevInfo_t . . . 605
unpack_dms_SLQSSwiGetOSInfo_t . . . 606
unpack_dms_SLQSSwiGetSerialNoExt_t . . . 607
unpack_dms_SLQSSwiSetDyingGaspCfg_t . . . 608
unpack_dms_SLQSSwiSetHostDevInfo_t . . . 608
unpack_dms_SLQSSwiSetOSInfo_t . . . 609
unpack_dms_SLQSUIMGetState_t . . . 609
unpack_dms_SwiEventReportCallBack_ind_t . . . 610
unpack_dms_SwiSetEventReport_t . . . 610
unpack_dms_UIMGetControlKeyStatus_t . . . 611
unpack_dms_UIMGetICCID_t . . . 612
unpack_dms_UIMGetPINStatus_t . . . 613
unpack_dms_UIMSetControlKeyProtection_t . . . 614
unpack_dms_UIMSetPINProtection_t . . . 615
unpack_dms_UIMUnblockControlKey_t . . . 616
unpack_fms_GetImagesPreference_t . . . 617
unpack_fms_GetStoredImages_t . . . 617
unpack_fms_SetImagesPreference_t . . . 618
unpack_ims_SLQSGetIMSSMSConfig_t . . . 619
unpack_ims_SLQSGetIMSUserConfig_t . . . 620
unpack_ims_SLQSGetIMSVoIPConfig_t . . . 621

unpack_ims_SLQSGetRegMgrConfig_t . . . 623
unpack_ims_SLQSGetSIPConfig_t . . . 624
unpack_ims_SLQSRegMgrCfgCallBack_ind_t . . . 626
unpack_ims_SLQSSetIMSSMSConfig_t . . . 627
unpack_ims_SLQSSetIMSUserConfig_t . . . 627
unpack_ims_SLQSSetIMSVoIPConfig_t . . . 628
unpack_ims_SLQSSetRegMgrConfig_t . . . 628
unpack_ims_SLQSSetSIPConfig_t . . . 629
unpack_ims_SLQSSIPCfgCallBack_ind_t . . . 629
unpack_ims_SLQSSMSCfgCallBack_ind_t . . . 631
unpack_ims_SLQSUserCfgCallBack_ind_t . . . 631
unpack_ims_SLQSVoIPCfgCallBack_ind_t . . . 632
unpack_imsa_SLQSGetIMSARegStatus_t . . . 634
unpack_imsa_SLQSGetIMSAServiceStatus_t . . . 635
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t . . . 638
unpack_imsa_SLQSImsaRatStatusCallBack_ind_t . . . 638
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t . . . 639
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t . . . 640
unpack_loc_BestAvailPos_Ind_t . . . 641
unpack_loc_CradleMountCallback_Ind_t . . . 648
unpack_loc_Delete_Assist_Data_t . . . 648
unpack_loc_DeleteAssistData_Ind_t . . . 649
unpack_loc_EngineState_Ind_t . . . 650
unpack_loc_EventNMEA_Ind_t . . . 651
unpack_loc_EventRegister_t . . . 652
unpack_loc_EventTimeSyncCallback_Ind_t . . . 652
unpack_loc_GetServer_Ind_t . . . 652
unpack_loc_GnssSvInfo_Ind_t . . . 654
unpack_loc_InjectPositionCallback_Ind_t . . . 655
unpack_loc_InjectSensorDataCallback_Ind_t . . . 655
unpack_loc_InjectTimeSyncDataCallback_Ind_t . . . 657
unpack_loc_InjectUTCTimeCallback_Ind_t . . . 658
unpack_loc_PositionRpt_Ind_t . . . 658
unpack_loc_SensorStreamingCallback_Ind_t . . . 665
unpack_loc_SetExtPowerConfig_Ind_t . . . 666
unpack_loc_SetExtPowerState_t . . . 667
unpack_loc_SetOperationMode_Ind_t . . . 668
unpack_loc_SetOperationMode_t . . . 668
unpack_loc_SetServer_Ind_t . . . 669
unpack_loc_SLQSLOCGetBestAvailPos_t . . . 670
unpack_loc_Start_t . . . 670
unpack_loc_Stop_t . . . 670
unpack_nas_GetACCOLC_t . . . 671
unpack_nas_GetANAAAAuthenticationStatus_t . . . 671
unpack_nas_GetCDMANetworkParameters_t . . . 672
unpack_nas_GetHomeNetwork3GPP2_t . . . 674
unpack_nas_GetHomeNetwork_t . . . 677
unpack_nas_GetNetworkPreference_t . . . 678
unpack_nas_GetRFInfo_t . . . 679
unpack_nas_GetServingNetwork_t . . . 680
unpack_nas_GetServingNetworkCapabilities_t . . . 683
unpack_nas_GetSignalStrengths_t . . . 684
unpack_nas_PerformNetworkScan_t . . . 685
unpack_nas_SetDataCapabilitiesCallback_ind_t . . . 686
unpack_nas_SetEventReportInd_t . . . 687
unpack_nas_SetNasLTECphyCaIndCallback_ind_t . . . 688
unpack_nas_SetNetworkPreference_t . . . 689
unpack_nas_SetRoamingIndicatorCallback_ind_t . . . 689

unpack_nas_SetServingSystemCallback_ind_t . . . 690
unpack_nas_SLQSGetErrorRate_t . . . 691
unpack_nas_SlqsGetLTECphyCAInfo_t . . . 692
unpack_nas_SLQSGetNetworkTime_t . . . 692
unpack_nas_SLQSGetOperatorNameData_t . . . 693
unpack_nas_SLQSGetPLMNName_t . . . 694
unpack_nas_SLQSGetServingSystem_t . . . 698
unpack_nas_SLQSGetSignalStrength_t . . . 702
unpack_nas_SLQSGetSysInfo_t . . . 705
unpack_nas_SLQSGetSysSelectionPref_t . . . 708
unpack_nas_SLQSGetSysSelectionPrefExt_t . . . 713
unpack_nas_SLQSNasGet3GPP2Subscription_t . . . 716
unpack_nas_SLQSNasGetCellLocationInfo_t . . . 717
unpack_nas_SLQSNASGeteDRXParams_t . . . 719
unpack_nas_SLQSNasGetHDRColorCode_t . . . 720
unpack_nas_SLQSNasGetRFInfo_t . . . 720
unpack_nas_SLQSNasGetSigInfo_t . . . 721
unpack_nas_SLQSNasGetTxRxInfo_t . . . 722
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t . . . 723
unpack_nas_SLQSNasSigInfoCallback_ind_t . . . 724
unpack_nas_SLQSNASSwiGetChannelLock_t . . . 725
unpack_nas_SLQSNasSwiModemStatus_t . . . 726
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t . . . 727
unpack_nas_SLQSNasTimerCallback_ind_t . . . 728
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t . . . 728
unpack_nas_SLQSSwiGetHDRPersonality_t . . . 729
unpack_nas_SLQSSwiGetHDRProtSubtype_t . . . 730
unpack_nas_SLQSSwiGetHRPDStats_t . . . 731
unpack_nas_SLQSSwiGetLteCQI_t . . . 731
unpack_nas_SLQSSwiGetLteSccRxInfo_t . . . 733
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t . . . 733
unpack_nas_SLQSSwiNetworkDebug_t . . . 734
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t . . . 735
unpack_nas_SLQSSysInfoCallback_ind_t . . . 735
unpack_omaDmConfigTlv_t . . . 739
unpack_omaDmFotaTlv_t . . . 739
unpack_omaDmNotificationsTlv_t . . . 742
unpack_pds_GetPDSDefaults_t . . . 742
unpack_pds_GetPDSState_t . . . 743
unpack_pds_GetPortAutomaticTracking_t . . . 744
unpack_pds_GetServiceAutomaticTracking_t . . . 745
unpack_pds_GetXTRAAutomaticDownload_t . . . 745
unpack_pds_GetXTRANetwork_t . . . 746
unpack_pds_GetXTRAValidity_t . . . 747
unpack_pds_SetEventReport_Ind_t . . . 747
unpack_pds_SetPdsState_Ind_t . . . 749
unpack_pds_SLQSGetAGPSConfig_t . . . 750
unpack_pds_SLQSGetGPSStateInfo_t . . . 751
unpack_qmi_t . . . 756
unpack_qos_BindDataPort_t . . . 756
unpack_qos_dataRate_t . . . 757
unpack_qos_IPv4Addr_t . . . 757
unpack_qos_IPv6Addr_t . . . 757
unpack_qos_IPv6TrafCls_t . . . 758
unpack_qos_pktErrRate_t . . . 758
unpack_qos_Port_t . . . 759
unpack_qos_QosFlowInfo_t . . . 759
unpack_qos_QosFlowInfoState_t . . . 761

unpack_qos_SLQSQosGetNetworkStatus_t . . . 761
unpack_qos_SLQSQosSwiReadApnExtraParams_t . . . 762
unpack_qos_SLQSQosSwiReadDataStats_t . . . 764
unpack_qos_SLQSSetQosEventCallback_ind_t . . . 765
unpack_qos_SLQSSetQosEventCallback_t . . . 766
unpack_qos_SLQSSetQosNWStatusCallback_ind_t . . . 766
unpack_qos_SLQSSetQosPriEventCallback_ind_t . . . 767
unpack_qos_SLQSSetQosStatusCallback_ind_t . . . 767
unpack_qos_swiQosFilter_t . . . 769
unpack_qos_swiQosFlow_t . . . 772
unpack_qos_tokenBucket_t . . . 776
unpack_qos_Tos_t . . . 777
unpack_QosFlowStat_t . . . 777
unpack_result_t . . . 778
unpack_rms_GetSMSWake_t . . . 779
unpack_rms_SetSMSWake_t . . . 779
unpack_RMTransferStatistics_ind_t . . . 780
unpack_sar_SLQSGetRfSarState_t . . . 781
unpack_sms_GetSMSCAddress_t . . . 781
unpack_sms_SaveSMS_t . . . 782
unpack_sms_SendSMS_t . . . 783
unpack_sms_SetNewSMSCallback_ind_t . . . 783
unpack_sms_SetNewSMSCallback_t . . . 785
unpack_sms_SetSMSCAddress_t . . . 785
unpack_sms_SLQSDeleteSMS_t . . . 786
unpack_sms_SLQSGetIndicationRegister_t . . . 786
unpack_sms_SLQSGetMessageWaiting_t . . . 787
unpack_sms_SLQSGetSMS_t . . . 787
unpack_sms_SLQSGetSmsBroadcastConfig_t . . . 788
unpack_sms_SLQSGetSMSList_t . . . 789
unpack_sms_SLQSGetTransLayerInfo_t . . . 790
unpack_sms_SLQSGetTransNWRegInfo_t . . . 790
unpack_sms_SLQSModifySMSStatus_t . . . 791
unpack_sms_SLQSNWRegInfoCallback_ind_t . . . 791
unpack_sms_SLQSSendAsyncSMS_t . . . 792
unpack_sms_SLQSSetIndicationRegister_t . . . 792
unpack_sms_SLQSSetSmsBroadcastActivation_t . . . 793
unpack_sms_SLQSSetSmsBroadcastConfig_t . . . 793
unpack_sms_SLQSSetSmsStorage_t . . . 794
unpack_sms_SLQSSmsGetMaxStorageSize_t . . . 794
unpack_sms_SLQSSmsGetMessageProtocol_t . . . 795
unpack_sms_SLQSSmsSetRoutes_t . . . 795
unpack_sms_SLQSSwiGetSMSStorage_t . . . 796
unpack_sms_SLQSTransLayerInfoCallback_ind_t . . . 796
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t . . . 797
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t . . . 799
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t . . . 800
unpack_swiaudio_SLQSGetM2MAudioProfile_t . . . 800
unpack_swiaudio_SLQSGetM2MAudioVolume_t . . . 802
unpack_swiaudio_SLQSGetM2MAVMute_t . . . 802
unpack_swiaudio_SLQSGetM2MSpkrGain_t . . . 803
unpack_swiavms_SLQSAVMSEventReportInd_t . . . 804
unpack_swiavms_SLQSAVMSGetSettings_t . . . 805
unpack_swiavms_SLQSAVMSGetSettings_v2_t . . . 808
unpack_swiavms_SLQSAVMSSendSelection_t . . . 810
unpack_swiavms_SLQSAVMSSessionGetInfo_t . . . 811
unpack_swiavms_SLQSAvmsSetEventReport_t . . . 812
unpack_swiavms_SLQSAVMSSetSettings_t . . . 812

unpack_swiavms_SLQSAVMSSetSettings_v2_t . . . 813
unpack_swiavms_SLQSAVMSStartSession_t . . . 813
unpack_swiavms_SLQSAVMSStopSession_t . . . 814
unpack_swidms_SLQSSwiDmsGetMTU_t . . . 814
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t . . . 815
unpack_swidms_SLQSSwiDmsSetMTU_t . . . 816
unpack_swiloc_SwiLocGetAutoStart_t . . . 816
unpack_swioma_SLQSOMADMAlertCallback_ind_t . . . 818
unpack_swioma_SLQSOMADMGetSessionInfo_t . . . 819
unpack_swioma_SLQSOMADMGetSettings_t . . . 821
unpack_swioma_SLQSOMADMStartSession_t . . . 823
unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t . . . 823
unpack_tmd_SLQSTmdGetMitigationDevList_t . . . 824
unpack_tmd_SLQSTmdGetMitigationLvl_t . . . 824
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t . . . 825
unpack_tmd_SLQSTmdRegNotMitigationLvl_t . . . 826
unpack_uim_ChangePin_t . . . 826
unpack_uim_GetCardStatus_t . . . 827
unpack_uim_ReadTransparent_t . . . 828
unpack_uim_SetPinProtection_t . . . 829
unpack_uim_SetUimSlotStatusChangeCallback_ind_t . . . 829
unpack_uim_SLQSUIMAuthenticate_t . . . 830
unpack_uim_SLQSUIMDepersonalization_t . . . 831
unpack_uim_SLQSUIMEventRegister_t . . . 831
unpack_uim_SLQSUIMGetConfiguration_t . . . 832
unpack_uim_SLQSUIMGetFileAttributes_t . . . 832
unpack_uim_SLQSUIMGetSlotsStatus_t . . . 833
unpack_uim_SLQSUIMRefreshCallback_Ind_t . . . 834
unpack_uim_SLQSUIMRefreshGetLastEvent_t . . . 834
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t . . . 835
unpack_uim_UnblockPin_t . . . 835
unpack_uim_VerifyPin_t . . . 836
unpack_voice_allCallStatusCallback_ind_t . . . 837
unpack_voice_DTMFEventCallback_ind_t . . . 839
unpack_voice_OTASPStatusCallback_ind_t . . . 840
unpack_voice_SLQSOriginateUSSD_t . . . 841
unpack_voice_SLQSVoiceAnswerCall_t . . . 844
unpack_voice_SLQSVoiceBurstDTMF_t . . . 845
unpack_voice_SLQSVoiceDialCall_t . . . 845
unpack_voice_SLQSVoiceEndCall_t . . . 846
unpack_voice_SLQSVoiceGetAllCallInfo_t . . . 847
unpack_voice_SLQSVoiceGetCallBarring_t . . . 849
unpack_voice_SLQSVoiceGetCallForwardingStatus_t . . . 851
unpack_voice_SLQSVoiceGetCallInfo_t . . . 852
unpack_voice_SLQSVoiceGetCallWaiting_t . . . 856
unpack_voice_SLQSVoiceGetCLIP_t . . . 857
unpack_voice_SLQSVoiceGetCLIR_t . . . 858
unpack_voice_SLQSVoiceGetCNAP_t . . . 860
unpack_voice_SLQSVoiceGetCOLP_t . . . 861
unpack_voice_SLQSVoiceGetCOLR_t . . . 862
unpack_voice_SLQSVoiceGetConfig_t . . . 864
unpack_voice_SLQSVoiceManageCalls_t . . . 866
unpack_voice_SLQSVoiceSendFlash_t . . . 866
unpack_voice_SLQSVoiceSetCallBarringPassword_t . . . 867
unpack_voice_SLQSVoiceSetConfig_t . . . 868
unpack_voice_SLQSVoiceSetSUPSService_t . . . 870
unpack_voice_SLQSVoiceStartContDTMF_t . . . 871
unpack_voice_SLQSVoiceStopContDTMF_t . . . 871

unpack_voice_SLQSVoiceSUPSCallback_ind_t . . . 872
unpack_voice_SUPSNotificationCallback_ind_t . . . 874
unpack_voice_USSDNotificationCallback_ind_t . . . 876
unpack_voice_VoiceInfoRecCallback_ind_t . . . 876
unpack_voice_voicePrivacyChangeCallback_ind_t . . . 879
unpack_wds_DHCPv4ClientLease_ind_t . . . 879
unpack_wds_GetAutoconnect_t . . . 880
unpack_wds_GetByteTotals_t . . . 880
unpack_wds_GetConnectionRate_t . . . 881
unpack_wds_GetDataBearerTechnology_t . . . 882
unpack_wds_GetDefaultProfile_t . . . 882
unpack_wds_GetDefaultProfileNum_t . . . 885
unpack_wds_GetDormancyState_t . . . 885
unpack_wds_GetLastMobileIPError_t . . . 886
unpack_wds_GetMobileIP_t . . . 886
unpack_wds_GetMobileIPProfile_t . . . 887
unpack_wds_GetPacketStatistics_t . . . 888
unpack_wds_GetPacketStatus_t . . . 890
unpack_wds_GetSessionDuration_t . . . 892
unpack_wds_GetSessionState_t . . . 893
unpack_wds_RMSetTransferStatistics_t . . . 893
unpack_wds_SetMobileIPProfile_t . . . 893
unpack_wds_SLQSCreateProfile_t . . . 894
unpack_wds_SLQSDeleteProfile_t . . . 894
unpack_wds_SLQSDUNCallInfoCallBack_ind_t . . . 895
unpack_wds_SLQSGet3GPPConfigItem_t . . . 896
unpack_wds_SLQSGetCurrDataSystemStat_t . . . 898
unpack_wds_SLQSGetCurrentChannelRate_t . . . 898
unpack_wds_SLQSGetDataBearerTechnology_t . . . 899
unpack_wds_SLQSGetDUNCallInfo_t . . . 900
unpack_wds_SLQSGetProfileSettings_t . . . 903
unpack_wds_SLQSGetRuntimeSettings_t . . . 903
unpack_wds_SLQSModifyProfile_t . . . 906
unpack_wds_SLQSSetIPFamilyPreference_t . . . 907
unpack_wds_SLQSSetPacketSrvStatusCallback_t . . . 907
unpack_wds_SLQSSetWdsEventCallback_ind_t . . . 909
unpack_wds_SLQSSGetDHCPv4ClientConfig_t . . . 914
unpack_wds_SLQSSGetLoopback_t . . . 915
unpack_wds_SLQSStartDataSession_t . . . 915
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t . . . 916
UnPackGetProfileSettingOut . . . 918
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo . . . 919
UnpackSwiAvmsEventReportConfig . . . 921
UnpackSwiAvmsEventReportConnectionRequest . . . 922
UnpackSwiAvmsEventReportDataSessionStatus . . . 922
UnpackSwiAvmsEventReportHTTPStatus . . . 923
UnpackSwiAvmsEventReportNotification . . . 925
UnpackSwiAvmsEventReportPackageID . . . 926
UnpackSwiAvmsEventReportRegStatus . . . 926
UnpackSwiAvmsEventReportSessionType . . . 927
UnpackSwiAvmsEventReportWAMSParamChange . . . 927
unpackWdsProfileParam . . . 928
voice_airTimer . . . 928
voice_allCallsAlphaIDInfo . . . 929
voice_allCallsDiagInfo . . . 929
voice_allCallsUUSInfo . . . 930
voice_alphaIDInfo . . . 930
voice_arrAlertingPattern . . . 931

voice_arrAlertingType . . . 932
voice_arrAlphaID . . . 933
voice_arrCalledPartyNum . . . 933
voice_arrCallEndReason . . . 934
voice_arrCallInfo . . . 935
voice_arrConnectPartyNum . . . 935
voice_arrDiagInfo . . . 936
voice_arrRedirPartyNum . . . 936
voice_arrRemotePartyName . . . 937
voice_arrRemotePartyNum . . . 937
voice_arrSvcOption . . . 938
voice_arrUUSInfo . . . 938
voice_burstDTMFInfo . . . 939
voice_calledPartyInfo . . . 940
voice_calledPartySubAdd . . . 941
voice_callerIDInfo . . . 942
voice_callFwdTypeAndPlan . . . 943
voice_callFWExtInfo . . . 944
voice_callFWInfo . . . 946
voice_callInfo . . . 947
voice_callingPartyInfo . . . 948
voice_ccSUPSType . . . 950
voice_CLIPResp . . . 951
voice_CLIRResp . . . 951
voice_CNAPResp . . . 952
voice_COLPResp . . . 953
voice_COLRResp . . . 953
voice_connectNumInfo . . . 954
voice_CUGInfo . . . 956
voice_curAMRConfig . . . 956
voice_diagInfo . . . 957
voice_DTMFInfo . . . 958
voice_DTMFLengths . . . 958
voice_ECTNum . . . 959
voice_extDispRecInfo . . . 960
voice_getAllCallInformation . . . 961
voice_getAllCallRmtPtyName . . . 961
voice_getAllCallRmtPtyNum . . . 962
voice_getCallFWExtInfo . . . 962
voice_getCallFWInfo . . . 963
voice_lineCtrlInfo . . . 963
voice_newPwdData . . . 964
voice_NSSAudioCtrl . . . 965
voice_peerNumberInfo . . . 965
voice_prefVoiceSO . . . 967
voice_redirNumInfo . . . 969
voice_remotePartyName . . . 971
voice_remotePartyNum . . . 971
voice_roamTimer . . . 972
voice_signalInfo . . . 973
voice_SUPSInfo . . . 974
voice_USSDNotificationNetworkInfo . . . 974
voice_USSInfo . . . 975
voice_UUSInfo . . . 975
wds_channelRate . . . 977
wds_ChannelRateTlv . . . 977
wds_ConnStatusTlv . . . 978
wds_currNetworkInfo . . . 978

wds_DataBearTechTlv . . . 979
wds_DataULongLongTlv . . . 980
wds_DataULongTlv . . . 980
wds_DHCPLeaseOptTlv . . . 980
wds_DHCPLeaseStateTlv . . . 981
wds_DHCPOpt . . . 982
wds_DHCPProfileIdTlv . . . 982
wds_DHCPv4HWConfig . . . 983
wds_DHCPv4Option . . . 983
wds_DHCPv4OptionList . . . 984
wds_DHCPv4ProfileId . . . 985
wds_Domain . . . 985
wds_DomainNameList . . . 986
wds_DormStatTlv . . . 986
wds_GPRSQoS . . . 987
wds_IPv4AdTlv . . . 987
wds_IPV6AddressInfo . . . 988
wds_IPV6GWAddressInfo . . . 988
wds_LastMdmCallEndRsnTlv . . . 989
wds_PCSCFFQDNAddress . . . 990
wds_PCSCFFQDNAddressList . . . 990
wds_PCSCFIPv4ServerAddressList . . . 991
wds_ProfileIdentifier . . . 991
wds_profileInfo . . . 992
wds_RXBytesOKTlv . . . 992
wds_transferStatInd . . . 993
wds_TrStatInd . . . 993
wds_TXBytesOKTlv . . . 994
wds_UMTSMinQoS . . . 994
wdsDhcpv4HwConfig . . . 997
wdsDhcpv4Option . . . 998
wdsDhcpv4OptionList . . . 998
wdsDhcpv4ProfileId . . . 999

# Chapter 5

# File Index

## 5.1 File List

Here is a list of all files with brief descriptions:


apdoxypages.c
Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages . . . 1001
audio.h . . . 1001
cat.h . . . 1007
common.h . . . 1010
dms.h . . . 1016
fms.h . . . 1074
ims.h . . . 1077
imsa.h . . . 1089
lite-fw.h . . . 1094
loc.h . . . 1105
nas.h . . . 1127
pds.h . . . 1182
qaGobiApiTableBandClasses.h
Network Access Service API Band Classes table . . . 1204
qaGobiApiTableCallControlReturnReasons.h
Call Control Return Reasons table . . . 1207
qaGobiApiTableCallEndReasons.h
Wireless Data Service Call End Reasons . . . 1208
qaGobiApiTableCarrierCodes.h
Carrier Codes table . . . 1225
qaGobiApiTableCodingScheme.h
Data Coding Scheme . . . 1227
qaGobiApiTableGpsCapabilityCodes.h
Position Determination Service API GPS Capability Codes . . . 1229
qaGobiApiTablePowerModes.h
Device Management Service API Power Modes table . . . 1230
qaGobiApiTableRadioInterfaces.h
Network Access Service API Radio Interfaces table . . . 1231
qaGobiApiTableRegionCodes.h
Region Codes table . . . 1231
qaGobiApiTableServiceOptions.h
Voice Service Options . . . 1232
qaGobiApiTableSupServiceInfoClasses.h
Voice Supplementary Service Information Classes . . . 1234
qaGobiApiTableSwiAudio.h
Swi Audio related tables . . . 1234

qaGobiApiTableSwiOMADMSessionStatus.h
Session Status table . . . 1235
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h
Update Complete Status table . . . 1236
qaGobiApiTableVoiceCallEndReasons.h
Voice Service Call and supplementary services end reasons . . . 1237
qmerrno.h . . . 1244
qos.h . . . 1251
rms.h . . . 1261
sar.h . . . 1263
sms.h . . . 1264
swiaudio.h . . . 1287
swiavms.h . . . 1296
SwiDataTypes.h
SWI data types . . . 1310
swidms.h . . . 1311
swiloc.h . . . 1315
swioma.h . . . 1316
swiomaext.h . . . 1325
switype_256bit.h . . . 1331
SWIWWANCMAPI.h . . . 1332
tmd.h . . . 1332
uim.h . . . 1336
voice.h . . . 1355
wds.h . . . 1386

# Chapter 6

# Module Documentation

## 6.1 QMI pack/unpack (lite-qmi)

### Files

- file common.h
- file dms.h
- file loc.h
- file nas.h
- file qos.h
- file sms.h
- file swioma.h
- file uim.h
- file wds.h
- file fms.h
- file swiloc.h
- file pds.h
- file rms.h
- file sar.h
- file swiavms.h
- file voice.h
- file cat.h
- file imsa.h
- file tmd.h
- file swiomaext.h
- file audio.h
- file swiaudio.h
- file ims.h
- file swidms.h

### 6.1.1 Detailed Description

## 6.2 Streaming Download Protocol (lite-fw)

### Files

- file lite-fw.h

### 6.2.1 Detailed Description

# Chapter 7

# Namespace Documentation

## 7.1 Tables Namespace Reference

### 7.1.1 Detailed Description

Tables referenced in the API function headers:

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

# Chapter 8

# Data Structure Documentation

## 8.1 _litefw_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**

| | |
|---|---|
| *szModelid_str* | • Model Name String |
| *szFwversion_str* | • Firmware Version String. |
| *szSku_str* | • SKU String. |
| *szPackageid_str* | • Package ID String. |
| *szCarrier_str* | • Carrier String. |
| *szCarrier-Priversion_str* | • Carrier PRI Version String. |

### 8.1.2 Field Documentation

#### 8.1.2.1 char _litefw_FirmwareInfo_::szCarrier_str[20]

#### 8.1.2.2 char _litefw_FirmwareInfo_::szCarrierPriversion_str[20]

#### 8.1.2.3 char _litefw_FirmwareInfo_::szFwversion_str[20]

#### 8.1.2.4 char _litefw_FirmwareInfo_::szModelid_str[20]

#### 8.1.2.5 char _litefw_FirmwareInfo_::szPackageid_str[20]

#### 8.1.2.6 char _litefw_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**

| | |
|---|---|
| *source* | • Specifies the source of the altitude<br>• Valid values<br>– 0 - Source is unknown<br>– 1 - GPS is the source<br>– 2 - Cell ID provided the source<br>– 3 - Source is enhanced cell ID<br>– 4 - Wi-Fi is the source<br>– 5 - Terrestrial source<br>– 6 - Hybrid terrestrial source<br>– 7 - Altitude database is the source<br>– 8 - Barometric altimeter is the source<br>– 9 - Other sources |
| *linkage* | • Specifies the dependency between the horizontal and altitude position components<br>• Valid values<br>– 0 - Not specified<br>– 1 - Fully interdependent<br>– 2 - Depends on latitude and longitude<br>– 3 - Fully independent |

| | |
|---|---|
| *coverage* | • Specifies the region of uncertainty.<br>• Valid values<br>– 0 - Not specified<br>– 1 - Altitude uncertainty is valid at the injected horizontal position coordinates only<br>– 2 - Altitude uncertainty applies to the position of the device regardless of horizontal position |

### 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**

| | |
|---|---|
| *appType* | • Indicates the type of the application.<br>– 0 - Unknown<br>– 1 - SIM card<br>– 2 - USIM application<br>– 3 - RUIM card<br>– 4 - CSIM application<br>– 5 - ISIM application<br>• Other values are reserved for the future and are to be handled as "Unknown". |

| | |
|---|---|
| *appState* | • Indicates the state of the application.<br>– 0 - Unknown<br>– 1 - Detected<br>– 2 - PIN1 or UPIN is required<br>– 3 - PUK1 or PUK for UPIN is required<br>– 4 - Personalization state must be checked<br>– 5 - PIN1 is blocked<br>– 6 - Illegal<br>– 7 - Ready |
| *persoState* | • Indicates the state of the personalization for the application.<br>– 0 - Unknown<br>– 1 - Personalization operation is in progress<br>– 2 - Ready<br>– 3 - Personalization code is required<br>– 4 - PUK for personalization code is required<br>– 5 - Permanently blocked |
| *persoFeature* | • Indicates the personalization feature.<br>• This applies only when a personalization code is required to deactivate or unblock personalization.<br>– 0 - GW network personalization<br>– 1 - GW network subset personalization<br>– 2 - GW service provider personalization<br>– 3 - GW corporate personalization<br>– 4 - GW UIM personalization<br>– 5 - 1X network type 1 personalization<br>– 6 - 1X network type 2 personalization<br>– 7 - 1X HRPD personalization<br>– 8 - 1X service provider personalization<br>– 9 - 1X corporate personalization<br>– 10 - 1X RUIM personalization<br>– 11 - Unknown |
| *persoRetries* | • Indicates the number of retries remaining to disable the personalization. |
| *persoUnblock-Retries* | • Indicates the number of retries remaining to unblock the personalization. |
| *aidLength* | • Number of sets of the following elements. i.e. aidVal<br>• If zero(0) then no aidVal information exists. |
| *aidVal[MAX_DE-SCRIPTION_LE-NGTH]* | • Application identifier value. |

| | |
|---|---|
| *univPin* | • Indicates whether UPIN replaces PIN1.<br>– 0 - PIN1 is used<br>– 1 - UPIN replaces PIN1 |
| *pin1State* | • Indicates the state of PIN1.<br>– 0 - Unknown<br>– 1 - Enabled and not verified<br>– 2 - Enabled and verified<br>– 3 - Disabled<br>– 4 - Blocked<br>– 5 - Permanently blocked |
| *pin1Retries* | • Indicates the number of retries remaining to verify PIN1. |
| *puk1Retries* | • Indicates the number of retries remaining to unblock PIN1. |
| *pin2State* | • Indicates the state of PIN2.<br>– 0 - Unknown<br>– 1 - Enabled and not verified<br>– 2 - Enabled and verified<br>– 3 - Disabled<br>– 4 - Blocked<br>– 5 - Permanently blocked |
| *pin2Retries* | • Indicates the number of retries remaining to verify PIN2. |
| *puk2Retries* | • Indicates the number of retries remaining to unblock PIN2. |

### 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 audio_RXAGCList Struct Reference

### Data Fields

- uint16_t * pRXStaticGain
- uint16_t * pRXAIG
- uint16_t * pRXExpThres
- uint16_t * pRXExpSlope
- uint16_t * pRXComprThres
- uint16_t * pRXComprSlope

### 8.4.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXAGCLIST.

**Parameters**

| | |
|---|---|
| *pRXStaticGain* | • RX pre-compressor static gain |
| *pRXAIG* | • RX pre-compressor gain selection flag |
| *pRXExpThres* | • RX expansion threshold |
| *pRXExpSlope* | • RX expansion slope |
| *pRXComprThres* | • RX compression threshold |
| *pRXComprSlope* | • RX compression slope |

### 8.4.2 Field Documentation

#### 8.4.2.1 uint16_t* audio_RXAGCList::pRXAIG

#### 8.4.2.2 uint16_t* audio_RXAGCList::pRXComprSlope

#### 8.4.2.3 uint16_t* audio_RXAGCList::pRXComprThres

#### 8.4.2.4 uint16_t* audio_RXAGCList::pRXExpSlope

#### 8.4.2.5 uint16_t* audio_RXAGCList::pRXExpThres

#### 8.4.2.6 uint16_t* audio_RXAGCList::pRXStaticGain

## 8.5 audio_RXAVCList Struct Reference

### Data Fields

- uint16_t * pAVRXAVCSens
- uint16_t * pAVRXAVCHeadroom

### 8.5.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXAVCLIST.

**Parameters**

| | |
|---|---|
| *pAVRXAVC-Sens* | • AVC variation from nominal sensitivity |
| *pAVRXAVC-Headroom* | • AVC headroom |

### 8.5.2 Field Documentation

#### 8.5.2.1 uint16_t* audio_RXAVCList::pAVRXAVCHeadroom

#### 8.5.2.2 uint16_t* audio_RXAVCList::pAVRXAVCSens

## 8.6 audio_RXPCMIIRFltr Struct Reference

### Data Fields

- uint16_t * pFlag
- uint16_t * pStageCnt
- uint8_t * pStage0Val
- uint8_t * pStage1Val
- uint8_t * pStage2Val
- uint8_t * pStage3Val
- uint8_t * pStage4Val

### 8.6.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_RXPCMIIRFLTR.

**Parameters**

| | |
|---|---|
| *pFlag* | • Flag<br>– 0x0000 - IIR filter disable<br>– 0xffff - IIR filter enable |
| *pStageCnt* | • Stage Count<br>– 0-4 |
| *pStage0Val* | • A 20 BYTE sized parameter indicating Stage 0 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage1Val* | • A 20 BYTE sized parameter indicating Stage 1 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage2Val* | • A 20 BYTE sized parameter indicating Stage 2 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage3Val* | • A 20 BYTE sized parameter indicating Stage 3 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage4Val* | • A 20 BYTE sized parameter indicating Stage 4 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |

### 8.6.2 Field Documentation

#### 8.6.2.1 uint16_t* audio_RXPCMIIRFltr::pFlag

#### 8.6.2.2 uint8_t* audio_RXPCMIIRFltr::pStage0Val

#### 8.6.2.3 uint8_t* audio_RXPCMIIRFltr::pStage1Val

#### 8.6.2.4 uint8_t* audio_RXPCMIIRFltr::pStage2Val

#### 8.6.2.5 uint8_t* audio_RXPCMIIRFltr::pStage3Val

#### 8.6.2.6 uint8_t* audio_RXPCMIIRFltr::pStage4Val

#### 8.6.2.7 uint16_t* audio_RXPCMIIRFltr::pStageCnt

## 8.7 audio_TXAGCList Struct Reference

### Data Fields

- uint16_t * pTXStaticGain
- uint16_t * pTXAIG
- uint16_t * pTXExpThres
- uint16_t * pTXExpSlope
- uint16_t * pTXComprThres
- uint16_t * pTXComprSlope

### 8.7.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_TXAGCLIST.

**Parameters**

| | |
|---|---|
| *pTXStaticGain* | • TX pre-compressor static gain |
| *pTXAIG* | • TX pre-compressor gain selection flag |
| *pTXExpThres* | • TX expansion threshold |
| *pTXExpSlope* | • TX expansion slope |
| *pTXComprThres* | • TX compression threshold |
| *pTXComprSlope* | • TX compression slope |

### 8.7.2 Field Documentation

#### 8.7.2.1 uint16_t* audio_TXAGCList::pTXAIG

#### 8.7.2.2 uint16_t* audio_TXAGCList::pTXComprSlope

#### 8.7.2.3 uint16_t* audio_TXAGCList::pTXComprThres

#### 8.7.2.4 uint16_t* audio_TXAGCList::pTXExpSlope

#### 8.7.2.5 uint16_t* audio_TXAGCList::pTXExpThres

#### 8.7.2.6 uint16_t* audio_TXAGCList::pTXStaticGain

## 8.8 audio_TXPCMIIRFltr Struct Reference

### Data Fields

- uint16_t * pFlag
- uint16_t * pStageCnt
- uint8_t * pStage0Val
- uint8_t * pStage1Val
- uint8_t * pStage2Val
- uint8_t * pStage3Val
- uint8_t * pStage4Val

### 8.8.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV_TXPCMIIRFLTR.

**Parameters**

| | |
|---|---|
| *pFlag* | • Flag<br>– 0x0000 - IIR filter disable<br>– 0xffff - IIR filter enable |
| *pStageCnt* | • Stage Count<br>– 0-4 |
| *pStage0Val* | • A 20 BYTE sized parameter indicating Stage 0 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |

| | |
|---|---|
| *pStage1Val* | • A 20 BYTE sized parameter indicating Stage 1 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage2Val* | • A 20 BYTE sized parameter indicating Stage 2 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage3Val* | • A 20 BYTE sized parameter indicating Stage 3 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |
| *pStage4Val* | • A 20 BYTE sized parameter indicating Stage 4 value<br>– A1<br>– A2<br>– B0<br>– B1<br>– B2 |

### 8.8.2 Field Documentation

#### 8.8.2.1 uint16_t* audio_TXPCMIIRFltr::pFlag

#### 8.8.2.2 uint8_t* audio_TXPCMIIRFltr::pStage0Val

#### 8.8.2.3 uint8_t* audio_TXPCMIIRFltr::pStage1Val

#### 8.8.2.4 uint8_t* audio_TXPCMIIRFltr::pStage2Val

#### 8.8.2.5 uint8_t* audio_TXPCMIIRFltr::pStage3Val

#### 8.8.2.6 uint8_t* audio_TXPCMIIRFltr::pStage4Val

#### 8.8.2.7 uint16_t* audio_TXPCMIIRFltr::pStageCnt

## 8.9 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 [100]
- uint8_t m_PriImageId [16]
- uint8_t m_PriBuildId [100]

## 8.9.1 Detailed Description

This structure contains the Carrier Image parameters.

**Parameters**

| | |
|---|---|
| *m_nCarrierId* | • Unique numeric carrier ID indicating the carrier that the following images belong to |
| *m_nFolderId* | • Unique numeric folder ID indicating the folder where the images should reside on the host storage. |
| *m_nStorage* | • Information of storage type<br>• Values<br>– 0 - Device<br>– 1 - Host |
| *m_FwImageId* | • Firmware image ID |
| *m_FwBuildId* | • Firmware build ID |
| *m_PriImageId* | • PRI image ID |
| *m_PriBuildId* | • PRI build ID |

## 8.9.2 Field Documentation

### 8.9.2.1 uint8_t CarrierImage_t::m_FwBuildId[100]

### 8.9.2.2 uint8_t CarrierImage_t::m_FwImageId[16]

### 8.9.2.3 uint32_t CarrierImage_t::m_nCarrierId

### 8.9.2.4 uint32_t CarrierImage_t::m_nFolderId

### 8.9.2.5 uint32_t CarrierImage_t::m_nStorage

#### 8.9.2.6 uint8_t CarrierImage_t::m_PriBuildId[100]

#### 8.9.2.7 uint8_t CarrierImage_t::m_PriImageId[16]

## 8.10 cat_AlPhaIdentifierTlv Struct Reference

### Data Fields

- uint8_t ReferenceID
- uint16_t AlphaIDLength
- uint8_t AlphaID [255]

### 8.10.1 Detailed Description

structure used to store all Alpha Identifier parameters.

**Parameters**

| | |
|---|---|
| *ReferenceID* | - proactive command type that included the alpha identifier - 0x01; sends SMS proactive command |
| *AlphaIDLength* | - length of AlphaID ( in bytes ) |
| *AlphaID* | - alpha identifier, encoded as in ETSI TS 102 223 [Section 8.2] |

### 8.10.2 Field Documentation

#### 8.10.2.1 uint8_t cat_AlPhaIdentifierTlv::AlphaID[255]

#### 8.10.2.2 uint16_t cat_AlPhaIdentifierTlv::AlphaIDLength

#### 8.10.2.3 uint8_t cat_AlPhaIdentifierTlv::ReferenceID

## 8.11 cat_commonEventTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t EventID
- uint16_t EventLength
- union cat_currentCatEvent CatEvent

### 8.11.1 Detailed Description

Structure used to store all Common CAT Event TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *EventID* | - Event ID. Can be any of the following<br>16 - Display Text 17 - Get In-Key 18 - Get Input 19 - Setup Menu 20 - Select Item 21 - Send SMS - Alpha Identifier 22 - Setup Event List 23 - Setup Idle Mode Text 24 - Language Notification 25 - Refresh 26 - End Proactive Session |
| *EventLength* | - Length of pData ( in Bytes ) |
| *CatEvent* | - Structure to the Data specific to the CAT event ID |

### 8.11.2 Field Documentation

#### 8.11.2.1 union cat_currentCatEvent cat_commonEventTlv::CatEvent

#### 8.11.2.2 uint8_t cat_commonEventTlv::EventID

#### 8.11.2.3 uint16_t cat_commonEventTlv::EventLength

#### 8.11.2.4 uint8_t cat_commonEventTlv::TlvPresent

## 8.12 cat_currentCatEvent Union Reference

### Data Fields

- struct cat_EventIDDataTlv CatEvIDData
- struct cat_AlPhaIdentifierTlv CatAlphaIdtfr
- struct cat_EventListTlv CatEventLst
- struct cat_RefreshTlv CatRefresh
- struct cat_EndProactiveSessionTlv CatEndPS

### 8.12.1 Detailed Description

Union used to represent the current CAT Event Data. Choose the structure based on the EventID received.

- Use cat_EventIDDataTlv if the Event ID is any of the below.
  - 16
  - 17
  - 18
  - 19
  - 20
  - 23
  - 24
- Use cat_AlPhaIdentifierTlv if the Event ID is 21
- Use cat_EventListTlv if the Event ID is 22
- Use cat_RefreshTlv if the Event ID is 25
- Use cat_EndProactiveSessionTlv if the Event ID is 26

### 8.12.2 Field Documentation

#### 8.12.2.1 struct cat_AlPhaIdentifierTlv cat_currentCatEvent::CatAlphaIdtfr

#### 8.12.2.2 struct cat_EndProactiveSessionTlv cat_currentCatEvent::CatEndPS

#### 8.12.2.3 struct cat_EventListTlv cat_currentCatEvent::CatEventLst

#### 8.12.2.4 struct cat_EventIDDataTlv cat_currentCatEvent::CatEvIDData

#### 8.12.2.5 struct cat_RefreshTlv cat_currentCatEvent::CatRefresh

## 8.13 cat_EndProactiveSessionTlv Struct Reference

### Data Fields

- uint8_t EndProactiveSession

### 8.13.1 Detailed Description

structure used to store End Proactive Session event parameters.

**Parameters**

| | |
|---|---|
| *EndProactive-Session* | - The proactive session end type values are:<br>• 0x01 - End proactive session command type received from the card<br>• 0x02 - End proactive session internal to ME |

### 8.13.2 Field Documentation

#### 8.13.2.1 uint8_t cat_EndProactiveSessionTlv::EndProactiveSession

## 8.14 cat_EventIDDataTlv Struct Reference

### Data Fields

- uint32_t ReferenceID
- uint16_t DataLength
- uint8_t Data [255]

### 8.14.1 Detailed Description

structure used to store all Common CAT Event parameters.

**Parameters**

| | |
|---|---|
| *ReferenceID* | - proactive command reference ID. |
| *DataLength* | - length of pData ( in Bytes ) |
| *Data* | - command specific to the CAT event ID, encoded as in ETSI TS 102 223 [Section 6.6.X] |

### 8.14.2 Field Documentation

#### 8.14.2.1 uint8_t cat_EventIDDataTlv::Data[255]

#### 8.14.2.2 uint16_t cat_EventIDDataTlv::DataLength

#### 8.14.2.3 uint32_t cat_EventIDDataTlv::ReferenceID

## 8.15 cat_EventListTlv Struct Reference

### Data Fields

- uint32_t SetupEventList

### 8.15.1 Detailed Description

structure used to store all Event List parameters.

**Parameters**

| | |
|---|---|
| *SetupEventList* | - Setup event list bit mask<br>• 0x00000001 - User Activity Notify<br>• 0x00000002 - Idle Screen Available<br>• 0x00000004 - Lang Selection Notify Each set bit indicates the availability of the corresponding event in Setup Event list proactive command; all unlisted bits are reserved for future use and will be ignored |

### 8.15.2 Field Documentation

#### 8.15.2.1 uint32_t cat_EventListTlv::SetupEventList

## 8.16 cat_RefreshTlv Struct Reference

### Data Fields

- uint16_t RefreshMode
- uint8_t RefreshStage

### 8.16.1 Detailed Description

structure used to store all Refresh Event parameters.

**Parameters**

| | |
|---|---|
| *RefreshMode* | - The Refresh Event as in ETSI TS 102 223 [Section 8.6] |
| *RefreshStage* | - Stage of a refresh procedure<br>• 0x01 - Refresh start<br>• 0x02 - Refresh success<br>• 0x03 - Refresh failed |

### 8.16.2 Field Documentation

#### 8.16.2.1 uint16_t cat_RefreshTlv::RefreshMode

#### 8.16.2.2 uint8_t cat_RefreshTlv::RefreshStage

## 8.17 cdmaSSInfo Struct Reference

### Data Fields

- int8_t rssi
- int16_t ecio

### 8.17.1 Detailed Description

This structure contains the parameters for CDMA/WCDMA Signal Strength Information

**Parameters**

| | |
|---|---|
| *rssi* | • RSSI in dBm (signed value).<br>• A value of -125 dBm or lower is used to indicate No Signal. |
| *ecio* | • ECIO value representing negative 0.5 dBm increments, i.e., 2 means -1 dBm (14 means -7 dBm, 63 means -31.5 dBm). |

### 8.17.2 Field Documentation

#### 8.17.2.1 int16_t cdmaSSInfo::ecio

#### 8.17.2.2 int8_t cdmaSSInfo::rssi

## 8.18 connectionStatus Struct Reference

### Data Fields

- uint8_t MDMConnStatus
- uint64_t MDMCallDuration

### 8.18.1 Detailed Description

This structure contains modem connection status

**Parameters**

| | |
|---|---|
| *MDMConn-Status* | • Current link status<br>– 0x01 - DISCONNECTED<br>– 0x02 - CONNECTED<br>– 0xff - Invalid data. |
| *MDMCall-Duration* | • Call duration in milliseconds.<br>• If the modem connection status is connected, this represent the duration of the current DUN call.<br>• If the modem connection status is disconnected, this represents the duration of the last DUN call since the device was powered up (zero, if no call has been made or if the last call was not DUN).<br>– 0xffffffffffffffff - Invalid data. |

### 8.18.2 Field Documentation

#### 8.18.2.1 uint64_t connectionStatus::MDMCallDuration

#### 8.18.2.2 uint8_t connectionStatus::MDMConnStatus

## 8.19 crashInfoParams Struct Reference

### Data Fields

- uint8_t crashStatus
- crashInformation crashInfo

### 8.19.1 Detailed Description

This structure contains crash information parameters

**Parameters**

| | |
|---|---|
| *crashStatus[OUT]* | • Device Crash Status<br>• 0 - no crash<br>• 1 - crash has occurred |
| *crashInfo[OUT]* | • See crashInformation |

### 8.19.2 Field Documentation

#### 8.19.2.1 crashInformation crashInfoParams::crashInfo

#### 8.19.2.2 uint8_t crashInfoParams::crashStatus

## 8.20 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]

### 8.20.1 Detailed Description

This structure contains crash information

**Parameters**

| | |
|---|---|
| *numCrashes[OUT]* | • number of instances of the remaining fields |
| *crashId[OUT]* | • ramdom crash id assigned at crash |

| | |
|---|---|
| *crashData* | • crash data[OUT] |
| *crashStrlen[IN/-OUT]* | • length of the crashString field as an input, length of the crashString field returned by API as an OUTPUT |
| *crashString* | • crash string |
| *gcdumpStrlen[O-UT]* | • length of the gcdumpString field as an input, length of the gcdumpString field returned by API as an OUTPUT |
| *gcdumpString[O-UT]* | • gcdump string for the crash |

### 8.20.2 Field Documentation

#### 8.20.2.1 uint32_t crashInformation::crashData

#### 8.20.2.2 uint32_t crashInformation::crashId

#### 8.20.2.3 char crashInformation::crashString[255]

#### 8.20.2.4 uint16_t crashInformation::crashStrlen

#### 8.20.2.5 char crashInformation::gcdumpString[1024]

#### 8.20.2.6 uint16_t crashInformation::gcdumpStrlen

#### 8.20.2.7 uint16_t crashInformation::numCrashes

## 8.21 currNetworkInfo Struct Reference

### Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

### 8.21.1 Detailed Description

Network information structure

**Parameters**

| | |
|---|---|
| *NetworkType* | • Values:<br>– 0 - 3GPP<br>– 1 - 3GPP2 |

| | |
|---|---|
| *RATMask* | RAT Mask |
| *SOMask* | SO Mask |

### 8.21.2 Field Documentation

#### 8.21.2.1 uint8_t currNetworkInfo::NetworkType

#### 8.21.2.2 uint32_t currNetworkInfo::RATMask

#### 8.21.2.3 uint32_t currNetworkInfo::SOMask

## 8.22 dms_ActivationStatusTlv Struct Reference

### Data Fields

- uint16_t TlvPresent
- uint32_t activationStatus

### 8.22.1 Detailed Description

Activation Status Tlv

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present |
| *activationStatus* | • Service Activation Code<br>– 0 - Service not activated<br>– 1 - Service activated<br>– 2 - Activation connecting<br>– 3 - Activation connected<br>– 4 - OTASP security authenticated<br>– 5 - OTASP NAM downloaded<br>– 6 - OTASP MDN downloaded<br>– 7 - OTASP IMSI downloaded<br>– 8 - OTASP PRL downloaded<br>– 9 - OTASP SPC downloaded<br>– 10 - OTASP settings committed |

### 8.22.2 Field Documentation

#### 8.22.2.1 uint32_t dms_ActivationStatusTlv::activationStatus

#### 8.22.2.2 uint16_t dms_ActivationStatusTlv::TlvPresent

## 8.23 dms_OperatingModeTlv Struct Reference

### Data Fields

- uint16_t TlvPresent
- uint32_t operatingMode

### 8.23.1 Detailed Description

Operating Mode Tlv

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present |
| *operatingMode* | • 0 - Online<br>• 1 - Low power<br>• 2 - Factory test mode<br>• 3 - Offline<br>• 4 - Resetting<br>• 5 - Shutting down<br>• 6 - Persistent low power<br>• 7 - Mode-only low power |

### 8.23.2 Field Documentation

#### 8.23.2.1 uint32_t dms_OperatingModeTlv::operatingMode

#### 8.23.2.2 uint16_t dms_OperatingModeTlv::TlvPresent

## 8.24 dms_PSMActiveTimerIndTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t ActiveTimerInd

### 8.24.1 Detailed Description

This Structure used to store DMS PSM Active Timer TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *ActiveTimerInd* | • Active Timer<br>• PSM active timer value (in seconds). |

### 8.24.2 Field Documentation

#### 8.24.2.1 uint32_t dms_PSMActiveTimerIndTlv::ActiveTimerInd

#### 8.24.2.2 uint8_t dms_PSMActiveTimerIndTlv::TlvPresent

## 8.25 dms_PSMActiveTimerTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t activeTimer

### 8.25.1 Detailed Description

This structure contains the parameters for Power Save Mode active timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *activeTimer* | Active Timer<br>• Power Save mode active timer value (in seconds). |

### 8.25.2 Field Documentation

#### 8.25.2.1 uint32_t dms_PSMActiveTimerTlv::activeTimer

#### 8.25.2.2 uint8_t dms_PSMActiveTimerTlv::TlvPresent

## 8.26 dms_PSMDurationDueToOOSTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t durationDueToOOS

### 8.26.1 Detailed Description

This structure contains the parameters for Power Save Mode Duration due to an outage.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *durationDueTo-OOS* | Duration Due to OOS<br>• Power Save mode duration due to an outage (in seconds). |

### 8.26.2 Field Documentation

#### 8.26.2.1 uint32_t dms_PSMDurationDueToOOSTlv::durationDueToOOS

#### 8.26.2.2 uint8_t dms_PSMDurationDueToOOSTlv::TlvPresent

## 8.27 dms_PSMDurationThresholdTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t durationThreshold

### 8.27.1 Detailed Description

This structure contains the parameters for Power Save Mode Duration Threshold.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *durationThreshold* | Duration Threshold<br>• Minimum duration for the device to benefit by entering PSM (in seconds). |

### 8.27.2 Field Documentation

#### 8.27.2.1 uint32_t dms_PSMDurationThresholdTlv::durationThreshold

#### 8.27.2.2 uint8_t dms_PSMDurationThresholdTlv::TlvPresent

## 8.28 dms_PSMEarlyWakeupTimeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t earlyWakeupTime

### 8.28.1 Detailed Description

This structure contains the parameters for Power Save Mode early wakeup time.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *earlyWakeupTime* | • Early Wakeup Time<br>• Power Save mode early wakeup time (in seconds) indicating how early the device should exit PSM to offset for bootup and acquisition delay. |

### 8.28.2 Field Documentation

#### 8.28.2.1 uint32_t dms_PSMEarlyWakeupTimeTlv::earlyWakeupTime

#### 8.28.2.2 uint8_t dms_PSMEarlyWakeupTimeTlv::TlvPresent

## 8.29 dms_PSMEnableStateIndTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t EnableStateInd

### 8.29.1 Detailed Description

This Structure used to store DMS PSM Enable state TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *EnableStateInd* | • PSM Enable State<br>• Values<br>– 0 - PSM is not enabled<br>– 1 - PSM is enabled |

### 8.29.2 Field Documentation

#### 8.29.2.1 uint8_t dms_PSMEnableStateIndTlv::EnableStateInd

#### 8.29.2.2 uint8_t dms_PSMEnableStateIndTlv::TlvPresent

## 8.30 dms_PSMEnableStateTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t enableState

### 8.30.1 Detailed Description

This structure contains the parameters for Power Save Mode Enable State.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |

| | |
|---|---|
| *enableState* | Power Save Mode Enable State Values<br>• 0 - PSM is not enabled<br>• 1 - PSM is enabled |

### 8.30.2 Field Documentation

#### 8.30.2.1 uint8_t dms_PSMEnableStateTlv::enableState

#### 8.30.2.2 uint8_t dms_PSMEnableStateTlv::TlvPresent

## 8.31 dms_PSMPeriodicUpdateTimerIndTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t PeriodicUpdateTimerInd

### 8.31.1 Detailed Description

This Structure used to store DMS PSM Periodic Update Timer TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *PeriodicUpdate-TimerInd* | • Periodic Update Timer<br>• PSM periodic update timer value (in seconds). |

### 8.31.2 Field Documentation

#### 8.31.2.1 uint32_t dms_PSMPeriodicUpdateTimerIndTlv::PeriodicUpdateTimerInd

#### 8.31.2.2 uint8_t dms_PSMPeriodicUpdateTimerIndTlv::TlvPresent

## 8.32 dms_PSMPeriodicUpdateTimerTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t periodicUpdateTimer

### 8.32.1 Detailed Description

This structure contains the parameters for Power Save Mode periodic update timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *periodicUpdate-Timer* | Periodic Update Timer<br>• Power Save mode periodic update timer value (in seconds). |

### 8.32.2 Field Documentation

#### 8.32.2.1 uint32_t dms_PSMPeriodicUpdateTimerTlv::periodicUpdateTimer

#### 8.32.2.2 uint8_t dms_PSMPeriodicUpdateTimerTlv::TlvPresent

## 8.33 dms_PSMRandomizationWindowTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t randomizationWindow

### 8.33.1 Detailed Description

This structure contains the parameters for Power Save Mode wakeup randomization window.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *randomization-Window* | Randomization Window<br>• Power Save mode wakeup randomization window (in seconds) |

### 8.33.2 Field Documentation

#### 8.33.2.1 uint32_t dms_PSMRandomizationWindowTlv::randomizationWindow

#### 8.33.2.2 uint8_t dms_PSMRandomizationWindowTlv::TlvPresent

## 8.34 dms_TemperatureTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t TempStat
- uint16_t Temperature

### 8.34.1 Detailed Description

This structure stores information about temperature TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *TempStat* | • Temperature state<br>– 0 – unknown<br>– 1 – normal<br>– 2 – high (warning)<br>– 3 – high (critical)<br>– 4 – low (critical) |
| *Temperature* | • temperature in degree celsius |

### 8.34.2 Field Documentation

#### 8.34.2.1 uint16_t dms_TemperatureTlv::Temperature

#### 8.34.2.2 uint8_t dms_TemperatureTlv::TempStat

#### 8.34.2.3 uint8_t dms_TemperatureTlv::TlvPresent

## 8.35 dms_VoltageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t VoltStat
- uint16_t Voltage

### 8.35.1 Detailed Description

This structure stores information about voltage TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *VoltStat* | • Voltage state<br>– 0 – unknown<br>– 1 – normal<br>– 2 – low (warning)<br>– 3 – low (critical)<br>– 4 – high (critical) |
| *Voltage* | • Voltage in mV |

### 8.35.2 Field Documentation

#### 8.35.2.1 uint8_t dms_VoltageTlv::TlvPresent

#### 8.35.2.2 uint16_t dms_VoltageTlv::Voltage

#### 8.35.2.3 uint8_t dms_VoltageTlv::VoltStat

## 8.36 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

### 8.36.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *id_length* | • length of cust_id field |
| *cust_id* | • Customization ID (Maximum 64 bytes)<br>• NULL terminated ASCII string. |
| *value_length* | • length of cust_value field |
| *cust_value* | • Customization Setting Value (Maximum 8 bytes) |
| *cust_attr* | • Customization Setting attribute through QMI<br>  – bit 0: Values:<br>  * 0 - read only<br>  * 1 - read/write |

### 8.36.2 Field Documentation

#### 8.36.2.1 uint16_t DMScustSettingInfo::cust_attr

#### 8.36.2.2 uint8_t DMScustSettingInfo::cust_id[64+1]

#### 8.36.2.3 uint8_t DMScustSettingInfo::cust_value[8+1]

#### 8.36.2.4 uint16_t DMScustSettingInfo::id_length

#### 8.36.2.5 uint16_t DMScustSettingInfo::value_length

## 8.37 DMScustSettingList Struct Reference

### Data Fields

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

### 8.37.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *list_type* | • list type requested |
| *num_instances* | • number of instances of customization setting |
| *custSetting* | • See DMScustSettingInfo for more information |

### 8.37.2 Field Documentation

#### 8.37.2.1 DMScustSettingInfo DMScustSettingList::custSetting[255+1]

#### 8.37.2.2 uint8_t DMScustSettingList::list_type

#### 8.37.2.3 uint16_t DMScustSettingList::num_instances

## 8.38 DMSgetCustomFeatureV2 Struct Reference

### Data Fields

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

### 8.38.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pGetCustomInput[IN]* | • See DMSgetCustomInput for more information |

| | |
|---|---|
| *pCustSetting-Info[OUT]* | • See DMScustSettingInfo for more information |
| *pCustSetting-List[OUT]* | • See DMScustSettingList for more information |

### 8.38.2 Field Documentation

#### 8.38.2.1 DMScustSettingInfo* DMSgetCustomFeatureV2::pCustSettingInfo

#### 8.38.2.2 DMScustSettingList* DMSgetCustomFeatureV2::pCustSettingList

#### 8.38.2.3 DMSgetCustomInput* DMSgetCustomFeatureV2::pGetCustomInput

## 8.39 DMSgetCustomInput Struct Reference

### Data Fields

- uint8_t cust_id [64+1]
- uint8_t list_type

### 8.39.1 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**

| | |
|---|---|
| *cust_id* | • Customization ID (Maximum 64 bytes)<br>• NULL terminated ASCII string. |
| *list_type* | • list type requested |

### 8.39.2 Field Documentation

#### 8.39.2.1 uint8_t DMSgetCustomInput::cust_id[64+1]

#### 8.39.2.2 uint8_t DMSgetCustomInput::list_type

## 8.40 dunchannelRate Struct Reference

### Data Fields

- uint32_t CurrChanTxRate
- uint32_t CurrChanRxRate
- uint32_t MaxChanTxRate
- uint32_t MaxChanRxRate

### 8.40.1 Detailed Description

This structure contains Channel Rate

**Parameters**

| | |
|---|---|
| *CurrChanTxRate* | • Instantaneous channel Tx rate in bits per second<br>• 0xffffffff - Invalid data. |
| *CurrChanRx-Rate* | • Instantaneous channel Rx rate in bits per second<br>• 0xffffffff - Invalid data. |
| *MaxChanTxRate* | • maximum Tx rate that can be assigned to the device by the serving system in bits per second<br>• 0xffffffff - Invalid data. |
| *MaxChanRx-Rate* | • maximum Rx rate that can be assigned to the device by the serving system in bits per second<br>• 0xffffffff - Invalid data. |

### 8.40.2 Field Documentation

#### 8.40.2.1 uint32_t dunchannelRate::CurrChanRxRate

#### 8.40.2.2 uint32_t dunchannelRate::CurrChanTxRate

#### 8.40.2.3 uint32_t dunchannelRate::MaxChanRxRate

#### 8.40.2.4 uint32_t dunchannelRate::MaxChanTxRate

## 8.41 eriDataparams Struct Reference

### Data Fields

- uint16_t eriDataLen
- uint8_t eriData [1024]

### 8.41.1 Detailed Description

This structure contains Extended Roaming Indicator(ERI) file parameters

**Parameters**

| | |
|---|---|
| *eriDataLen* | • Upon input, the maximum number of bytes that file contents array can contain.<br>• Upon successful output, actual number of bytes written to file contents array |
| *eriData* | • ERI data read from persistent storage( Max size is 1024 ) |

### 8.41.2 Field Documentation

#### 8.41.2.1 uint8_t eriDataparams::eriData[1024]

#### 8.41.2.2 uint16_t eriDataparams::eriDataLen

## 8.42 eTWSPLMNInfoTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSEtwsPlmnInfo ETWSPLMNInfo

### 8.42.1 Detailed Description

This structure contains ETWS PLMN TLV information

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *ETWSPLMNInfo* | • ETWS PLMN Information<br>• See sMSEtwsPlmnInfo for more information |

### 8.42.2 Field Documentation

#### 8.42.2.1 sMSEtwsPlmnInfo eTWSPLMNInfoTlv::ETWSPLMNInfo

#### 8.42.2.2 uint8_t eTWSPLMNInfoTlv::TlvPresent

## 8.43 FMSImageElement Struct Reference

### Data Fields

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

### 8.43.1 Detailed Description

This structure contains the Image Element information

**Parameters**

| | |
|---|---|
| *imageType* | • Type of image 0 - Modem 1 - PRI |
| *imageId* | • Unique image identifier |

| | |
|---|---|
| *buildIdLength* | • Length of the build ID string (may be zero) |
| *pBuildId* | • Build ID ANSI string with length provided by the previous field |

### 8.43.2 Field Documentation

#### 8.43.2.1 uint8_t FMSImageElement::buildId[100]

#### 8.43.2.2 uint8_t FMSImageElement::buildIdLength

#### 8.43.2.3 uint8_t FMSImageElement::imageId[16]

#### 8.43.2.4 uint8_t FMSImageElement::imageType

## 8.44 FMSImageIdElement Struct Reference

### Data Fields

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

### 8.44.1 Detailed Description

This structure contains the Image ID list element Information

**Parameters**

| | |
|---|---|
| *storageIndex* | • Index in storage where the image is located(a value of 0xFF indicates that the storage for this type of image is not relevant) |
| *failureCount* | • Number of consecutive write attempts to this storage index that have failed(a value of 0xFF indicates unspecified) |
| *imageID* | • Image unique identifier(max 16 chars.) |
| *buildIDLength* | • Length of the build ID string. If there is no build ID, this field will be 0 and no data will follow. |
| *buildID* | • String containing image build information( Max 100 characters ) |

### 8.44.2 Field Documentation

#### 8.44.2.1 uint8_t FMSImageIdElement::buildID[100]

#### 8.44.2.2 uint8_t FMSImageIdElement::buildIDLength

#### 8.44.2.3 uint8_t FMSImageIdElement::failureCount

#### 8.44.2.4 uint8_t FMSImageIdElement::imageID[16]

#### 8.44.2.5 uint8_t FMSImageIdElement::storageIndex

## 8.45 FMSImageIDEntries Struct Reference

### Data Fields

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

### 8.45.1 Detailed Description

This structure contains the list entry Information

**Parameters**

| | |
|---|---|
| *imageType* | • Type of image<br>– 0 - Modem<br>– 1 - PRI |
| *maxImages* | • Maximum number of images of this type that may be stored concurrently on the device |
| *executingImage* | • Index (into the next array) of image that is currently executing |
| *imageIDSize* | • The number of elements in the image ID list |
| *imageIDElement* | • Array of ImageIDElement Structure ( Max 50 elements )<br>• See FMSImageIdElement |

### 8.45.2 Field Documentation

#### 8.45.2.1 uint8_t FMSImageIDEntries::executingImage

#### 8.45.2.2 FMSImageIdElement FMSImageIDEntries::imageIDElement[50]

#### 8.45.2.3 uint8_t FMSImageIDEntries::imageIDSize

#### 8.45.2.4 uint8_t FMSImageIDEntries::imageType

#### 8.45.2.5 uint8_t FMSImageIDEntries::maxImages

## 8.46 FMSImageList Struct Reference

### Data Fields

- uint8_t listSize
- FMSImageIDEntries imageIDEntries [2]

### 8.46.1 Detailed Description

This structure contains the Get Stored Images List

**Parameters**

| | |
|---|---|
| *listSize* | • The number of elements in the image list |
| *imageIDEntries* | • Array of ImageIDEntries Structure ( Max 2 entries ) |

### 8.46.2 Field Documentation

#### 8.46.2.1 FMSImageIDEntries FMSImageList::imageIDEntries[2]

#### 8.46.2.2 uint8_t FMSImageList::listSize

## 8.47 FMSPrefImageList Struct Reference

### Data Fields

- uint8_t listSize
- FMSImageElement listEntries [2]

### 8.47.1 Detailed Description

This structure contains the Preference Image List information

**Parameters**

| | |
|---|---|
| *listSize* | • The number of elements in the image list |
| *pListEntries* | • Array of Image entries with size provided by previous field<br>• See FMSImageElement |

### 8.47.2 Field Documentation

#### 8.47.2.1 FMSImageElement FMSPrefImageList::listEntries[2]

#### 8.47.2.2 uint8_t FMSPrefImageList::listSize

## 8.48 hdrSSInfo Struct Reference

### Data Fields

- int8_t rssi
- int16_t ecio
- uint8_t sinr
- int32_t io

### 8.48.1 Detailed Description

This structure contains the parameters for HDR Signal Strength Information

**Parameters**

| | |
|---|---|
| *rssi* | • RSSI in dBm (signed value).<br>• A value of -125 dBm or lower is used to indicate No Signal. |
| *ecio* | • ECIO value representing negative 0.5 dBm increments, i.e., 2 means -1 dBm (14 means -7 dBm, 63 means -31.5 dBm). |
| *sinr* | • SINR level.<br>• SINR is only applicable for 1xEV-DO.<br>• Valid levels are 0 to 8, where the maximum value for:<br>– 0 - SINR_LEVEL_0 is -9 dB<br>– 1 - SINR_LEVEL_1 is -6 dB<br>– 2 - SINR_LEVEL_2 is -4.5 dB<br>– 3 - SINR_LEVEL_3 is -3 dB<br>– 4 - SINR_LEVEL_4 is -2 dB<br>– 5 - SINR_LEVEL_5 is +1 dB<br>– 6 - SINR_LEVEL_6 is +3 dB<br>– 7 - SINR_LEVEL_7 is +6 dB<br>– 8 - SINR_LEVEL_8 is +9 dB<br>– 0xFF - Not Available |
| *io* | • Received IO in dBm.<br>• IO is only applicable for 1xEV-DO. |

### 8.48.2 Field Documentation

#### 8.48.2.1 int16_t hdrSSInfo::ecio

#### 8.48.2.2 int32_t hdrSSInfo::io

#### 8.48.2.3 int8_t hdrSSInfo::rssi

#### 8.48.2.4 uint8_t hdrSSInfo::sinr

## 8.49 image_info_t Struct Reference

### Data Fields

- uint8_t imageType
- uint8_t uniqueID [16]
- uint8_t buildIDLen
- uint8_t buildID [255]

### 8.49.1 Detailed Description

This structure is used to store image information

**Parameters**

| | |
|---|---|
| *imageType* | • Image Type<br>• Values:<br>– 0 - FW<br>– 1 - configuration |
| *uniqueID* | • Image Unique Identifier ( ASCII characters ) |
| *buildIDLen* | • Length of build ID string to follow<br>• If set to zero, build ID string will be blank |
| *buildID* | • String containing image information( ASCII characters )<br>• Maximum length of this string is 255 chars |

### 8.49.2 Field Documentation

#### 8.49.2.1 uint8_t image_info_t::buildID[255]

#### 8.49.2.2 uint8_t image_info_t::buildIDLen

#### 8.49.2.3 uint8_t image_info_t::imageType

#### 8.49.2.4 uint8_t image_info_t::uniqueID[16]

## 8.50 ims_AMRModeInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t amrMode

### 8.50.1 Detailed Description

This structure store information about AMR NB mode.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *amrMode* | Bitmask indicating AMR modes. Values:<br>• 0x1 – 4.75 kbps<br>• 0x2 – 5.15 kbps<br>• 0x4 – 5.9 kbps<br>• 0x8 – 6.17 kbps<br>• 0x10 – 7.4 kbps<br>• 0x20 – 7.95 kbps<br>• 0x40 – 10.2 kbps<br>• 0x80 – 12.2 kbps |

### 8.50.2 Field Documentation

#### 8.50.2.1 uint8_t ims_AMRModeInfo::amrMode

#### 8.50.2.2 uint8_t ims_AMRModeInfo::TlvPresent

## 8.51 ims_AMROctAlgnInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t amrOctAlgn

### 8.51.1 Detailed Description

This structure store information about AMR NB octet aligned.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *amrOctAlgn* | -1- if octet aligned -0- if octed not aligned |

### 8.51.2 Field Documentation

#### 8.51.2.1 uint8_t ims_AMROctAlgnInfo::amrOctAlgn

#### 8.51.2.2 uint8_t ims_AMROctAlgnInfo::TlvPresent

## 8.52 ims_AMRWBModeInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t amrWBMode

### 8.52.1 Detailed Description

This structure store information about AMR WB mode.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *amrWBMode* | Bitmask indicating AMR WB modes. Values:<br>• 0x1 – 6.60 kbps<br>• 0x2 – 8.85 kbps<br>• 0x4 – 12.65 kbps<br>• 0x8 – 14.25 kbps<br>• 0x10 – 15.85 kbps<br>• 0x20 – 18.25 kbps<br>• 0x40 – 19.85 kbps<br>• 0x80 – 23.05 kbps<br>• 0x100 – 23.85 kbps |

### 8.52.2 Field Documentation

#### 8.52.2.1 uint16_t ims_AMRWBModeInfo::amrWBMode

#### 8.52.2.2 uint8_t ims_AMRWBModeInfo::TlvPresent

## 8.53 ims_AMRWBOctAlgnInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t amrWBOctAlgn

### 8.53.1 Detailed Description

This structure store information about AMR WB octet aligned.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *amrWBOctAlgn* | -1- if octet aligned -0- if octed not aligned |

### 8.53.2 Field Documentation

#### 8.53.2.1 uint8_t ims_AMRWBOctAlgnInfo::amrWBOctAlgn

#### 8.53.2.2 uint8_t ims_AMRWBOctAlgnInfo::TlvPresent

## 8.54 ims_CSCFPortNameInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t cscfPortName [255]

### 8.54.1 Detailed Description

This structure hold parameters about CSCF port info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *cscfPortName* | -CSCF port name string |

### 8.54.2 Field Documentation

#### 8.54.2.1 uint8_t ims_CSCFPortNameInfo::cscfPortName[255]

#### 8.54.2.2 uint8_t ims_CSCFPortNameInfo::TlvPresent

## 8.55 ims_EnabAMRWBInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t amrWBEnable

### 8.55.1 Detailed Description

This structure store information about Enable AMR WB.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *amrWBEnable* | -1- Enable -0- Disable |

### 8.55.2 Field Documentation

#### 8.55.2.1 uint8_t ims_EnabAMRWBInfo::amrWBEnable

#### 8.55.2.2 uint8_t ims_EnabAMRWBInfo::TlvPresent

## 8.56 ims_EnabSCRAMRInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t scrAmrEnable

### 8.56.1 Detailed Description

This structure store information about Enable SCR AMR .

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *scrAmrEnable* | -1- Enable -0- Disable |

### 8.56.2 Field Documentation

#### 8.56.2.1 uint8_t ims_EnabSCRAMRInfo::scrAmrEnable

#### 8.56.2.2 uint8_t ims_EnabSCRAMRInfo::TlvPresent

## 8.57 ims_EnabSCRAMRWBInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t scrAmrWBEnable

### 8.57.1 Detailed Description

This structure store information about Enable SCR AMR WB.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *scrAmrWB-Enable* | -1- Enable -0- Disable |

### 8.57.2 Field Documentation

#### 8.57.2.1 uint8_t ims_EnabSCRAMRWBInfo::scrAmrWBEnable

#### 8.57.2.2 uint8_t ims_EnabSCRAMRWBInfo::TlvPresent

## 8.58 ims_IMSDomainInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t imsDomainName [255]

### 8.58.1 Detailed Description

This structure hold parameters about IMS domain info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *imsDomain-Name* | -IMS domain name string |

### 8.58.2 Field Documentation

#### 8.58.2.1 uint8_t ims_IMSDomainInfo::imsDomainName[255]

#### 8.58.2.2 uint8_t ims_IMSDomainInfo::TlvPresent

## 8.59 ims_IMSTestModeInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t imsTestMode

### 8.59.1 Detailed Description

This structure hold parameters about ims test mode info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *imsTestMode* | -1- if IMS test mode is enabled -0- if IMS test mode is disabled |

### 8.59.2 Field Documentation

#### 8.59.2.1 uint8_t ims_IMSTestModeInfo::imsTestMode

#### 8.59.2.2 uint8_t ims_IMSTestModeInfo::TlvPresent

## 8.60 ims_MinSessExpInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t minSessExp

### 8.60.1 Detailed Description

This structure store information about minimum session expiry info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *minSessExp* | • minimum session expiry in seconds |

### 8.60.2 Field Documentation

#### 8.60.2.1 uint16_t ims_MinSessExpInfo::minSessExp

#### 8.60.2.2 uint8_t ims_MinSessExpInfo::TlvPresent

## 8.61 ims_PCSCFPortInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t priCSCFPort

### 8.61.1 Detailed Description

This structure hold parameters about primary CSCF port info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *priCSCFPort* | • primary CSCF port number |

### 8.61.2 Field Documentation

#### 8.61.2.1 uint16_t ims_PCSCFPortInfo::priCSCFPort

#### 8.61.2.2 uint8_t ims_PCSCFPortInfo::TlvPresent

## 8.62 ims_PhCtxtURIInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t PhCtxtURI [255]

### 8.62.1 Detailed Description

This structure hold parameters about Phone context URI info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *PhCtxtURI* | -Phone context URI string |

### 8.62.2 Field Documentation

#### 8.62.2.1 uint8_t ims_PhCtxtURIInfo::PhCtxtURI[255]

#### 8.62.2.2 uint8_t ims_PhCtxtURIInfo::TlvPresent

## 8.63 ims_RngBkTmrInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t RingBkTmr

### 8.63.1 Detailed Description

This structure hold parameters about Ring back timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *RingBkTmr* | • Duration, in seconds, of the Ringback timer |

### 8.63.2 Field Documentation

#### 8.63.2.1 uint16_t ims_RngBkTmrInfo::RingBkTmr

#### 8.63.2.2 uint8_t ims_RngBkTmrInfo::TlvPresent

## 8.64 ims_RngTmrInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t RingTmr

### 8.64.1 Detailed Description

This structure hold parameters about Ring timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *RingTmr* | • Duration, in seconds, of the Ring timer |

### 8.64.2 Field Documentation

#### 8.64.2.1 uint16_t ims_RngTmrInfo::RingTmr

#### 8.64.2.2 uint8_t ims_RngTmrInfo::TlvPresent

## 8.65 ims_RTPRTCPInactTmrDurInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t InactTmr

### 8.65.1 Detailed Description

This structure hold parameters about RTP/RTCP timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *InactTmr* | • Duration, in seconds, of the RTP/RTCP inactivity timer |

### 8.65.2 Field Documentation

#### 8.65.2.1 uint16_t ims_RTPRTCPInactTmrDurInfo::InactTmr

#### 8.65.2.2 uint8_t ims_RTPRTCPInactTmrDurInfo::TlvPresent

## 8.66 ims_SessDurInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t sessExp

### 8.66.1 Detailed Description

This structure store information about session duration info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *sessExp* | • session duration in seconds |

### 8.66.2 Field Documentation

#### 8.66.2.1 uint16_t ims_SessDurInfo::sessExp

#### 8.66.2.2 uint8_t ims_SessDurInfo::TlvPresent

## 8.67 ims_SigCompEnInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t SigCompEn

### 8.67.1 Detailed Description

This structure hold parameters about SigComp status.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *SigCompEn* | -1- if SigComp is enabled -0- if IMS SigComp is disabled |

### 8.67.2 Field Documentation

#### 8.67.2.1 uint8_t ims_SigCompEnInfo::SigCompEn

#### 8.67.2.2 uint8_t ims_SigCompEnInfo::TlvPresent

## 8.68 ims_SIPPortInfo Struct Reference

### Data Fields

- uint8_t TlvPresent

- uint16_t SIPLocalPort

### 8.68.1 Detailed Description

This structure stores information about SIP port info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *SIPLocalPort* | -SIP local port |

### 8.68.2 Field Documentation

#### 8.68.2.1 uint16_t ims_SIPPortInfo::SIPLocalPort

#### 8.68.2.2 uint8_t ims_SIPPortInfo::TlvPresent

## 8.69 ims_SIPRegnTmrInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t tmrSIPRegn

### 8.69.1 Detailed Description

This structure stores information about SIP registration timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *tmrSIPRegn* | -Initial SIP registration duration, in seconds |

### 8.69.2 Field Documentation

#### 8.69.2.1 uint8_t ims_SIPRegnTmrInfo::TlvPresent

#### 8.69.2.2 uint32_t ims_SIPRegnTmrInfo::tmrSIPRegn

## 8.70 ims_SMSFmtInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t smsFormat

### 8.70.1 Detailed Description

This structure hold parameters about SMS format info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *smsFormat* | • 0- IMS_SETTINGS_SMS_FORMAT_3GPP2<br>• 1- IMS_SETTINGS_SMS_FORMAT_3GPP |

### 8.70.2 Field Documentation

#### 8.70.2.1 uint8_t ims_SMSFmtInfo::smsFormat

#### 8.70.2.2 uint8_t ims_SMSFmtInfo::TlvPresent

## 8.71 ims_SMSoIPNwInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t smsoIPNW

### 8.71.1 Detailed Description

This structure hold parameters about SMS over IP network.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *smsoIPNW* | -1- if MO SMS turned on -0- if MO SMS turned off |

### 8.71.2 Field Documentation

#### 8.71.2.1 uint8_t ims_SMSoIPNwInfo::smsoIPNW

#### 8.71.2.2 uint8_t ims_SMSoIPNwInfo::TlvPresent

## 8.72 ims_SubscrTmrInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t subscrTmr

### 8.72.1 Detailed Description

This structure stores information about subscriber timer.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *subscrTmr* | -Duration, in seconds, of the subscription by the UE for IMS registration notifications. |

### 8.72.2 Field Documentation

#### 8.72.2.1 uint32_t ims_SubscrTmrInfo::subscrTmr

#### 8.72.2.2 uint8_t ims_SubscrTmrInfo::TlvPresent

## 8.73 ims_TmrT1Info Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t tmrT1

### 8.73.1 Detailed Description

This structure stores information about timer T1.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *tmrT1* | -RTT estimate, in milliseconds. |

### 8.73.2 Field Documentation

#### 8.73.2.1 uint8_t ims_TmrT1Info::TlvPresent

#### 8.73.2.2 uint32_t ims_TmrT1Info::tmrT1

## 8.74 ims_TmrT2Info Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t tmrT2

### 8.74.1 Detailed Description

This structure stores information about timer T2.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *tmrT2* | -Maximum retransmit interval, in msec, for non-invite requests and invite responses. |

### 8.74.2 Field Documentation

#### 8.74.2.1 uint8_t ims_TmrT2Info::TlvPresent

#### 8.74.2.2 uint32_t ims_TmrT2Info::tmrT2

## 8.75 ims_TmrTfInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t tmrTf

### 8.75.1 Detailed Description

This structure store information about timer TF.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *tmrTf* | -Non-invite transaction timeout timer, in msec |

### 8.75.2 Field Documentation

#### 8.75.2.1 uint8_t ims_TmrTfInfo::TlvPresent

#### 8.75.2.2 uint32_t ims_TmrTfInfo::tmrTf

## 8.76 imsa_IMSFailErrCodeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t ImsFailErrCode

### 8.76.1 Detailed Description

This structure hold parameters about IMS failure code for PDP connection.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |

| | |
|---|---|
| *ImsFailErrCode* | • 0 -IMSA_PDP_STATUS_OTHER_FAILURE – Generic failure reason<br>• 1 -IMSA_PDP_STATUS_OPTION_UNSUBSCRIBED – Option is unsubscribed<br>• 2 -IMSA_PDP_STATUS_UNKNOWN_PDP – PDP was unknown |

### 8.76.2 Field Documentation

#### 8.76.2.1 uint32_t imsa_IMSFailErrCodeTlv::ImsFailErrCode

#### 8.76.2.2 uint8_t imsa_IMSFailErrCodeTlv::TlvPresent

## 8.77 imsa_IMSRegStatusErrorCodeInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t ErrorCode

### 8.77.1 Detailed Description

This structure hold parameters about IMS registration error info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *ErrorCode* | Error codes when registration status is IMSA_STATUS_NOT_REGISTERED<br>• Values:<br>• 3xx – Redirection responses<br>• 4xx – Client failure responses<br>• 5xx – Server failure responses<br>• 6xx – Global failure responses |

### 8.77.2 Field Documentation

#### 8.77.2.1 uint16_t imsa_IMSRegStatusErrorCodeInfo::ErrorCode

#### 8.77.2.2 uint8_t imsa_IMSRegStatusErrorCodeInfo::TlvPresent

## 8.78 imsa_IMSRegStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t ImsRegistered

### 8.78.1 Detailed Description

This structure hold parameters about IMS registration info (decprecated tlv).

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *ImsRegistered* | • Values:<br>• 0 -Not registered<br>• 1- Registered |

### 8.78.2 Field Documentation

#### 8.78.2.1 uint8_t imsa_IMSRegStatusInfo::ImsRegistered

#### 8.78.2.2 uint8_t imsa_IMSRegStatusInfo::TlvPresent

## 8.79 imsa_NewIMSRegStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t ImsRegStatus

### 8.79.1 Detailed Description

This structure hold parameters about IMS registration info (new tlv).

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *ImsRegStatus* | • Values:<br>• 0 -IMSA_STATUS_NOT_REGISTERED- Not registered for IMS<br>• 1- IMSA_STATUS_REGISTERING – Registering for IMS<br>• 2- IMSA_STATUS_REGISTERED – Registered for IMS |

### 8.79.2 Field Documentation

#### 8.79.2.1 uint32_t imsa_NewIMSRegStatusInfo::ImsRegStatus

#### 8.79.2.2 uint8_t imsa_NewIMSRegStatusInfo::TlvPresent

## 8.80 imsa_RatHandoverStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t RatHandoverStatus
- uint32_t SourceRAT
- uint32_t TargetRAT
- uint8_t ErrorCodeLen
- uint8_t ErrorCodeData [256]

### 8.80.1 Detailed Description

This structure hold parameters about RAT handover status.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *RatHandover-Status* | • RAT handover status |
| *SourceRAT* | • source RAT info: IWLAN/WWAN |
| *TargetRAT* | • target TAT info: IWLAN/WWAN |
| *ErrorCodeLen* | • error code length |
| *ErrorCodeData* | • handover failure code string when status is IMSA_STATUS_RAT_HO_FAILURE |

### 8.80.2 Field Documentation

#### 8.80.2.1 uint8_t imsa_RatHandoverStatusInfo::ErrorCodeData[256]

#### 8.80.2.2 uint8_t imsa_RatHandoverStatusInfo::ErrorCodeLen

#### 8.80.2.3 uint32_t imsa_RatHandoverStatusInfo::RatHandoverStatus

#### 8.80.2.4 uint32_t imsa_RatHandoverStatusInfo::SourceRAT

#### 8.80.2.5 uint32_t imsa_RatHandoverStatusInfo::TargetRAT

#### 8.80.2.6 uint8_t imsa_RatHandoverStatusInfo::TlvPresent

## 8.81 imsa_SmsRatInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t SmsRatVal

### 8.81.1 Detailed Description

This structure hold parameters about SMS RAT info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *SmsRatVal* | • Values:<br>• 0-IMSA_WLAN – IMS service is registered on WLAN<br>• 1-IMSA_WWAN – IMS service is registered on WWAN<br>• 2-IMSA_IWLAN – IMS service is registered on interworking WLAN |

### 8.81.2 Field Documentation

#### 8.81.2.1 uint32_t imsa_SmsRatInfo::SmsRatVal

#### 8.81.2.2 uint8_t imsa_SmsRatInfo::TlvPresent

## 8.82 imsa_SmsSvcStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t SmsSvcStatus

### 8.82.1 Detailed Description

This structure hold parameters about SMS service info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *SmsSvcStatus* | • Values: -0 – IMS SMS service is not available -1 – IMS SMS is in limited service -2 – IMS SMS is in full service |

### 8.82.2 Field Documentation

#### 8.82.2.1 uint32_t imsa_SmsSvcStatusInfo::SmsSvcStatus

#### 8.82.2.2 uint8_t imsa_SmsSvcStatusInfo::TlvPresent

## 8.83 imsa_UtRatInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t UtRatVal

### 8.83.1 Detailed Description

This structure hold parameters about UT RAT info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *UtRatVal* | • Values:<br>• 0-IMSA_WLAN – IMS service is registered on WLAN<br>• 1-IMSA_WWAN – IMS service is registered on WWAN<br>• 2-IMSA_IWLAN – IMS service is registered on interworking WLAN |

### 8.83.2 Field Documentation

#### 8.83.2.1 uint8_t imsa_UtRatInfo::TlvPresent

#### 8.83.2.2 uint32_t imsa_UtRatInfo::UtRatVal

## 8.84 imsa_UtSvcStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t UtSvcStatus

### 8.84.1 Detailed Description

This structure hold parameters about UT service info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *UtSvcStatus* | • Values: -0 – IMS UT service is not available -2 – IMS UT is in full service |

### 8.84.2 Field Documentation

#### 8.84.2.1 uint8_t imsa_UtSvcStatusInfo::TlvPresent

#### 8.84.2.2 uint32_t imsa_UtSvcStatusInfo::UtSvcStatus

## 8.85 imsa_VoipRatInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t VoipRatVal

### 8.85.1 Detailed Description

This structure hold parameters about VOIP RAT info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *VoipRatVal* | • Values:<br>• 0-IMSA_WLAN – IMS service is registered on WLAN<br>• 1-IMSA_WWAN – IMS service is registered on WWAN<br>• 2-IMSA_IWLAN – IMS service is registered on interworking WLAN |

### 8.85.2 Field Documentation

#### 8.85.2.1 uint8_t imsa_VoipRatInfo::TlvPresent

#### 8.85.2.2 uint32_t imsa_VoipRatInfo::VoipRatVal

## 8.86 imsa_VoipSvcStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t VoipSvcStatus

### 8.86.1 Detailed Description

This structure hold parameters about VOIP service info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *VoipSvcStatus* | • Values: -0 – IMS VOIP service is not available -2 – IMS VOIP is in full service |

### 8.86.2 Field Documentation

#### 8.86.2.1 uint8_t imsa_VoipSvcStatusInfo::TlvPresent

#### 8.86.2.2 uint32_t imsa_VoipSvcStatusInfo::VoipSvcStatus

## 8.87 imsa_VtRatInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t VtRatVal

### 8.87.1 Detailed Description

This structure hold parameters about VT RAT info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *VtRatVal* | • Values:<br>• 0-IMSA_WLAN – IMS service is registered on WLAN<br>• 1-IMSA_WWAN – IMS service is registered on WWAN<br>• 2-IMSA_IWLAN – IMS service is registered on interworking WLAN |

### 8.87.2 Field Documentation

#### 8.87.2.1 uint8_t imsa_VtRatInfo::TlvPresent

#### 8.87.2.2 uint32_t imsa_VtRatInfo::VtRatVal

## 8.88 imsa_VtSvcStatusInfo Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t VtSvcStatus

### 8.88.1 Detailed Description

This structure hold parameters about VT service info.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *VtSvcStatus* | • Values: -0 – IMS VT service is not available -2 – IMS VT is in full service |

### 8.88.2 Field Documentation

#### 8.88.2.1 uint8_t imsa_VtSvcStatusInfo::TlvPresent

#### 8.88.2.2 uint32_t imsa_VtSvcStatusInfo::VtSvcStatus

## 8.89 ipv6AddressInfo Struct Reference

### Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

### 8.89.1 Detailed Description

This structure contains the IPv6 address information

**Parameters**

| | |
|---|---|
| *IPV6PrefixLen* | Length of the received IPv6 address<br>• 0xff - Invalid IPv6 address information. |
| *IPAddressV6* | IPv6 address(in network byte order) |

### 8.89.2 Field Documentation

#### 8.89.2.1 uint16_t ipv6AddressInfo::IPAddressV6[8]

#### 8.89.2.2 uint8_t ipv6AddressInfo::IPV6PrefixLen

## 8.90 LibPackGPRSRequestedQoS Struct Reference

### Data Fields

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

### 8.90.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

**Parameters**

| | |
|---|---|
| *precedence-Class* | • Precedence class |
| *delayClass* | • Delay class |

| | |
|---|---|
| *reliabilityClass* | • Reliability class |
| *peak-Throughput-Class* | • Peak throughput class |
| *mean-Throughput-Class* | • Mean throughput class |

### 8.90.2 Field Documentation

#### 8.90.2.1 uint32_t LibPackGPRSRequestedQoS::delayClass

#### 8.90.2.2 uint32_t LibPackGPRSRequestedQoS::meanThroughputClass

#### 8.90.2.3 uint32_t LibPackGPRSRequestedQoS::peakThroughputClass

#### 8.90.2.4 uint32_t LibPackGPRSRequestedQoS::precedenceClass

#### 8.90.2.5 uint32_t LibPackGPRSRequestedQoS::reliabilityClass

## 8.91 LibpackProfile3GPP Struct Reference

### Data Fields

- uint8_t * pProfilename
- uint16_t * pProfilenameSize
- uint8_t * pPDPtype
- uint8_t * pPdpHdrCompType
- uint8_t * pPdpDataCompType
- uint8_t * pAPNName
- uint16_t * pAPNnameSize
- uint32_t * pPriDNSIPv4AddPref
- uint32_t * pSecDNSIPv4AddPref
- LibPackUMTSQoS * pUMTSReqQoS
- LibPackUMTSQoS * pUMTSMinQoS
- LibPackGPRSRequestedQoS * pGPRSRequestedQos
- LibPackGPRSRequestedQoS * pGPRSMinimumQoS
- uint8_t * pUsername
- uint16_t * pUsernameSize
- uint8_t * pPassword
- uint16_t * pPasswordSize
- uint8_t * pAuthenticationPref
- uint32_t * pIPv4AddrPref
- uint8_t * pPcscfAddrUsingPCO
- uint8_t * pPdpAccessConFlag
- uint8_t * pPcscfAddrUsingDhcp
- uint8_t * pImCnFlag
- LibPackTFTIDParams * pTFTID1Params
- LibPackTFTIDParams * pTFTID2Params
- uint8_t * pPdpContext
- uint8_t * pSecondaryFlag

- uint8_t * pPrimaryID
- uint16_t * pIPv6AddPref
- LibPackUMTSReqQoSSigInd * pUMTSReqQoSSigInd
- LibPackUMTSReqQoSSigInd * pUMTSMinQosSigInd
- uint16_t * pPriDNSIPv6addpref
- uint16_t * pSecDNSIPv6addpref
- uint8_t * pAddrAllocPref
- LibPackQosClassID * pQosClassID
- uint8_t * pAPNDisabledFlag
- uint32_t * pPDNInactivTimeout
- uint8_t * pAPNClass
- uint8_t * pSupportEmergencyCalls

### 8.91.1 Detailed Description

This structure contains Input parameters of unpack_wds_SLQSGetProfileSettings_t Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.

**Parameters**

| | |
|---|---|
| *pProfileName* | • One or more uint8_ts describing the profile<br>• NULL pointer - Invalid data. |
| *pProfilename-Size;* | • This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 uint8_ts.<br>• NULL pointer - Invalid data. |
| *pPDPType* | • 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<br>– 0x00 - PDP-IP (IPv4)<br>– 0x01 - PDP-PPP<br>– 0x02 - PDP-IPV6<br>– 0x03 - PDP-IPV4V6<br>• NULL pointer - Invalid data. |
| *pPdpHdrComp-Type* | • PDP header compression type<br>– 0 - PDP header compression is OFF<br>– 1 - Manufacturer preferred compression<br>– 2 - PDP header compression based on RFC 1144<br>– 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095<br>• NULL pointer - Invalid data. |
| *pPdpDataComp-Type* | • PDP data compression type<br>– 0 - PDP data compression is OFF<br>– 1 - Manufacturer preferred compression<br>– 2 - V.42BIS data compression<br>– 3 - V.44 data compression<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pAPNName* | • Access point name<br>• NULL pointer - Invalid data. |
| *pAPNnameSize;* | • This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 uint8_ts.<br>• NULL pointer - Invalid data. |
| *pPriDNSIPv4-AddPref* | • Primary DNS IPv4 Address Preference<br>• NULL pointer - Invalid data. |
| *pSecDNSIPv4-AddPref* | • Secondary DNS IPv4 Address Preference<br>• NULL pointer - Invalid data. |
| *pUMTSReqQoS* | • UMTS Requested QoS<br>• NULL pointer - Invalid data. |
| *pUMTSMinQoS* | • UMTS Minimum QoS<br>• NULL pointer - Invalid data. |
| *pGPRS-RequestedQoS* | • GPRS Requested QoS<br>• NULL pointer - Invalid data. |
| *pGPRS-MinimumQoS* | • GPRS Minimum QoS<br>• NULL pointer - Invalid data. |
| *pUsername* | • User name<br>• NULL pointer - Invalid data. |
| *pUsernameSize;* | • This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 uint8_ts.<br>• NULL pointer - Invalid data. |
| *pPassword* | • Password<br>• NULL pointer - Invalid data. |
| *pPasswordSize;* | • This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 uint8_ts.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pAuthentication-Pref* | • Authentication Preference<br>– Bit map that indicates the authentication algorithm preference<br>* Bit 0 - PAP preference<br>· 0 - PAP is never performed<br>· 1 - PAP may be performed<br>* Bit 1 - CHAP preference<br>· 0 - CHAP is never performed<br>· 1 - CHAP may be performed<br>* 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.<br>• NULL pointer - Invalid data. |
| *pIPv4AddrPref* | • IPv4 Address Preference<br>• NULL pointer - Invalid data. |
| *pPcscfAddr-UsingPCO* | • P-CSCF Address using PCO Flag<br>– 1 - (TRUE) implies request PCSCF address using PCO<br>– 0 - (FALSE) implies do not request By default, this value is 0<br>• NULL pointer - Invalid data. |
| *pPdpAccess-ConFlag* | • PDP access control flag<br>– 0 - PDP access control none<br>– 1 - PDP access control reject<br>– 2 - PDP access control permission<br>• NULL pointer - Invalid data. |
| *pPcscfAddr-UsingDhcp* | • P-CSCF address using DHCP<br>– 1 - (TRUE) implies Request PCSCF address using DHCP<br>– 0 - (FALSE) implies do not request By default, value is 0<br>• NULL pointer - Invalid data. |
| *pImCnFlag* | • IM CN flag<br>– 1 - (TRUE) implies request IM CN flag for this profile<br>– 0 - (FALSE) implies do not request IM CN flag for this profile<br>• NULL pointer - Invalid data. |
| *pTFTID1Params* | • Traffic Flow Template<br>• NULL pointer - Invalid data. |
| *pTFTID2Params* | • Traffic Flow Template<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pPdpContext* | • PDP context number<br>• NULL pointer - Invalid data. |
| *pSecondaryFlag* | • PDP context secondary flag<br>– 1 - (TRUE) implies this is secondary profile<br>– 0 - (FALSE) implies this is not secondary profile<br>• NULL pointer - Invalid data. |
| *pPrimaryID* | • PDP context primary ID<br>• function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device<br>• NULL pointer - Invalid data. |
| *pIPv6AddPref* | • 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<br>• NULL pointer - Invalid data. |
| *pUMTSReqQoS-SigInd* | • UMTS requested QoS with Signalling Indication flag<br>• NULL pointer - Invalid data. |
| *pUMTSMinQoS-SigInd* | • UMTS minimum QoS with Signalling Indication flag<br>• NULL pointer - Invalid data. |
| *pPrimaryDNSI-Pv6addpref* | • Primary DNS IPv6 address preference<br>– 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<br>• NULL pointer - Invalid data. |
| *pSecondaryDN-SIPv6addpref* | • Secondary DNS IPv6 address preference<br>• NULL pointer - Invalid data. |
| *paddrAllocation-Pref* | • DHCP/NAS preference<br>– This enumerated value may be used to indicate the address allocation preference<br>* 0 - NAS signaling is used for address allocation<br>* 1 - DHCP is used for address allocation<br>• NULL pointer - Invalid data. |
| *pQosClassID* | • 3GPP LTE QoS parameters<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pAPNDisabled-Flag* | • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled<br>• If set, the profile can not be used for making data calls<br>• Any data call is failed locally<br>• Values:<br>– 0 - FALSE(default)<br>– 1 - True<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().<br>• NULL pointer - Invalid data. |
| *pPDNInactiv-Timeout* | • Optional 4 uint8_ts indicating the duration of inactivity timer in seconds<br>• 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<br>• Default value of zero indicates infinite value<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().<br>• NULL pointer - Invalid data. |
| *pAPNClass* | • Optional 1 uint8_t numeric identifier representing the APN in profile<br>• Can be set and queried but is not used by the modem<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings().<br>• NULL pointer - Invalid data. |
| *pSupport-EmergencyCalls* | • Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled<br>• Can be queried to get current status<br>• Values:<br>– 0 - FALSE(default)<br>– 1 - TRUE<br>• NULL pointer - Invalid data. |

### 8.91.2 Field Documentation

#### 8.91.2.1 uint8_t* LibpackProfile3GPP::pAddrAllocPref

#### 8.91.2.2 uint8_t* LibpackProfile3GPP::pAPNClass

#### 8.91.2.3 uint8_t* LibpackProfile3GPP::pAPNDisabledFlag

#### 8.91.2.4 uint8_t* LibpackProfile3GPP::pAPNName

#### 8.91.2.5 uint16_t* LibpackProfile3GPP::pAPNnameSize

#### 8.91.2.6 uint8_t* LibpackProfile3GPP::pAuthenticationPref

#### 8.91.2.7 LibPackGPRSRequestedQoS* LibpackProfile3GPP::pGPRSMinimumQoS

8.91.2.8 **LibPackGPRSRequestedQoS*** LibpackProfile3GPP::pGPRSRequestedQos

8.91.2.9 uint8_t* LibpackProfile3GPP::plmCnFlag

8.91.2.10 uint32_t* LibpackProfile3GPP::pIPv4AddrPref

8.91.2.11 uint16_t* LibpackProfile3GPP::pIPv6AddPref

8.91.2.12 uint8_t* LibpackProfile3GPP::pPassword

8.91.2.13 uint16_t* LibpackProfile3GPP::pPasswordSize

8.91.2.14 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingDhcp

8.91.2.15 uint8_t* LibpackProfile3GPP::pPcscfAddrUsingPCO

8.91.2.16 uint32_t* LibpackProfile3GPP::pPDNInactivTimeout

8.91.2.17 uint8_t* LibpackProfile3GPP::pPdpAccessConFlag

8.91.2.18 uint8_t* LibpackProfile3GPP::pPdpContext

8.91.2.19 uint8_t* LibpackProfile3GPP::pPdpDataCompType

8.91.2.20 uint8_t* LibpackProfile3GPP::pPdpHdrCompType

8.91.2.21 uint8_t* LibpackProfile3GPP::pPDPtype

8.91.2.22 uint32_t* LibpackProfile3GPP::pPriDNSIPv4AddPref

8.91.2.23 uint16_t* LibpackProfile3GPP::pPriDNSIPv6addpref

8.91.2.24 uint8_t* LibpackProfile3GPP::pPrimaryID

8.91.2.25 uint8_t* LibpackProfile3GPP::pProfilename

8.91.2.26 uint16_t* LibpackProfile3GPP::pProfilenameSize

8.91.2.27 **LibPackQosClassID*** LibpackProfile3GPP::pQosClassID

8.91.2.28 uint32_t* LibpackProfile3GPP::pSecDNSIPv4AddPref

8.91.2.29 uint16_t* LibpackProfile3GPP::pSecDNSIPv6addpref

8.91.2.30 uint8_t* LibpackProfile3GPP::pSecondaryFlag

8.91.2.31 uint8_t* LibpackProfile3GPP::pSupportEmergencyCalls

8.91.2.32 **LibPackTFTIDParams*** LibpackProfile3GPP::pTFTID1Params

8.91.2.33 **LibPackTFTIDParams*** LibpackProfile3GPP::pTFTID2Params

8.91.2.34 **LibPackUMTSQoS*** LibpackProfile3GPP::pUMTSMinQoS

8.91.2.35 **LibPackUMTSReqQoSSigInd*** LibpackProfile3GPP::pUMTSMinQosSigInd

#### 8.91.2.36 LibPackUMTSQoS* LibpackProfile3GPP::pUMTSReqQoS

#### 8.91.2.37 LibPackUMTSReqQoSSigInd* LibpackProfile3GPP::pUMTSReqQoSSigInd

#### 8.91.2.38 uint8_t* LibpackProfile3GPP::pUsername

#### 8.91.2.39 uint16_t* LibpackProfile3GPP::pUsernameSize

## 8.92 LibpackProfile3GPP2 Struct Reference

### Data Fields

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

### 8.92.1 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**

| | |
|---|---|
| *pNegoDnsSrvr-Pref* | • Negotiate DNS Server Preference<br>– 1 - (TRUE)implies request DNS addresses from the PDSN<br>– 0 - (FALSE)implies do not request DNS addresses from the PDSN<br>– Default value is 1 (TRUE) |
| *pPppSessClose-TimerDO* | • PPP Session Close Timer for DO<br>– Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down |
| *pPppSessClose-Timer1x* | • PPP Session Close Timer for 1X<br>– Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down |
| *pAllowLinger* | • Allow/disallow lingering of interface<br>– 1 -(TRUE) implies allow lingering<br>– 0 -(FALSE) implies do not allow lingering |
| *pLcpAckTimeout* | • LCP ACK Timeout<br>– Value of LCP ACK Timeout in milliseconds |
| *pIpcpAck-Timeout* | • IPCP ACK Timeout<br>– Value of IPCP ACK Timeout in milliseconds |
| *pAuthTimeout* | • AUTH Timeout<br>– Value of Authentication Timeout in milliseconds |
| *pLcpCreqRetry-Count* | • LCP Configuration Request Retry Count |
| *pIpcpCreqRetry-Count* | • IPCP Configuration Request Retry Count |
| *pAuthRetry-Count* | • Authentication Retry Count value |
| *pAuthProtocol* | • Authentication Protocol<br>– 1 - PAP<br>– 2 - CHAP<br>– 3 - PAP or CHAP |

| | |
|---|---|
| *pUserId* | • User ID to be used during data network authentication<br>• maximum length allowed is 127 uint8_ts;<br>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value. |
| *pUserIdSize;* | • This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 uint8_ts. |
| *pAuthPassword* | • Password to be used during data network authentication;<br>• maximum length allowed is 127 uint8_ts<br>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value. |
| *pAuthPassword-Size;* | • This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 uint8_ts. |
| *pDataRate* | • Data Rate Requested<br>– 0 - Low (Low speed Service Options (SO15) only)<br>– 1 - Medium (SO33 + low R-SCH)<br>– 2 - High (SO33 + high R-SCH)<br>– Default is 2 |
| *pAppType* | • Application Type:<br>– 0x00000001 - Default Application Type<br>– 0x00000020 - LBS Application Type<br>– 0x00000040 - Tethered Application Type<br>– This parameter is not used while creating/modifying a profile |
| *pDataMode* | • Data Mode to use:<br>– 0 - CDMA or HDR (Hybrid 1X/1xEV-DO)<br>– 1 - CDMA Only (1X only)<br>– 2 - HDR Only (1xEV-DO only)<br>– Default is 0 |
| *pAppPriority* | • Application Priority<br>– Numerical 1 uint8_t value defining the application priority; higher value implies higher priority<br>– This parameter is not used while creating/modifying a profile |
| *pApnString* | • String representing the Access Point Name<br>• maximum length allowed is 100 uint8_ts<br>• QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long. |

| | |
|---|---|
| *pApnStringSize;* | • This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 uint8_ts. |
| *pPdnType* | • Packed Data Network Type Requested:<br>– 0 - IPv4 PDN Type<br>– 1 - IPv6 PDN Type<br>– 2 - IPv4 or IPv6 PDN Type<br>– 3 - Unspecified PDN Type (implying no preference) |
| *pIsPcscf-AddressNedded* | • This boolean value is used to control if PCSCF address is requested from PDSN<br>– 1 -(TRUE) implies request for PCSCF value from the PDSN<br>– 0 -(FALSE) implies do not request for PCSCF value from the PDSN |
| *pPrimaryV4Dns-Address* | • IPv4 Primary DNS address<br>– The Primary IPv4 DNS address that can be statically assigned to the UE |
| *pSecondaryV4-DnsAddress* | • IPv4 Secondary DNS address<br>– The Secondary IPv4 DNS address that can be statically assigned to the UE |
| *pPriV6Dns-Address* | • Primary IPv6 DNS address<br>– The Primary IPv6 DNS address that can be statically assigned to the UE |
| *pSecV6Dns-Address* | • Secondary IPv6 DNS address<br>– The Secondary IPv6 DNS address that can be statically assigned to the UE |
| *pRATType* | • Optional 1 uint8_t Flag indicating RAT Type<br>• Values:<br>– 1 - HRPD<br>– 2 - EHRPD<br>– 3 - HRPD_EHRPD<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). |
| *pAPNEnabled3-GPP2* | • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled<br>• If disabled, the profile can not be used for making data calls<br>• Values:<br>– 0 - Disabled<br>– 1 - Enabled(default value)<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). |

| | |
|---|---|
| *pPDNInactivTimeout3GPP2* | • Optional 4 uint8_ts indicating the duration of inactivity timer in seconds<br>• 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<br>• Default value of zero indicates infinite value<br>• This parameter is currently read only and can be read by using the function SLQSGetProfileSettings(). |
| *pAPNClass3GPP2* | • Optional 1 uint8_t numeric identifier representing the APN in profile<br>• Can be set and queried but is not used by the modem<br>• This parameter is currently read only and can be read by using the function SLQSGetProfileSettings(). |

### 8.92.2 Field Documentation

#### 8.92.2.1 uint8_t* LibpackProfile3GPP2::pAllowLinger

#### 8.92.2.2 uint8_t* LibpackProfile3GPP2::pAPNClass3GPP2

#### 8.92.2.3 uint8_t* LibpackProfile3GPP2::pAPNEnabled3GPP2

#### 8.92.2.4 uint8_t* LibpackProfile3GPP2::pApnString

#### 8.92.2.5 uint16_t* LibpackProfile3GPP2::pApnStringSize

#### 8.92.2.6 uint8_t* LibpackProfile3GPP2::pAppPriority

#### 8.92.2.7 uint32_t* LibpackProfile3GPP2::pAppType

#### 8.92.2.8 uint8_t* LibpackProfile3GPP2::pAuthPassword

#### 8.92.2.9 uint16_t* LibpackProfile3GPP2::pAuthPasswordSize

#### 8.92.2.10 uint8_t* LibpackProfile3GPP2::pAuthProtocol

#### 8.92.2.11 uint8_t* LibpackProfile3GPP2::pAuthRetryCount

#### 8.92.2.12 uint16_t* LibpackProfile3GPP2::pAuthTimeout

#### 8.92.2.13 uint8_t* LibpackProfile3GPP2::pDataMode

#### 8.92.2.14 uint8_t* LibpackProfile3GPP2::pDataRate

#### 8.92.2.15 uint16_t* LibpackProfile3GPP2::pIpcpAckTimeout

#### 8.92.2.16 uint8_t* LibpackProfile3GPP2::pIpcpCreqRetryCount

#### 8.92.2.17 uint8_t* LibpackProfile3GPP2::pIsPcscfAddressNedded

#### 8.92.2.18 uint16_t* LibpackProfile3GPP2::pLcpAckTimeout

#### 8.92.2.19 uint8_t* LibpackProfile3GPP2::pLcpCreqRetryCount

#### 8.92.2.20 uint8_t* LibpackProfile3GPP2::pNegoDnsSrvrPref

#### 8.92.2.21 uint32_t* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

#### 8.92.2.22 uint8_t* LibpackProfile3GPP2::pPdnType

#### 8.92.2.23 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimer1x

#### 8.92.2.24 uint32_t* LibpackProfile3GPP2::pPppSessCloseTimerDO

#### 8.92.2.25 uint32_t* LibpackProfile3GPP2::pPrimaryV4DnsAddress

#### 8.92.2.26 uint16_t* LibpackProfile3GPP2::pPriV6DnsAddress

#### 8.92.2.27 uint8_t* LibpackProfile3GPP2::pRATType

#### 8.92.2.28 uint32_t* LibpackProfile3GPP2::pSecondaryV4DnsAddress

#### 8.92.2.29 uint16_t* LibpackProfile3GPP2::pSecV6DnsAddress

#### 8.92.2.30 uint8_t* LibpackProfile3GPP2::pUserId

#### 8.92.2.31 uint16_t* LibpackProfile3GPP2::pUserIdSize

## 8.93 LibPackprofile_3GPP Struct Reference

### Data Fields

- uint8_t * pProfilename
- uint16_t * pProfilenameSize
- uint8_t * pPDPtype
- uint8_t * pPdpHdrCompType
- uint8_t * pPdpDataCompType
- uint8_t * pAPNName
- uint16_t * pAPNnameSize
- uint32_t * pPriDNSIPv4AddPref
- uint32_t * pSecDNSIPv4AddPref
- LibPackUMTSQoS * pUMTSReqQoS
- LibPackUMTSQoS * pUMTSMinQoS
- LibPackGPRSRequestedQoS * pGPRSRequestedQos
- LibPackGPRSRequestedQoS * pGPRSMinimumQoS
- uint8_t * pUsername
- uint16_t * pUsernameSize
- uint8_t * pPassword
- uint16_t * pPasswordSize
- uint8_t * pAuthenticationPref
- uint32_t * pIPv4AddrPref
- uint8_t * pPcscfAddrUsingPCO
- uint8_t * pPdpAccessConFlag
- uint8_t * pPcscfAddrUsingDhcp
- uint8_t * pImCnFlag
- LibPackTFTIDParams * pTFTID1Params
- LibPackTFTIDParams * pTFTID2Params

- uint8_t * pPdpContext
- uint8_t * pSecondaryFlag
- uint8_t * pPrimaryID
- uint16_t * pIPv6AddPref
- LibPackUMTSReqQoSSigInd * pUMTSReqQoSSigInd
- LibPackUMTSReqQoSSigInd * pUMTSMinQosSigInd
- uint16_t * pPriDNSIPv6addpref
- uint16_t * pSecDNSIPv6addpref
- uint8_t * pAddrAllocPref
- LibPackQosClassID * pQosClassID
- uint8_t * pAPNDisabledFlag
- uint32_t * pPDNInactivTimeout
- uint8_t * pAPNClass
- uint8_t * pSupportEmergencyCalls

### 8.93.1 Detailed Description

This structure contains Input/Output parameters of pack_wds_SLQSCreateProfile

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

**Parameters**

| | |
|---|---|
| *pProfileName* | • One or more bytes describing the profile |
| *pProfilenameSize;* | • This parameter is an input parameter and should be initialised to the size of pProfileName field. Size of this parameter is 2 bytes. |
| *pPDPType* | • 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<br>– 0x00 - PDP-IP (IPv4)<br>– 0x01 - PDP-PPP<br>– 0x02 - PDP-IPV6<br>– 0x03 - PDP-IPV4V6 |
| *pPdpHdrCompType* | • PDP header compression type<br>– 0 - PDP header compression is OFF<br>– 1 - Manufacturer preferred compression<br>– 2 - PDP header compression based on RFC 1144<br>– 3 - PDP header compression based on RFC 25074 PDP header compression based on RFC 3095 |
| *pPdpDataCompType* | • PDP data compression type<br>– 0 - PDP data compression is OFF<br>– 1 - Manufacturer preferred compression<br>– 2 - V.42BIS data compression<br>– 3 - V.44 data compression |

| | |
|---|---|
| *pAPNName* | • Access point name |
| *pAPNnameSize;* | • This parameter is an input parameter and should be initialised to the size of pAPNName field. Size of this parameter is 2 bytes. |
| *pPriDNSIPv4-AddPref* | • Primary DNS IPv4 Address Preference |
| *pSecDNSIPv4-AddPref* | • Secondary DNS IPv4 Address Preference |
| *pUMTSReqQoS* | • UMTS Requested QoS |
| *pUMTSMinQoS* | • UMTS Minimum QoS |
| *pGPRS-RequestedQoS* | • GPRS Minimum QoS |
| *pUsername* | • User name |
| *pUsernameSize;* | • This parameter is an input parameter and should be initialised to the size of pUsername field. Size of this parameter is 2 bytes. |
| *pPassword* | • Password |
| *pPasswordSize;* | • This parameter is an input parameter and should be initialised to the size of pPassword field. Size of this parameter is 2 bytes. |
| *pAuthentication-Pref* | • Authentication Preference<br>– Bit map that indicates the authentication algorithm preference<br>* Bit 0 - PAP preference<br>· 0 - PAP is never performed<br>· 1 - PAP may be performed<br>* Bit 1 - CHAP preference<br>· 0 - CHAP is never performed<br>· 1 - CHAP may be performed<br>* 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. |
| *pIPv4AddrPref* | • IPv4 Address Preference |

| | |
|---|---|
| *pPcscfAddr-UsingPCO* | • P-CSCF Address using PCO Flag<br>– 1 - (TRUE) implies request PCSCF address using PCO<br>– 0 - (FALSE) implies do not request By default, this value is 0 |
| *pPdpAccess-ConFlag* | • PDP access control flag<br>– 0 - PDP access control none<br>– 1 - PDP access control reject<br>– 2 - PDP access control permission |
| *pPcscfAddr-UsingDhcp* | • P-CSCF address using DHCP<br>– 1 - (TRUE) implies Request PCSCF address using DHCP<br>– 0 - (FALSE) implies do not request By default, value is 0 |
| *pImCnFlag* | • IM CN flag<br>– 1 - (TRUE) implies request IM CN flag for this profile<br>– 0 - (FALSE) implies do not request IM CN flag for this profile |
| *pTFTID1Params* | • Traffic Flow Template |
| *pTFTID2Params* | • Traffic Flow Template |
| *pPdpContext* | • PDP context number |
| *pSecondaryFlag* | • PDP context secondary flag<br>– 1 - (TRUE) implies this is secondary profile<br>– 0 - (FALSE) implies this is not secondary profile |
| *pPrimaryID* | • PDP context primary ID<br>• function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device |
| *pIPv6AddPref* | • 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 |
| *pUMTSReqQoS-SigInd* | • UMTS requested QoS with Signalling Indication flag |
| *pUMTSMinQoS-SigInd* | • UMTS minimum QoS with Signalling Indication flag |

| | |
|---|---|
| *pPrimaryDNSIPv6addpref* | • Primary DNS IPv6 address preference<br>– 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 |
| *pSecondaryDNSIPv6addpref* | • Secondary DNS IPv6 address preference |
| *paddrAllocationPref* | • DHCP/NAS preference<br>– This enumerated value may be used to indicate the address allocation preference<br>* 0 - NAS signaling is used for address allocation<br>* 1 - DHCP is used for address allocation |
| *pQosClassID* | • 3GPP LTE QoS parameters |
| *pAPNDisabledFlag* | • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled<br>• If set, the profile can not be used for making data calls<br>• Any data call is failed locally<br>• Values:<br>– 0 - FALSE(default)<br>– 1 - True<br>• This parameter is currently read only and can be read by using the function SLQSGetProfileSettings(). |
| *pPDNInactivTimeout* | • Optional 4 Bytes indicating the duration of inactivity timer in seconds<br>• 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<br>• Default value of zero indicates infinite value<br>• This parameter is currently read only and can be read by using the function SLQSGetProfileSettings(). |
| *pAPNClass* | • Optional 1 uint8_t numeric identifier representing the APN in profile<br>• Can be set and queried but is not used by the modem<br>• This parameter is currently read only and can be read by using the function SLQSGetProfileSettings(). |
| *pSupportEmergencyCalls* | • Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled<br>• If set, the profile can be used for making emergency calls<br>• Values:<br>– 0 - FALSE(default)<br>– 1 - TRUE |

### 8.93.2 Field Documentation

8.93.2.1 uint8_t* LibPackprofile_3GPP::pAddrAllocPref

8.93.2.2 uint8_t* LibPackprofile_3GPP::pAPNClass

8.93.2.3 uint8_t* LibPackprofile_3GPP::pAPNDisabledFlag

8.93.2.4 uint8_t* LibPackprofile_3GPP::pAPNName

8.93.2.5 uint16_t* LibPackprofile_3GPP::pAPNnameSize

8.93.2.6 uint8_t* LibPackprofile_3GPP::pAuthenticationPref

8.93.2.7 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSMinimumQoS

8.93.2.8 LibPackGPRSRequestedQoS* LibPackprofile_3GPP::pGPRSRequestedQos

8.93.2.9 uint8_t* LibPackprofile_3GPP::pImCnFlag

8.93.2.10 uint32_t* LibPackprofile_3GPP::pIPv4AddrPref

8.93.2.11 uint16_t* LibPackprofile_3GPP::pIPv6AddPref

8.93.2.12 uint8_t* LibPackprofile_3GPP::pPassword

8.93.2.13 uint16_t* LibPackprofile_3GPP::pPasswordSize

8.93.2.14 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingDhcp

8.93.2.15 uint8_t* LibPackprofile_3GPP::pPcscfAddrUsingPCO

8.93.2.16 uint32_t* LibPackprofile_3GPP::pPDNInactivTimeout

8.93.2.17 uint8_t* LibPackprofile_3GPP::pPdpAccessConFlag

8.93.2.18 uint8_t* LibPackprofile_3GPP::pPdpContext

8.93.2.19 uint8_t* LibPackprofile_3GPP::pPdpDataCompType

8.93.2.20 uint8_t* LibPackprofile_3GPP::pPdpHdrCompType

8.93.2.21 uint8_t* LibPackprofile_3GPP::pPDPtype

8.93.2.22 uint32_t* LibPackprofile_3GPP::pPriDNSIPv4AddPref

8.93.2.23 uint16_t* LibPackprofile_3GPP::pPriDNSIPv6addpref

8.93.2.24 uint8_t* LibPackprofile_3GPP::pPrimaryID

8.93.2.25 uint8_t* LibPackprofile_3GPP::pProfilename

8.93.2.26 uint16_t* LibPackprofile_3GPP::pProfilenameSize

8.93.2.27 LibPackQosClassID* LibPackprofile_3GPP::pQosClassID

8.93.2.28 uint32_t* LibPackprofile_3GPP::pSecDNSIPv4AddPref

#### 8.93.2.29 uint16_t* LibPackprofile_3GPP::pSecDNSIPv6addpref

#### 8.93.2.30 uint8_t* LibPackprofile_3GPP::pSecondaryFlag

#### 8.93.2.31 uint8_t* LibPackprofile_3GPP::pSupportEmergencyCalls

#### 8.93.2.32 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID1Params

#### 8.93.2.33 LibPackTFTIDParams* LibPackprofile_3GPP::pTFTID2Params

#### 8.93.2.34 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSMinQoS

#### 8.93.2.35 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSMinQosSigInd

#### 8.93.2.36 LibPackUMTSQoS* LibPackprofile_3GPP::pUMTSReqQoS

#### 8.93.2.37 LibPackUMTSReqQoSSigInd* LibPackprofile_3GPP::pUMTSReqQoSSigInd

#### 8.93.2.38 uint8_t* LibPackprofile_3GPP::pUsername

#### 8.93.2.39 uint16_t* LibPackprofile_3GPP::pUsernameSize

## 8.94 LibPackprofile_3GPP2 Struct Reference

### Data Fields

- uint8_t * pNegoDnsSrvrPref
- uint32_t * pPppSessCloseTimerDO
- uint32_t * pPppSessCloseTimer1x
- uint8_t * pAllowLinger
- uint16_t * pLcpAckTimeout
- uint16_t * pIpcpAckTimeout
- uint16_t * pAuthTimeout
- uint8_t * pLcpCreqRetryCount
- uint8_t * pIpcpCreqRetryCount
- uint8_t * pAuthRetryCount
- uint8_t * pAuthProtocol
- uint8_t * pUserId
- uint16_t * pUserIdSize
- uint8_t * pAuthPassword
- uint16_t * 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

### 8.94.1 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**

| | |
|---|---|
| *pNegoDnsSrvr-Pref* | • Negotiate DNS Server Preference<br>– 1 - (TRUE)implies request DNS addresses from the PDSN<br>– 0 - (FALSE)implies do not request DNS addresses from the PDSN<br>– Default value is 1 (TRUE) |
| *pPppSessClose-TimerDO* | • PPP Session Close Timer for DO<br>– Timer value (in seconds) on DO indicating how long the PPP Session should linger before closing down |
| *pPppSessClose-Timer1x* | • PPP Session Close Timer for 1X<br>– Timer value (in seconds) on 1X indicating how long the PPP session should linger before closing down |
| *pAllowLinger* | • Allow/disallow lingering of interface<br>– 1 -(TRUE) implies allow lingering<br>– 0 -(FALSE) implies do not allow lingering |
| *pLcpAckTimeout* | • LCP ACK Timeout<br>– Value of LCP ACK Timeout in milliseconds |
| *pIpcpAck-Timeout* | • IPCP ACK Timeout<br>– Value of IPCP ACK Timeout in milliseconds |
| *pAuthTimeout* | • AUTH Timeout<br>– Value of Authentication Timeout in milliseconds |
| *pLcpCreqRetry-Count* | • LCP Configuration Request Retry Count |
| *pIpcpCreqRetry-Count* | • IPCP Configuration Request Retry Count |
| *pAuthRetry-Count* | • Authentication Retry Count value |

| | |
|---|---|
| *pAuthProtocol* | • Authentication Protocol<br>– 1 - PAP<br>– 2 - CHAP<br>– 3 - PAP or CHAP |
| *pUserId* | • User ID to be used during data network authentication<br>• maximum length allowed is 127 bytes;<br>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value. |
| *pUserIdSize;* | • This parameter is an input parameter and should be initialised to the size of pUserId field. Size of this parameter is 2 bytes. |
| *pAuthPassword* | • Password to be used during data network authentication;<br>• maximum length allowed is 127 bytes<br>• QMI_ERR_ARG_TOO_LONG will be returned if the storage on the wireless device is insufficient in size to hold the value. |
| *pAuthPassword-Size;* | • This parameter is an input parameter and should be initialised to the size of pAuthPassword field. Size of this parameter is 2 bytes. |
| *pDataRate* | • Data Rate Requested<br>– 0 - Low (Low speed Service Options (SO15) only)<br>– 1 - Medium (SO33 + low R-SCH)<br>– 2 - High (SO33 + high R-SCH)<br>– Default is 2 |
| *pAppType* | • Application Type:<br>– 0x00000001 - Default Application Type<br>– 0x00000020 - LBS Application Type<br>– 0x00000040 - Tethered Application Type<br>– This parameter is not used while creating/modifying a profile |
| *pDataMode* | • Data Mode to use:<br>– 0 - CDMA or HDR (Hybrid 1X/1xEV-DO)<br>– 1 - CDMA Only (1X only)<br>– 2 - HDR Only (1xEV-DO only)<br>– Default is 0 |

| | |
|---|---|
| *pAppPriority* | • Application Priority<br>– Numerical 1 uint8_t value defining the application priority; higher value implies higher priority<br>– This parameter is not used while creating/modifying a profile |
| *pApnString* | • String representing the Access Point Name<br>• maximum length allowed is 100 bytes<br>• QMI_ERR_ARG_TOO_LONG will be returned if the APN name is too long. |
| *pApnStringSize;* | • This parameter is an input parameter and should be initialised to the size of pApnString field. Size of this parameter is 2 bytes. |
| *pPdnType* | • Packed Data Network Type Requested:<br>– 0 - IPv4 PDN Type<br>– 1 - IPv6 PDN Type<br>– 2 - IPv4 or IPv6 PDN Type<br>– 3 - Unspecified PDN Type (implying no preference) |
| *pIsPcscf-AddressNedded* | • This boolean value is used to control if PCSCF address is requested from PDSN<br>– 1 -(TRUE) implies request for PCSCF value from the PDSN<br>– 0 -(FALSE) implies do not request for PCSCF value from the PDSN |
| *pPrimaryV4Dns-Address* | • IPv4 Primary DNS address<br>– The Primary IPv4 DNS address that can be statically assigned to the UE |
| *pSecondaryV4-DnsAddress* | • IPv4 Secondary DNS address<br>– The Secondary IPv4 DNS address that can be statically assigned to the UE |
| *pPriV6Dns-Address* | • Primary IPv6 DNS address<br>– The Primary IPv6 DNS address that can be statically assigned to the UE |
| *pSecV6Dns-Address* | • Secondary IPv6 DNS address<br>– The Secondary IPv6 DNS address that can be statically assigned to the UE |
| *pRATType* | • Optional 1 uint8_t Flag indicating RAT Type<br>• Values:<br>– 1 - HRPD<br>– 2 - EHRPD<br>– 3 - HRPD_EHRPD<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). |

| | |
|---|---|
| *pAPNEnabled3-GPP2* | • Optional 1 uint8_t Flag indicating if the APN is disabled/enabled<br>• If disabled, the profile can not be used for making data calls<br>• Values:<br>– 0 - Disabled<br>– 1 - Enabled(default value)<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). |
| *pPDNInactiv-Timeout3GPP2* | • Optional 4 Bytes indicating the duration of inactivity timer in seconds<br>• 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<br>• Default value of zero indicates infinite value<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). |
| *pAPNClass3GP-P2* | • Optional 1 uint8_t numeric identifier representing the APN in profile<br>• Can be set and queried but is not used by the modem<br>• This parameter is currently read only and can be read by using the function SLQSGetProfile-Settings(). |

## 8.94.2 Field Documentation

### 8.94.2.1 uint8_t* LibPackprofile_3GPP2::pAllowLinger

### 8.94.2.2 uint8_t* LibPackprofile_3GPP2::pAPNClass3GPP2

### 8.94.2.3 uint8_t* LibPackprofile_3GPP2::pAPNEnabled3GPP2

### 8.94.2.4 uint8_t* LibPackprofile_3GPP2::pApnString

### 8.94.2.5 uint16_t* LibPackprofile_3GPP2::pApnStringSize

### 8.94.2.6 uint8_t* LibPackprofile_3GPP2::pAppPriority

### 8.94.2.7 uint32_t* LibPackprofile_3GPP2::pAppType

### 8.94.2.8 uint8_t* LibPackprofile_3GPP2::pAuthPassword

### 8.94.2.9 uint16_t* LibPackprofile_3GPP2::pAuthPassword_tSize

### 8.94.2.10 uint8_t* LibPackprofile_3GPP2::pAuthProtocol

### 8.94.2.11 uint8_t* LibPackprofile_3GPP2::pAuthRetryCount

### 8.94.2.12 uint16_t* LibPackprofile_3GPP2::pAuthTimeout

### 8.94.2.13 uint8_t* LibPackprofile_3GPP2::pDataMode

### 8.94.2.14 uint8_t* LibPackprofile_3GPP2::pDataRate

#### 8.94.2.15 uint16_t* LibPackprofile_3GPP2::pIpcpAckTimeout

#### 8.94.2.16 uint8_t* LibPackprofile_3GPP2::pIpcpCreqRetryCount

#### 8.94.2.17 uint8_t* LibPackprofile_3GPP2::pIsPcscfAddressNedded

#### 8.94.2.18 uint16_t* LibPackprofile_3GPP2::pLcpAckTimeout

#### 8.94.2.19 uint8_t* LibPackprofile_3GPP2::pLcpCreqRetryCount

#### 8.94.2.20 uint8_t* LibPackprofile_3GPP2::pNegoDnsSrvrPref

#### 8.94.2.21 uint32_t* LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2

#### 8.94.2.22 uint8_t* LibPackprofile_3GPP2::pPdnType

#### 8.94.2.23 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimer1x

#### 8.94.2.24 uint32_t* LibPackprofile_3GPP2::pPppSessCloseTimerDO

#### 8.94.2.25 uint32_t* LibPackprofile_3GPP2::pPrimaryV4DnsAddress

#### 8.94.2.26 uint16_t* LibPackprofile_3GPP2::pPriV6DnsAddress

#### 8.94.2.27 uint8_t* LibPackprofile_3GPP2::pRATType

#### 8.94.2.28 uint32_t* LibPackprofile_3GPP2::pSecondaryV4DnsAddress

#### 8.94.2.29 uint16_t* LibPackprofile_3GPP2::pSecV6DnsAddress

#### 8.94.2.30 uint8_t* LibPackprofile_3GPP2::pUserId

#### 8.94.2.31 uint16_t* LibPackprofile_3GPP2::pUserIdSize

## 8.95 LibPackQosClassID Struct Reference

### Data Fields

- uint8_t QCI
- uint8_t gDlBitRate
- uint32_t maxDlBitRate
- uint32_t gUlBitRate
- uint32_t maxUlBitRate

### 8.95.1 Detailed Description

This structure contains 3GPP LTE QoS parameters

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

**Parameters**

| | |
|---|---|
| *QCI* | • 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 |
| *gDlBitRate* | • Guaranteed DL bit rate |
| *maxDlBitRate* | • Maximum DL bit rate |
| *gUlBitRate* | • Guaranteed UL bit rate |
| *maxUlBitRate* | • Maximum UL bit rate |

### 8.95.2 Field Documentation

#### 8.95.2.1 uint8_t LibPackQosClassID::gDlBitRate

#### 8.95.2.2 uint32_t LibPackQosClassID::gUlBitRate

#### 8.95.2.3 uint32_t LibPackQosClassID::maxDlBitRate

#### 8.95.2.4 uint32_t LibPackQosClassID::maxUlBitRate

#### 8.95.2.5 uint8_t LibPackQosClassID::QCI

## 8.96 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

### 8.96.1 Detailed Description

This structure contains traffic flow template parameters

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

**Parameters**

| | |
|---|---|
| *filterId* | • Filter identifier |
| *eValid* | • Evaluation precedence index |
| *ipVersion* | • IP version number<br>– 4 - IPv4<br>– 6 - IPv6 |
| *pSourceIP* | • Source IP address<br>– IPv4 - Fill the first 4 uint8_ts<br>– IPv6 - Fill all the 16 uint8_ts |
| *sourceIPMask* | • Mask value for the source address |
| *nextHeader* | • Next header/protocol value |
| *destPortRange-Start* | • Start value of the destination port range |
| *destPortRange-End* | • End value of the destination port range |
| *srcPortRange-Start* | • Start value of the source port range |
| *srcPortRange-End* | • End value of the source port range |
| *IPSECSPI* | • IPSEC security parameter index |
| *tosMask* | • TOS mask (Traffic class for IPv6) |
| *flowLabel* | • Flow label |

## 8.96.2 Field Documentation

### 8.96.2.1 uint16_t LibPackTFTIDParams::destPortRangeEnd

### 8.96.2.2 uint32_t LibPackTFTIDParams::destPortRangeStart

#### 8.96.2.3 uint8_t LibPackTFTIDParams::eValid

#### 8.96.2.4 uint8_t LibPackTFTIDParams::filterId

#### 8.96.2.5 uint32_t LibPackTFTIDParams::flowLabel

#### 8.96.2.6 uint32_t LibPackTFTIDParams::IPSECSPI

#### 8.96.2.7 uint8_t LibPackTFTIDParams::ipVersion

#### 8.96.2.8 uint8_t LibPackTFTIDParams::nextHeader

#### 8.96.2.9 uint16_t* LibPackTFTIDParams::pSourceIP

#### 8.96.2.10 uint8_t LibPackTFTIDParams::sourceIPMask

#### 8.96.2.11 uint16_t LibPackTFTIDParams::srcPortRangeEnd

#### 8.96.2.12 uint16_t LibPackTFTIDParams::srcPortRangeStart

#### 8.96.2.13 uint16_t LibPackTFTIDParams::tosMask

## 8.97 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

### 8.97.1 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**

| | |
|---|---|
| *trafficClass* | • 0x00 - Subscribed<br>• 0x01 - Conversational<br>• 0x02 - Streaming<br>• 0x03 - Interactive<br>• 0x04 - Background<br>• 0xff - Invalid UMTS Quality Of Service Information. |

| | |
|---|---|
| *maxUplinkBitrate* | • Maximum uplink bit rate in bits/sec |
| *maxDownlink-Bitrate* | • Maximum downlink bit rate in bits/sec |
| *grntUplinkBitrate* | • Guaranteed uplink bit rate in bits/sec |
| *grntDownlink-Bitrate* | • Guranteed downlink bit rate in bits/sec |
| *qosDelivery-Order* | - Qos delivery order<br>• 0x00 - Subscribe<br>• 0x01 - delivery order on<br>• 0x02 - delivery order off |
| *maxSDUSize* | • Maximum SDU size |
| *sduErrorRatio* | - SDU error ratio<br>• Target value for fraction of SDUs lost or detected as erroneous.<br>• 0x00 - Subscribe<br>• 0x01 - 1∗10^(-2)<br>• 0x02 - 7∗10^(-3)<br>• 0x03 - 1∗10^(-3)<br>• 0x04 - 1∗10^(-4)<br>• 0x05 - 1∗10^(-5)<br>• 0x06 - 1∗10^(-6)<br>• 0x07 - 1∗10^(-1) |
| *resBerRatio* | - Residual bit error ratio<br>• Target value for undetected bit error ratio in in the delivered SDUs.<br>• 0x00 - Subscribe<br>• 0x01 - 5∗10^(-2)<br>• 0x02 - 1∗10^(-2)<br>• 0x03 - 5∗10^(-3)<br>• 0x04 - 4∗10^(-3)<br>• 0x05 - 1∗10^(-3)<br>• 0x06 - 1∗10^(-4)<br>• 0x07 - 1∗10^(-5)<br>• 0x08 - 1∗10^(-6)<br>• 0x09 - 1∗10^(-8) |

| | |
|---|---|
| *deliveryErrSDU* | - Delivery of erroneous SDUs<br>• Indicates whether SDUs detected as erroneous shall be delivered or not.<br>• 0x00 - Subscribe<br>• 0x01 - 5∗10^(-2)<br>• 0x02 - 1∗10^(-2)<br>• 0x03 - 5∗10^(-3)<br>• 0x04 - 4∗10^(-3)<br>• 0x05 - 1∗10^(-3)<br>• 0x06 - 1∗10^(-4)<br>• 0x07 - 1∗10^(-5)<br>• 0x08 - 1∗10^(-6)<br>• 0x09 - 1∗10^(-8) |
| *transferDelay* | - Transfer delay (ms)<br>• Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds. |
| *trafficPriority* | - Transfer handling priority<br>• Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers. |

**Note**

Check trafficClass before use.

## 8.97.2 Field Documentation

### 8.97.2.1 uint8_t LibPackUMTSQoS::deliveryErrSDU

### 8.97.2.2 uint32_t LibPackUMTSQoS::grntDownlinkBitrate

### 8.97.2.3 uint32_t LibPackUMTSQoS::grntUplinkBitrate

### 8.97.2.4 uint32_t LibPackUMTSQoS::maxDownlinkBitrate

### 8.97.2.5 uint32_t LibPackUMTSQoS::maxSDUSize

### 8.97.2.6 uint32_t LibPackUMTSQoS::maxUplinkBitrate

### 8.97.2.7 uint8_t LibPackUMTSQoS::qosDeliveryOrder

### 8.97.2.8 uint8_t LibPackUMTSQoS::resBerRatio

### 8.97.2.9 uint8_t LibPackUMTSQoS::sduErrorRatio

### 8.97.2.10 uint8_t LibPackUMTSQoS::trafficClass

### 8.97.2.11 uint32_t LibPackUMTSQoS::trafficPriority

### 8.97.2.12 uint32_t LibPackUMTSQoS::transferDelay

## 8.98 LibPackUMTSReqQoSSigInd Struct Reference

### Data Fields

- LibPackUMTSQoS UMTSReqQoS
- uint8_t SigInd

### 8.98.1 Detailed Description

This structure contains UMTS requested QoS with Signaling Indication flag

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

**Parameters**

| | |
|---|---|
| *UMTSReqQoS* | • Contains the UMTS Quality Of Service Information<br>• See LibPackUMTSQoS |
| *SigInd* | - Signaling Indication flag<br>• TRUE - Signaling indication ON<br>• FALSE - Signaling indication OFF |

### 8.98.2 Field Documentation

#### 8.98.2.1 uint8_t LibPackUMTSReqQoSSigInd::SigInd

#### 8.98.2.2 LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS

## 8.99 loc_accelAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.99.1 Detailed Description

This structure contains Accelerometer Accept Ready Info

**Parameters**

| | |
|---|---|
| *injectEnable* | • GNSS location engine is ready to accept data from sensor.<br>• Values<br>• 0x01 - Ready to accept sensor data<br>• 0x00 - Not ready to accept sensor data |

| | |
|---|---|
| *samplesPer-Batch* | • number of samples per batch the GNSS location engine is to receive.<br>• samplingFrequency = samplesPerBatch ∗ batchesPerSecond<br>• samplesPerBatch must be a nonzero positive value. |
| *batchPerSec* | • LTE NAS version minor<br>• Number of sensor-data batches the GNSS location engine is to receive per second.<br>• BatchesPerSecond must be a nonzero positive value. |

### 8.99.2 Field Documentation

#### 8.99.2.1 uint16_t loc_accelAcceptReady::batchPerSec

#### 8.99.2.2 uint8_t loc_accelAcceptReady::injectEnable

#### 8.99.2.3 uint16_t loc_accelAcceptReady::samplesPerBatch

## 8.100 loc_accelTempAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.100.1 Detailed Description

This structure contains Accelerometer Temperature Accept Ready Info

**Parameters**

| | |
|---|---|
| *injectEnable* | • GNSS location engine is ready to accept data from sensor.<br>• Values<br>• 0x01 - Ready to accept sensor data<br>• 0x00 - Not ready to accept sensor data |
| *samplesPer-Batch* | • number of samples per batch the GNSS location engine is to receive.<br>• samplingFrequency = samplesPerBatch ∗ batchesPerSecond<br>• samplesPerBatch must be a nonzero positive value. |
| *batchPerSec* | • LTE NAS version minor<br>• Number of sensor-data batches the GNSS location engine is to receive per second.<br>• BatchesPerSecond must be a nonzero positive value. |

### 8.100.2 Field Documentation

#### 8.100.2.1 uint16_t loc_accelTempAcceptReady::batchPerSec

#### 8.100.2.2 uint8_t loc_accelTempAcceptReady::injectEnable

#### 8.100.2.3 uint16_t loc_accelTempAcceptReady::samplesPerBatch

## 8.101 loc_BdsSV Struct Reference

### Data Fields

- uint16_t id
- uint8_t mask

### 8.101.1 Detailed Description

This structure contains the BDS SV Info

**Parameters**

| | |
|---|---|
| *id* | • SV ID of the satellite whose data is to be deleted.<br>– Range for BDS: 201 to 237 |
| *mask* | • Indicates if the ephemeris or almanac for a satellite is to be deleted<br>• Valid values:<br>– QMI_LOC_MASK_DELETE_EPHEMERIS (0x01) - Delete ephemeris for the satellite<br>– QMI_LOC_MASK_DELETE_ALMANAC (0x02) - Delete almanac for the satellite |

### 8.101.2 Field Documentation

#### 8.101.2.1 uint16_t loc_BdsSV::id

#### 8.101.2.2 uint8_t loc_BdsSV::mask

## 8.102 loc_BdsSVInfo Struct Reference

### Data Fields

- uint8_t len
- loc_BdsSV * pSV

### 8.102.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *len* | • Number of sets of the following elements:<br>– gnssSvId<br>– deleteSvInfoMask |
| *pSV* | • Pointer to struct loc_BdsSV. See loc_BdsSV for more information |

### 8.102.2 Field Documentation

#### 8.102.2.1 uint8_t loc_BdsSVInfo::len

#### 8.102.2.2 loc_BdsSV* loc_BdsSVInfo::pSV

## 8.103 loc_CellDb Struct Reference

### Data Fields

- uint32_t mask

### 8.103.1 Detailed Description

This structure contains the cell database

**Parameters**

| | |
|---|---|
| *mask* | • Mask for the cell database assistance data that is to be deleted<br>• Valid values:<br>– 0x00000001 - DELETE_CELLDB_POS<br>– 0x00000002 - DELETE_CELLDB_LATEST_GPS_POS<br>– 0x00000004 - DELETE_CELLDB_OTA_POS<br>– 0x00000008 - DELETE_CELLDB_EXT_REF_POS<br>– 0x00000010 - DELETE_CELLDB_TIMETAG<br>– 0x00000020 - DELETE_CELLDB_CELLID<br>– 0x00000040 - DELETE_CELLDB_CACHED_CELLID<br>– 0x00000080 - DELETE_CELLDB_LAST_SRV_CELL<br>– 0x00000100 - DELETE_CELLDB_CUR_SRV_CELL<br>– 0x00000200 - DELETE_CELLDB_NEIGHBOR_INFO |

### 8.103.2 Field Documentation

#### 8.103.2.1 uint32_t loc_CellDb::mask

## 8.104 loc_ClkInfo Struct Reference

### Data Fields

- uint32_t mask

### 8.104.1 Detailed Description

This structure contains the clock Info

**Parameters**

| | |
|---|---|
| *mask* | • Mask for the clock information assistance data that is to be deleted<br>• Valid bitmasks:<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_TIME_EST (0x00000001) - Mask to delete time estimate from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_FREQ_EST (0x00000002) - Mask to delete frequency estimate from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_WEEK_NUMBER (0x00000004) - Mask to delete week number from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_RTC_TIME (0x00000008) - Mask to delete RTC time from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_TIME_TRANSFER (0x00000010) - Mask to delete time transfer from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GPSTIME_EST (0x00000020) - Mask to delete GPS time estimate from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLOTIME_EST (0x00000040) - Mask to delete GLONASS time estimate from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLODAY_NUMBER (0x00000080) - Mask to delete GLONASS day number from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLO4YEAR_NUMBER (0x00000100) - Mask to delete GLONASS four year number from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GLO_RF_GRP_DELAY (0x00000200) - Mask to delete GLONASS RF GRP delay from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_DISABLE_TT (0x00000400) - Mask to delete disable TT from clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GG_LEAPSEC (0x00000800) - Mask to delete a BDS time estimate from the clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GG_GGTB (0x00001000) - Mask to delete a BDS time estimate from the clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BDSTIME_EST (0x00002000) - Mask to delete a BDS time estimate from the clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_GB_GBTB (0x00004000) - Mask to delete Glonass-to-BDS time bias-related information from the clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BG_BGTB (0x00008000) - Mask to delete BDS-to-GLONASS time bias-related information from the clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BDSWEEK_NUMBER (0x00010000) - Mask to delete the BDS week number from the clock information<br>– QMI_LOC_MASK_DELETE_CLOCK_INFO_BDS_RF_GRP_DELAY (0x00020000) - Mask to delete the BDS RF GRP delay from the clock information |

### 8.104.2 Field Documentation

#### 8.104.2.1 uint32_t loc_ClkInfo::mask

## 8.105 loc_GnssData Struct Reference

### Data Fields

- uint64_t mask

### 8.105.1 Detailed Description

This structure contains the GNSS data

**Parameters**

| | |
|---|---|
| *mask* | • Mask for the GNSS data that is to be deleted<br>• Valid values:<br>– QMI_LOC_MASK_DELETE_GPS_SVDIR (0x00000001) - Mask to delete GPS SVDIR<br>– QMI_LOC_MASK_DELETE_GPS_SVSTEER (0x00000002) - Mask to delete GPS SVSTEER<br>– QMI_LOC_MASK_DELETE_GPS_TIME (0x00000004) - Mask to delete GPS time<br>– QMI_LOC_MASK_DELETE_GPS_ALM_CORR (0x00000008) - Mask to delete almanac correlation<br>– QMI_LOC_MASK_DELETE_GLO_SVDIR (0x00000010) - Mask to delete GLONASS SVDIR<br>– QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete GLONASS SVSTEER<br>– QMI_LOC_MASK_DELETE_GLO_TIME (0x00000040) - Mask to delete GLONASS time<br>– QMI_LOC_MASK_DELETE_GLO_ALM_CORR (0x00000080) - Mask to delete GLONASS almanac correlation<br>– QMI_LOC_MASK_DELETE_SBAS_SVDIR (0x00000100) - Mask to delete SBAS SVDIR<br>– QMI_LOC_MASK_DELETE_SBAS_SVSTEER (0x00000200) - Mask to delete SBAS SVSTEER<br>– QMI_LOC_MASK_DELETE_POSITION (0x00000400) - Mask to delete position estimate<br>– QMI_LOC_MASK_DELETE_TIME (0x00000800) - Mask to delete time estimate<br>– QMI_LOC_MASK_DELETE_IONO (0x00001000) - Mask to delete IONO<br>– QMI_LOC_MASK_DELETE_UTC (0x00002000) - Mask to delete UTC estimate<br>– QMI_LOC_MASK_DELETE_HEALTH (0x00004000) - Mask to delete SV health record<br>– QMI_LOC_MASK_DELETE_SADATA (0x00008000) - Mask to delete SADATA<br>– QMI_LOC_MASK_DELETE_RTI (0x00010000) - Mask to delete RTI<br>– QMI_LOC_MASK_DELETE_SV_NO_EXIST (0x00020000) - Mask to delete SV_NO_EXIST<br>– QMI_LOC_MASK_DELETE_FREQ_BIAS_EST (0x00040000) - Mask to delete frequency bias estimate<br>– QMI_LOC_MASK_DELETE_BDS_SVDIR (0x00080000) - Mask to delete BDS SVDIR<br>– QMI_LOC_MASK_DELETE_BDS_SVSTEER (0x00100000) - Mask to delete BDS SVSTEER<br>– QMI_LOC_MASK_DELETE_BDS_TIME (0x00200000) - Mask to delete BDS time<br>– QMI_LOC_MASK_DELETE_BDS_ALM_CORR (0x00400000) - Mask to delete BDS almanac correlation<br>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GPS (0x00800000) - Mask to delete GNSS SV blacklist GPS<br>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GLO (0x01000000) - Mask to delete GNSS SV blacklist GLO<br>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_BDS (0x02000000) - Mask to delete GNSS SV blacklist BDS |

### 8.105.2 Field Documentation

#### 8.105.2.1 uint64_t loc_GnssData::mask

## 8.106 loc_gpsTime Struct Reference

### Data Fields

- uint16_t gpsWeek
- uint32_t gpsTimeOfWeekMs

### 8.106.1 Detailed Description

This structure contains GPS Time info.

**Parameters**

| | |
|---|---|
| *gpsWeek* | • Current GPS week as calculated from midnight, Jan. 6, 1980.<br>• Units - Weeks |
| *gpsTimeOf-WeekMs* | • Amount of time into the current GPS week.<br>• Units - Milliseconds |

### 8.106.2 Field Documentation

#### 8.106.2.1 uint32_t loc_gpsTime::gpsTimeOfWeekMs

#### 8.106.2.2 uint16_t loc_gpsTime::gpsWeek

## 8.107 loc_gyroAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.107.1 Detailed Description

This structure contains Gyroscope Accept Ready Info

**Parameters**

| | |
|---|---|
| *injectEnable* | • GNSS location engine is ready to accept data from sensor.<br>• Values<br>• 0x01 - Ready to accept sensor data<br>• 0x00 - Not ready to accept sensor data |
| *samplesPer-Batch* | • number of samples per batch the GNSS location engine is to receive.<br>• samplingFrequency = samplesPerBatch $*$ batchesPerSecond<br>• samplesPerBatch must be a nonzero positive value. |

| | |
|---|---|
| *batchPerSec* | • LTE NAS version minor<br>• Number of sensor-data batches the GNSS location engine is to receive per second.<br>• BatchesPerSecond must be a nonzero positive value. |

### 8.107.2 Field Documentation

#### 8.107.2.1 uint16_t loc_gyroAcceptReady::batchPerSec

#### 8.107.2.2 uint8_t loc_gyroAcceptReady::injectEnable

#### 8.107.2.3 uint16_t loc_gyroAcceptReady::samplesPerBatch

## 8.108 loc_gyroTempAcceptReady Struct Reference

### Data Fields

- uint8_t injectEnable
- uint16_t samplesPerBatch
- uint16_t batchPerSec

### 8.108.1 Detailed Description

This structure contains Gyroscope Temperature Accept Ready Info

**Parameters**

| | |
|---|---|
| *injectEnable* | • GNSS location engine is ready to accept data from sensor.<br>• Values<br>• 0x01 - Ready to accept sensor data<br>• 0x00 - Not ready to accept sensor data |
| *samplesPer-Batch* | • number of samples per batch the GNSS location engine is to receive.<br>• samplingFrequency = samplesPerBatch ∗ batchesPerSecond<br>• samplesPerBatch must be a nonzero positive value. |
| *batchPerSec* | • LTE NAS version minor<br>• Number of sensor-data batches the GNSS location engine is to receive per second.<br>• BatchesPerSecond must be a nonzero positive value. |

### 8.108.2 Field Documentation

#### 8.108.2.1 uint16_t loc_gyroTempAcceptReady::batchPerSec

#### 8.108.2.2 uint8_t loc_gyroTempAcceptReady::injectEnable

#### 8.108.2.3 uint16_t loc_gyroTempAcceptReady::samplesPerBatch

## 8.109 loc_IPv4Config Struct Reference

### Data Fields

- uint32_t IPv4Addr
- uint16_t IPv4Port

### 8.109.1 Detailed Description

This structure specifies information regarding the IPv4 address and port.

**Parameters**

| | |
|---|---|
| *IPv4Addr* | • IPv4 address. |
| *IPv4Port* | • IPv4 port. |

### 8.109.2 Field Documentation

#### 8.109.2.1 uint32_t loc_IPv4Config::IPv4Addr

#### 8.109.2.2 uint16_t loc_IPv4Config::IPv4Port

## 8.110 loc_IPv4Info Struct Reference

### Data Fields

- uint32_t address
- uint16_t port
- uint8_t TlvPresent

### 8.110.1 Detailed Description

This structure contains LOC Server IPV4 info field.

**Parameters**

| | |
|---|---|
| *address* | • IPv4 address. |
| *port* | • IPv4 port. |
| *TlvPresent* | • Tlv Present or not |

### 8.110.2 Field Documentation

#### 8.110.2.1 uint32_t loc_IPv4Info::address

#### 8.110.2.2 uint16_t loc_IPv4Info::port

#### 8.110.2.3 uint8_t loc_IPv4Info::TlvPresent

## 8.111 loc_IPv6Config Struct Reference

### Data Fields

- uint16_t IPv6Addr [8]
- uint32_t IPv6Port

### 8.111.1 Detailed Description

This structure specifies information regarding the IPv6 address and port.

**Parameters**

| | |
|---|---|
| *IPv6Addr* | • IPv6 address.<br>• Type - Array of unsigned integers<br>• Maximum length of the array: 8 |
| *IPv6Port* | • IPv6 port. |

### 8.111.2 Field Documentation

#### 8.111.2.1 uint16_t loc_IPv6Config::IPv6Addr[8]

#### 8.111.2.2 uint32_t loc_IPv6Config::IPv6Port

## 8.112 loc_IPv6Info Struct Reference

### Data Fields

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

### 8.112.1 Detailed Description

This structure contains LOC Get Server IPV6 info field.

**Parameters**

| | |
|---|---|
| *address* | • IPv6 address.<br>• Type - Array of unsigned integers<br>• Maximum length of the array - 8 |
| *port* | • IPv6 port. |
| *TlvPresent* | • Tlv Present or not |

### 8.112.2 Field Documentation

#### 8.112.2.1 uint16_t loc_IPv6Info::address[8]

#### 8.112.2.2 uint32_t loc_IPv6Info::port

#### 8.112.2.3 uint8_t loc_IPv6Info::TlvPresent

## 8.113 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

### 8.113.1 Detailed Description

This structure contains the Application Information

**Parameters**

| | |
|---|---|
| *appProvider-Length* | • Length of the Application Provider |
| *pAppProvider* | • Application Provider<br>• Depends upon the Length of application Provider |
| *appNameLength* | • Length of Application Name |
| *pAppName* | • Application Name<br>• Depends upon the Length of application Name |

| | |
|---|---|
| *appVersionValid* | • Specifies whether the application version string contains a valid value<br>• 0x00 (FALSE) Application version string is invalid<br>• 0x01 (TRUE) Application version string is valid |
| *appVersion-Length* | • Length of Application Version |
| *pAppVersion* | • Application Version<br>• Depends upon the Length of application Version |

### 8.113.2 Field Documentation

#### 8.113.2.1 uint8_t loc_LocApplicationInfo::appNameLength

#### 8.113.2.2 uint8_t loc_LocApplicationInfo::appProviderLength

#### 8.113.2.3 uint8_t loc_LocApplicationInfo::appVersionLength

#### 8.113.2.4 uint8_t loc_LocApplicationInfo::appVersionValid

#### 8.113.2.5 uint8_t* loc_LocApplicationInfo::pAppName

#### 8.113.2.6 uint8_t* loc_LocApplicationInfo::pAppProvider

#### 8.113.2.7 uint8_t* loc_LocApplicationInfo::pAppVersion

## 8.114 loc_precisionDilution Struct Reference

### Data Fields

- uint32_t PDOP
- uint32_t HDOP
- uint32_t VDOP

### 8.114.1 Detailed Description

This structure contains Dilution of precision associated with this position.

**Parameters**

| | |
|---|---|
| *PDOP* | • Position dilution of precision.<br>• Range - 1 (highest accuracy) to 50 (lowest accuracy)<br>• PDOP = square root of (Square of HDOP + Square of VDOP2 ) |
| *HDOP* | • Horizontal dilution of precision.<br>• Range - 1 (highest accuracy) to 50 (lowest accuracy) |

| | |
|---|---|
| *VDOP* | • Vertical dilution of precision.<br>• Range- 1 (highest accuracy) to 50 (lowest accuracy) |

### 8.114.2 Field Documentation

#### 8.114.2.1 uint32_t loc_precisionDilution::HDOP

#### 8.114.2.2 uint32_t loc_precisionDilution::PDOP

#### 8.114.2.3 uint32_t loc_precisionDilution::VDOP

## 8.115 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

### 8.115.1 Detailed Description

Contain fields in struct loc_satelliteInfo

**Parameters**

| | |
|---|---|
| *svListLen* | • number of sets of the following elements:<br>– validMask<br>– system<br>– gnssSvid<br>– healthStatus<br>– svStatus<br>– svInfoMask<br>– elevation<br>– azimuth<br>– snr |

| | |
|---|---|
| *validMask* | • Bitmask indicating which of the fields in this TLV are valid. Valid bitmasks:<br>– 0x00000001 - VALID_SYSTEM<br>– 0x00000002 - VALID_GNSS_SVID<br>– 0x00000004 - VALID_HEALTH_STATUS<br>– 0x00000008 - VALID_PROCESS_STATUS<br>– 0x00000010 - VALID_SVINFO_MASK<br>– 0x00000020 - VALID_ELEVATION<br>– 0x00000040 - VALID_AZIMUTH<br>– 0x00000080 - VALID_SNR |
| *system* | • Indicates to which constellation this SV belongs. Valid values:<br>– eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite<br>– eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite<br>– eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite<br>– eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite<br>– eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite<br>– eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite |
| *gnssSvId* | • GNSS SV ID. The GPS and GLONASS SVs can be disambiguated using the system field. Range:<br>– FOR GPS: 1 to 32<br>– FOR GLONASS: 1 to 32<br>– FOR SBAS: 120 to 151<br>– for BDS: 201 to 237 |
| *healthStatus* | • health status. Range: 0 - 1<br>– 0 - unhealthy<br>– 1 - healthy |
| *svStatus* | • SV process status. Valid values:<br>– eQMI_LOC_SV_STATUS_IDLE (1) - SV is not being actively processed<br>– eQMI_LOC_SV_STATUS_SEARCH (2) - The system is searching for this SV<br>– eQMI_LOC_SV_STATUS_TRACK (3) - SV is being tracked |
| *svInfoMask* | • Indicates whether almanac and ephemeris information is available. Valid bitmasks:<br>– 0x01 - SVINFO_HAS_EPHEMERIS<br>– 0x02 - SVINFO_HAS_ALMANAC |
| *elevation* | • SV elevation angle.<br>– Units: Degrees<br>– Range: 0 to 90 |

| | |
|---|---|
| *azimuth* | • SV azimuth angle.<br>– Units: Degrees<br>– Range: 0 to 360 |
| *snr* | • SV signal-to-noise ratio<br>– Units: dB-Hz |

### 8.115.2 Field Documentation

#### 8.115.2.1 float loc_satelliteInfo::azimuth

#### 8.115.2.2 float loc_satelliteInfo::elevation

#### 8.115.2.3 uint16_t loc_satelliteInfo::gnssSvId

#### 8.115.2.4 uint8_t loc_satelliteInfo::healthStatus

#### 8.115.2.5 float loc_satelliteInfo::snr

#### 8.115.2.6 uint8_t loc_satelliteInfo::svInfoMask

#### 8.115.2.7 uint8_t loc_satelliteInfo::svListLen

#### 8.115.2.8 uint32_t loc_satelliteInfo::svStatus

#### 8.115.2.9 uint32_t loc_satelliteInfo::system

#### 8.115.2.10 uint32_t loc_satelliteInfo::validMask

## 8.116 loc_sensorDataUsage Struct Reference

### Data Fields

- uint32_t usageMask
- uint32_t aidingIndicatorMask

### 8.116.1 Detailed Description

This structure contains Sensor Data Usage info.

**Parameters**

| | |
|---|---|
| *usageMask* | • Specifies which sensors were used in calculating the position in the position report.<br>– Value<br>* 0x00000001 - Accelerometer used<br>* 0x00000002 - Gyroscope used |

| | |
|---|---|
| *aidingIndicator-Mask* | • Specifies which results were aided by sensors.<br>– Value<br>* 0x00000001 - AIDED_HEADING<br>* 0x00000002 - AIDED_SPEED<br>* 0x00000004 - AIDED_POSITION<br>* 0x00000008 - AIDED_VELOCITY |

### 8.116.2 Field Documentation

#### 8.116.2.1 uint32_t loc_sensorDataUsage::aidingIndicatorMask

#### 8.116.2.2 uint32_t loc_sensorDataUsage::usageMask

## 8.117 loc_SV Struct Reference

### Data Fields

- uint16_t id
- uint32_t system
- uint8_t mask

### 8.117.1 Detailed Description

This structure contains the Delete LOC SV Info

**Parameters**

| | |
|---|---|
| *id* | • LOC SV ID of the satellite whose data is to be deleted<br>• Range:<br>– For GPS: 1 to 32<br>– For SBAS: 33 to 64<br>– For GLONASS: 65 to 96 |
| *system* | • Indicates to which constellation this loc_SV belongs<br>• Valid values:<br>– eQMI_LOC_SV_SYSTEM_GPS (1) - GPS satellite<br>– eQMI_LOC_SV_SYSTEM_GALILEO (2) - GALILEO satellite<br>– eQMI_LOC_SV_SYSTEM_SBAS (3) - SBAS satellite<br>– eQMI_LOC_SV_SYSTEM_COMPASS (4) - COMPASS satellite<br>– eQMI_LOC_SV_SYSTEM_GLONASS (5) - GLONASS satellite<br>– eQMI_LOC_SV_SYSTEM_BDS (6) - BDS satellite |
| *mask* | • Indicates if the ephemeris or almanac for a satellite is to be deleted<br>• Valid values:<br>– 0x01 - DELETE_EPHEMERIS<br>– 0x02 - DELETE_ALMANAC |

### 8.117.2 Field Documentation

#### 8.117.2.1 uint16_t loc_SV::id

#### 8.117.2.2 uint8_t loc_SV::mask

#### 8.117.2.3 uint32_t loc_SV::system

## 8.118 loc_SVInfo Struct Reference

### Data Fields

- uint8_t len
- loc_SV * pSV

### 8.118.1 Detailed Description

This structure contains the elements of Delete LOC SV Info

**Parameters**

| | |
|---|---|
| *len* | • Number of sets of the following elements in struct loc_SV:<br>– gnssSvId<br>– system<br>– deleteSvInfoMask |
| *pSV* | • Pointer to struct loc_SV. See loc_SV for more information |

### 8.118.2 Field Documentation

#### 8.118.2.1 uint8_t loc_SVInfo::len

#### 8.118.2.2 loc_SV* loc_SVInfo::pSV

## 8.119 loc_svUsedforFix Struct Reference

### Data Fields

- uint8_t gnssSvUsedList_len
- uint16_t gnssSvUsedList [255]

### 8.119.1 Detailed Description

This structure contains SVs Used to Calculate the Fix.

**Parameters**

| | |
|---|---|
| *gnssSvUsedList-_len* | • Number of sets of gnssSvUsedList |

| | |
|---|---|
| *pGnssSvUsed-List* | • Entry in the list contains the SV ID of a satellite used for calculating this position report.<br>• Following information is associated with each SV ID:<br>– GPS - 1 to 32<br>– SBAS - 33 to 64<br>– GLONASS - 65 to 96<br>– QZSS - 193 to 197<br>– BDS - 201 to 237 |

### 8.119.2 Field Documentation

#### 8.119.2.1 uint16_t loc_svUsedforFix::gnssSvUsedList[255]

#### 8.119.2.2 uint8_t loc_svUsedforFix::gnssSvUsedList_len

## 8.120 loc_urlAddr Struct Reference

### Data Fields

- char address [256]
- uint8_t TlvPresent

### 8.120.1 Detailed Description

This structure contains LOC Get Server URL address field.

**Parameters**

| | |
|---|---|
| *address* | • Uniform Resource Locator<br>• Type - NULL terminated string<br>• Maximum string length (including NULL terminator) 256 |
| *TlvPresent* | • Tlv Present or not |

### 8.120.2 Field Documentation

#### 8.120.2.1 char loc_urlAddr::address[256]

#### 8.120.2.2 uint8_t loc_urlAddr::TlvPresent

## 8.121 loc_URLAddrInfo Struct Reference

### Data Fields

- char urlAddr [256]

### 8.121.1 Detailed Description

This structure specifies information regarding the URL.

**Parameters**

| | |
|---|---|
| *urlAddr* | • URL address.<br>• Type - NULL-terminated string<br>• Maximum string length (including NULL terminator) - 256 |

### 8.121.2 Field Documentation

#### 8.121.2.1 char loc_URLAddrInfo::urlAddr[256]

## 8.122 IteSSInfo Struct Reference

### Data Fields

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

### 8.122.1 Detailed Description

This structure contains the parameters for LTE Signal Strength Information

**Parameters**

| | |
|---|---|
| *rssi* | • RSSI in dBm (signed value).<br>• A value of -125 dBm or lower is used to indicate No Signal.<br>  – For CDMA and UMTS, this indicates forward link pilot Ec<br>  – For GSM, this indicates received signal strength |
| *rsrq* | • RSRQ value in dB (signed integer value) as measured by L1.<br>• Range: -3 to -20 (-3 means -3 dB, -20 means -20 dB). |
| *rsrp* | • Current RSRP in dBm as measured by L1.<br>• Range: -44 to -140 (-44 means -44 dBm, -140 means -140 dBm). |
| *snr* | • SNR level as a scaled integer in units of 0.1 dB. e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246, |

### 8.122.2 Field Documentation

#### 8.122.2.1 int16_t lteSSInfo::rsrp

#### 8.122.2.2 int8_t lteSSInfo::rsrq

#### 8.122.2.3 int8_t lteSSInfo::rssi

#### 8.122.2.4 int16_t lteSSInfo::snr

## 8.123 messageModeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSMessageModeInfo MessageModeInfo

### 8.123.1 Detailed Description

This structure contains message mode TLV information.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *MessageMode-Info* | • Message Mode<br>• See sMSMessageModeInfo for more information |

### 8.123.2 Field Documentation

#### 8.123.2.1 sMSMessageModeInfo messageModeTlv::MessageModeInfo

#### 8.123.2.2 uint8_t messageModeTlv::TlvPresent

## 8.124 nas_acqOrderPref Struct Reference

### Data Fields

- uint8_t acqOrdeLen
- uint8_t * pAcqOrder

### 8.124.1 Detailed Description

Contain the Acquisition Order Preference.

**Parameters**

| | |
|---|---|
| *acqOrdeLen* | • Number of sets of the following elements. |

| | |
|---|---|
| *pAcqOrder* | • Acquisition order preference to be set. Values:<br>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X<br>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)<br>– 0x04 - NAS_RADIO_IF_GSM - GSM<br>– 0x05 - NAS_RADIO_IF_UMTS - UMTS<br>– 0x08 - NAS_RADIO_IF_LTE - LTE<br>– 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA |

### 8.124.2 Field Documentation

#### 8.124.2.1 uint8_t nas_acqOrderPref::acqOrdeLen

#### 8.124.2.2 uint8_t∗ nas_acqOrderPref::pAcqOrder

## 8.125 nas_AcqOrderPrefTlv Struct Reference

### Data Fields

- uint8_t acqOrdeLen
- uint8_t ∗ pAcqOrder
- uint8_t TlvPresent

### 8.125.1 Detailed Description

Contain the Acquisition Order Preference.

**Parameters**

| | |
|---|---|
| *acqOrdeLen* | • Number of sets of the following elements. |
| *pAcqOrder* | • Acquisition order preference to be set. Values:<br>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X<br>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)<br>– 0x04 - NAS_RADIO_IF_GSM - GSM<br>– 0x05 - NAS_RADIO_IF_UMTS - UMTS<br>– 0x08 - NAS_RADIO_IF_LTE - LTE<br>– 0x09 - NAS_RADIO_IF_TDSCDMA - TD-SCDMA<br>– 0x0C - NAS_RADIO_IF_NR5G - NR5G |
| *TlvPresent* | • Tlv Present |

### 8.125.2 Field Documentation

#### 8.125.2.1 uint8_t nas_AcqOrderPrefTlv::acqOrdeLen

#### 8.125.2.2 uint8_t* nas_AcqOrderPrefTlv::pAcqOrder

#### 8.125.2.3 uint8_t nas_AcqOrderPrefTlv::TlvPresent

## 8.126 nas_ActPilotPNElement Struct Reference

### Data Fields

- uint16_t ActSetPilotPN
- uint8_t ActSetPilotPNStrength

### 8.126.1 Detailed Description

This structure contains pilot PN Element.

**Parameters**

| | |
|---|---|
| *ActSetPilotPN* | • Active Pilot PN |
| *ActSetPilotPN-Strength* | • Active Pilot PN strength |

### 8.126.2 Field Documentation

#### 8.126.2.1 uint16_t nas_ActPilotPNElement::ActSetPilotPN

#### 8.126.2.2 uint8_t nas_ActPilotPNElement::ActSetPilotPNStrength

## 8.127 nas_AddCDMASysInfo Struct Reference

### Data Fields

- uint16_t geoSysIdx
- uint16_t regPrd

### 8.127.1 Detailed Description

Structure for storing the Additional CDMA System Information.

**Parameters**

| | |
|---|---|
| *geoSysIdx* | • System table index referencing the beginning of the geo in which the current serving system is present.<br>• When the system index is not known, 0xFFFF is used. |
| *regPrd* | • Registration period after the CDMA system is acquired.<br>• When the CDMA registration period is not valid, 0xFFFF is used. |

### 8.127.2 Field Documentation

#### 8.127.2.1 uint16_t nas_AddCDMASysInfo::geoSysIdx

#### 8.127.2.2 uint16_t nas_AddCDMASysInfo::regPrd

## 8.128 nas_AddSysInfo Struct Reference

### Data Fields

- uint16_t geoSysIdx
- uint32_t cellBroadcastCap

### 8.128.1 Detailed Description

Structure for storing the Additional GSM and WCDMA System Information.

**Parameters**

| | |
|---|---|
| *geoSysIdx* | • System table index referencing the beginning of the geo in which the current serving system is present.<br>• When the system index is not known, 0xFFFF is used. |
| *cellBroadcast-Cap* | • Cell broadcast capability of the serving system.<br>• When the CDMA registration period is not valid, 0xFFFF is used.<br>– NAS_CELL_BROADCAST_CAP_UNKNOWN - Cell broadcast support is unknown<br>– NAS_CELL_BROADCAST_CAP_OFF - Cell broadcast is not supported<br>– NAS_CELL_BROADCAST_CAP_ON - Cell broadcast is supported |

### 8.128.2 Field Documentation

#### 8.128.2.1 uint32_t nas_AddSysInfo::cellBroadcastCap

#### 8.128.2.2 uint16_t nas_AddSysInfo::geoSysIdx

## 8.129 nas_BandPrefTlv Struct Reference

### Data Fields

- uint64_t BandPref
- uint8_t TlvPresent

### 8.129.1 Detailed Description

Contain the band preference for system selection preferences.

**Parameters**

| | |
|---|---|
| *BandPref* | • Bitmask representing the radio technologies that are disabled.<br>• Bit values:<br>• Bit mask representing the band preference<br>• Bit values:<br>– Bit 0 - Band Class 0, A-System<br>– Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band<br>– Bit 2 - Band Class 1, all blocks<br>– Bit 3 - Band Class 2 place holder<br>– Bit 4 - Band Class 3, A-System<br>– Bit 5 - Band Class 4, all blocks<br>– Bit 6 - Band Class 5, all blocks<br>– Bit 7 - GSM_DCS_1800 band<br>– Bit 8 - GSM Extended GSM (E-GSM) 900 band<br>– Bit 9 - GSM Primary GSM (P-GSM) 900 band<br>– Bit 10 - Band Class 6<br>– Bit 11 - Band Class 7<br>– Bit 12 - Band Class 8<br>– Bit 13 - Band Class 9<br>– Bit 14 - Band Class 10<br>– Bit 15 - Band Class 11<br>– Bit 16 - GSM 450 band<br>– Bit 17 - GSM 480 band<br>– Bit 18 - GSM 750 band<br>– Bit 19 - GSM 850 band<br>– Bit 20 - GSM Railways GSM 900 Band<br>– Bit 21 - GSM PCS 1900 band<br>– Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band<br>– Bit 23 - WCDMA U.S. PCS 1900 band<br>– Bit 24 - WCDMA Europe and China DCS 1800 band<br>– Bit 25 - WCDMA U.S. 1700 band<br>– Bit 26 - WCDMA U.S. 850 band<br>– Bit 27 - WCDMA Japan 800 band<br>– Bit 28 - Band Class 12<br>– Bit 29 - Band Class 14<br>– Bit 30 - Reserved<br>– Bit 31 - Band Class 15<br>– Bit 32 to 47 - Reserved<br>– Bit 48 - WCDMA Europe 2600 band<br>– Bit 49 - WCDMA Europe and Japan 900 band<br>– Bit 50 - WCDMA Japan 1700 band<br>– Bit 51 to 55 - Reserved<br>– Bit 56 - Band Class 16<br>– Bit 57 - Band Class 17<br>– Bit 58 - Band Class 18<br>– Bit 59 - Band Class 19<br>– Bit 60 to 64 - Reserved |

| | |
|---|---|
| *TlvPresent* | • Tlv Present |

### 8.129.2 Field Documentation

#### 8.129.2.1 uint64_t nas_BandPrefTlv::BandPref

#### 8.129.2.2 uint8_t nas_BandPrefTlv::TlvPresent

## 8.130 nas_CallBarringSysInfo Struct Reference

### Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

### 8.130.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *csBarStatus* | • Call barring status for circuit-switched calls.<br>– NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only<br>– NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only<br>– NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type<br>– NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types<br>– NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown |
| *psBarStatus* | • Call barring status for packet-switched calls.<br>– NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only<br>– NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only<br>– NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type<br>– NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types<br>– NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown |

### 8.130.2 Field Documentation

#### 8.130.2.1 uint32_t nas_CallBarringSysInfo::csBarStatus

#### 8.130.2.2 uint32_t nas_CallBarringSysInfo::psBarStatus

## 8.131 nas_callBarStatus Struct Reference

### Data Fields

- uint32_t csBarStatus
- uint32_t psBarStatus

### 8.131.1 Detailed Description

This structure contains Call Barring Status.

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

**Parameters**

| | |
|---|---|
| *csBarStatus* | • Call Barring Status for circuit-switched calls.<br>• Values:<br>• NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only<br>• NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only<br>• NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type<br>• NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types<br>• NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown |
| *psBarStatus* | • Call Barring Status for packet-switched calls.<br>• Values:<br>– NAS_CELL_ACCESS_NORMAL_ONLY - Cell access is allowed for normal calls only<br>– NAS_CELL_ACCESS_EMERGENCY_ONLY - Cell access is allowed for emergency calls only<br>– NAS_CELL_ACCESS_NO_CALLS - Cell access is not allowed for any call type<br>– NAS_CELL_ACCESS_ALL_CALLS - Cell access is allowed for all call types<br>– NAS_CELL_ACCESS_UNKNOWN - Cell access type is unknown |

### 8.131.2 Field Documentation

#### 8.131.2.1 uint32_t nas_callBarStatus::csBarStatus

#### 8.131.2.2 uint32_t nas_callBarStatus::psBarStatus

## 8.132 nas_CDMAChannel Struct Reference

### Data Fields

- uint16_t priChA
- uint16_t priChB
- uint16_t secChA
- uint16_t secChB

### 8.132.1 Detailed Description

This structure contains CDMA channel.

**Parameters**

| | |
|---|---|
| *priChA* | • A Channel number for the primary carrier.<br>– 0xFFFF - Not Available |
| *priChB* | • B Channel number for the primary carrier.<br>– 0xFFFF - Not Available |
| *secChA* | • A Channel number for the secondary carrier.<br>– 0xFFFF - Not Available |
| *secChB* | • B Channel number for the secondary carrier.<br>– 0xFFFF - Not Available |

### 8.132.2 Field Documentation

#### 8.132.2.1 uint16_t nas_CDMAChannel::priChA

#### 8.132.2.2 uint16_t nas_CDMAChannel::priChB

#### 8.132.2.3 uint16_t nas_CDMAChannel::secChA

#### 8.132.2.4 uint16_t nas_CDMAChannel::secChB

## 8.133 nas_CDMAECIOThresh Struct Reference

### Data Fields

- uint8_t CDMAECIOThreshListLen
- int16_t * pCDMAECIOThreshList

### 8.133.1 Detailed Description

This structure contains CDMA ECIO threshold related parameters.

**Parameters**

| | |
|---|---|
| *CDMAECIO-ThreshListLen* | • Length of the CDMA ECIO threshold list parameter to follow |
| *pCDMAECIO-ThreshList* | • Array of ECIO thresholds (in units of 0.1 dB)<br>• Maximum of 32 values. Range for ECIO values: -31.5 to 0 (in dB) |

### 8.133.2 Field Documentation

#### 8.133.2.1 uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen

#### 8.133.2.2 int16_t* nas_CDMAECIOThresh::pCDMAECIOThreshList

## 8.134 nas_CDMAInfo Struct Reference

### Data Fields

- uint16_t sid
- uint16_t nid
- uint16_t baseId
- uint16_t refpn
- uint32_t baseLat
- uint32_t baseLong

### 8.134.1 Detailed Description

This structure contains information about the CDMA Network.

**Parameters**

| | |
|---|---|
| *sid* | • System ID.<br>– 0xFFFF - Not Available |
| *nid* | • Network ID.<br>– 0xFFFF - Not Available |
| *baseId* | • Base station ID.<br>– 0xFFFF - Not Available |
| *refpn* | • Reference PN.<br>– 0xFFFF - Not Available |
| *baseLat* | • Latitude of the current base station in units of 0.25 sec.<br>– 0xFFFFFFFF - Not Available |
| *baseLong* | • Longitude of the current base station in units of 0.25 sec.<br>– 0xFFFFFFFF - Not Available |

### 8.134.2 Field Documentation

#### 8.134.2.1 uint16_t nas_CDMAInfo::baseId

#### 8.134.2.2 uint32_t nas_CDMAInfo::baseLat

#### 8.134.2.3 uint32_t nas_CDMAInfo::baseLong

#### 8.134.2.4 uint16_t nas_CDMAInfo::nid

#### 8.134.2.5 uint16_t nas_CDMAInfo::refpn

#### 8.134.2.6 uint16_t nas_CDMAInfo::sid

## 8.135 nas_CDMARSSIThresh Struct Reference

### Data Fields

- uint8_t CDMARSSIThreshListLen
- int16_t * pCDMARSSIThreshList

### 8.135.1 Detailed Description

This structure contains CDMA RSSI threshold related parameters.

**Parameters**

| | |
|---|---|
| *CDMARSSI-ThreshListLen* | • Length of the CDMARSSI threshold list parameter to follow |
| *pCDMARSSI-ThreshList* | • Array of RSSI thresholds (in units of 0.1 dBm)<br>• maximum of 32 values.<br>• Range for RSSI values:-105 to -21 (in dBm). |

### 8.135.2 Field Documentation

#### 8.135.2.1 uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen

#### 8.135.2.2 int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList

## 8.136 nas_CDMASysInfo Struct Reference

### Data Fields

- nas_sysInfoCommon sysInfoCDMA
- uint8_t isSysPrlMatchValid
- uint8_t isSysPrlMatch
- 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]

### 8.136.1 Detailed Description

Structure for storing the CDMA System Information.

**Parameters**

| | |
|---|---|
| *sysInfoCDMA* | • See nas_sysInfoCommon for more information. |
| *isSysPrlMatch-Valid* | • Indicates whether the system PRL match is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *isSysPrlMatch* | • Indicates whether the system is in a PRL.<br>• Only applies to CDMA/HDR.<br>– 0x00 - System is not in a PRL<br>– 0x01 - System is in a PRL<br>– 0xFF - Not Available<br>• If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL.<br>• If the system is in a PRL, roam_status is set to the value based on the standard specification. |
| *pRevInUseValid* | • Indicates whether the P_Rev in use is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *pRevInUse* | • The lesser of the base station P_Rev and mobile P_Rev<br>• Only applicable for CDMA.<br>– 0xFF - Not Available |
| *bsPRevValid* | • Indicates whether the base station P_Rev is valid<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *bsPRev* | • Base station P_Rev.<br>• Only applicable for CDMA.<br>– 0xFF - Not Available |

| | |
|---|---|
| *ccsSupported-Valid* | • Indicates whether the supported concurrent service is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *ccsSupported* | • Whether concurrent service is supported.<br>• Only applicable for CDMA.<br>– 0x00 - Not supported<br>– 0x01 - Supported<br>– 0xFF - Not Available |
| *cdmaSysIdValid* | • Indicates whether the CDMA system ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *systemID* | • System ID.<br>– 0xFFFF - Not Available |
| *networkID* | • Network ID.<br>– 0xFFFF - Not Available |
| *bsInfoValid* | • Indicates whether the base station information is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *baseLat* | • Base station latitude in units of 0.25 sec.<br>• Expressed as a two's complement signed number with positive numbers signifying North latitudes.<br>– 0xFFFFFFFF - Not Available |
| *baseLong* | • Base station longitude in units of 0.25 sec.<br>• Expressed as a two's complement signed number with positive numbers signifying East latitudes.<br>– 0xFFFFFFFF - Not Available |
| *packetZoneValid* | • Indicates whether the packet zone is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |

| | |
|---|---|
| *packetZone* | • Packet zone (8-bit).<br>  – 0xFFFF indicates no packet zone.<br>• Only applicable for CDMA. |
| *networkIdValid* | • Indicates whether the network ID is valid.<br>  – 0x00 - Invalid<br>  – 0x01 - Valid<br>  – 0xFF - Not Available |
| *MCC[NAS_PLMN_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters<br>• MCC wildcard value is returned as {'3', 0xFF, 0xFF}. |
| *MNC[NAS_PLMN_LENGTH]* | • Mobile Network Code.<br>• MNC digits in ASCII characters<br>• An unused byte is set to 0xFF.<br>• MNC wildcard value is returned as {'7', 0xFF, 0xFF}. |

### 8.136.2 Field Documentation

#### 8.136.2.1 uint16_t nas_CDMASysInfo::baseId

#### 8.136.2.2 uint32_t nas_CDMASysInfo::baseLat

#### 8.136.2.3 uint32_t nas_CDMASysInfo::baseLong

#### 8.136.2.4 uint8_t nas_CDMASysInfo::bsInfoValid

#### 8.136.2.5 uint8_t nas_CDMASysInfo::bsPRev

#### 8.136.2.6 uint8_t nas_CDMASysInfo::bsPRevValid

#### 8.136.2.7 uint8_t nas_CDMASysInfo::ccsSupported

#### 8.136.2.8 uint8_t nas_CDMASysInfo::ccsSupportedValid

#### 8.136.2.9 uint8_t nas_CDMASysInfo::cdmaSysIdValid

#### 8.136.2.10 uint8_t nas_CDMASysInfo::isSysPrlMatch

#### 8.136.2.11 uint8_t nas_CDMASysInfo::isSysPrlMatchValid

#### 8.136.2.12 uint8_t nas_CDMASysInfo::MCC[3]

#### 8.136.2.13 uint8_t nas_CDMASysInfo::MNC[3]

#### 8.136.2.14 uint16_t nas_CDMASysInfo::networkID

#### 8.136.2.15 uint8_t nas_CDMASysInfo::networkIdValid

#### 8.136.2.16 uint16_t nas_CDMASysInfo::packetZone

#### 8.136.2.17 uint8_t nas_CDMASysInfo::packetZoneValid

#### 8.136.2.18 uint8_t nas_CDMASysInfo::pRevInUse

#### 8.136.2.19 uint8_t nas_CDMASysInfo::pRevInUseValid

#### 8.136.2.20 nas_sysInfoCommon nas_CDMASysInfo::sysInfoCDMA

#### 8.136.2.21 uint16_t nas_CDMASysInfo::systemID

## 8.137 nas_CDMASysInfoExt Struct Reference

### Data Fields

- uint16_t MCC
- uint8_t imsi_11_12

### 8.137.1 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**

| | |
|---|---|
| *MCC* | • Mobile Country Code |
| *imsi_11_12* | • IMSI_11_12 |

### 8.137.2 Field Documentation

#### 8.137.2.1 uint8_t nas_CDMASysInfoExt::imsi_11_12

#### 8.137.2.2 uint16_t nas_CDMASysInfoExt::MCC

## 8.138 nas_cellParams Struct Reference

### Data Fields

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

### 8.138.1 Detailed Description

This structure contains information about the Cell parameters.

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID.<br>• Range: 0 to 503. |
| *rsrq* | • Current RSRQ in 1/10 dB as measured by L1.<br>• Range: -20.0 dB to -3.0 dB. |
| *rsrp* | • Current RSRP in 1/10 dBm as measured by L1.<br>• Range: -140.0 dBm to -44.0 dBm. |
| *rssi* | • Current RSSI in 1/10 dBm as measured by L1.<br>• Range: -120.0 dBm to 0. |
| *srxlev* | • Cell selection Rx level (Srxlev) value.<br>• Range: -128 to 128.<br>• This field is only valid when ue_in_idle is TRUE. |

### 8.138.2 Field Documentation

#### 8.138.2.1 uint16_t nas_cellParams::pci

#### 8.138.2.2 int16_t nas_cellParams::rsrp

#### 8.138.2.3 int16_t nas_cellParams::rsrq

#### 8.138.2.4 int16_t nas_cellParams::rssi

#### 8.138.2.5 int16_t nas_cellParams::srxlev

## 8.139 nas_ciotAcqOrderPref Struct Reference

### Data Fields

- uint8_t ciotAcqOrderLen
- uint32_t * pCiotAcqOrder

### 8.139.1 Detailed Description

Contain the CIOT Acquisition Order Preference.

**Parameters**

| | |
|---|---|
| *ciotAcqOrderLen* | • Number of sets of the following elements. |
| *pCiotAcqOrder* | • CIOT Acquisition order preference to be set.<br>• Values<br>– 0x01 - cdma2000 ®1X<br>– 0x02 - cdma2000 ® HRPD (1xEV-DO)<br>– 0x04 - GSM<br>– 0x05 - UMTS<br>– 0x08 - LTE<br>– 0x09 - TD-SCDMA<br>– 0x0a - LTE_M1<br>– 0x0b - LTE_NB1 |

### 8.139.2 Field Documentation

#### 8.139.2.1 uint8_t nas_ciotAcqOrderPref::ciotAcqOrderLen

#### 8.139.2.2 uint32_t∗ nas_ciotAcqOrderPref::pCiotAcqOrder

## 8.140 nas_CiotAcqOrderPrefTlv Struct Reference

### Data Fields

- uint8_t ciotAcqOrderLen
- uint32_t ∗ pCiotAcqOrder
- uint8_t TlvPresent

### 8.140.1 Detailed Description

Contain the CIOT Acquisition Order Preference.

**Parameters**

| | |
|---|---|
| *ciotAcqOrderLen* | • Number of sets of the following elements. |
| *pCiotAcqOrder* | • CIOT Acquisition order preference to be set.<br>• Values<br>– 0x01 - cdma2000 ®1X<br>– 0x02 - cdma2000 ® HRPD (1xEV-DO)<br>– 0x04 - GSM<br>– 0x05 - UMTS<br>– 0x08 - LTE<br>– 0x09 - TD-SCDMA<br>– 0x0a - LTE_M1<br>– 0x0b - LTE_NB1 |

| | |
|---|---|
| *TlvPresent* | • Tlv Present |

### 8.140.2 Field Documentation

#### 8.140.2.1 uint8_t nas_CiotAcqOrderPrefTlv::ciotAcqOrderLen

#### 8.140.2.2 uint32_t* nas_CiotAcqOrderPrefTlv::pCiotAcqOrder

#### 8.140.2.3 uint8_t nas_CiotAcqOrderPrefTlv::TlvPresent

## 8.141 nas_CiotLteOpModePrefTlv Struct Reference

### Data Fields

- uint64_t ciotLteOpModePref
- uint8_t TlvPresent

### 8.141.1 Detailed Description

Contain the CIOT LTE Operational Mode Preference.

**Parameters**

| | |
|---|---|
| *ciotLteOpMode-Pref* | • CIOT LTE Operational Mode Preference<br>• Values:<br>– Bit 0 (0x01) - LTE wideband<br>– Bit 1 (0x02) - LTE M1<br>– Bit 2 (0x04) - LTE NB1 All unlisted bits are reserved for future use and the service point ignores them if used. |
| *TlvPresent* | • Tlv Present |

### 8.141.2 Field Documentation

#### 8.141.2.1 uint64_t nas_CiotLteOpModePrefTlv::ciotLteOpModePref

#### 8.141.2.2 uint8_t nas_CiotLteOpModePrefTlv::TlvPresent

## 8.142 nas_CommInfo Struct Reference

### Data Fields

- int8_t temperature
- uint8_t modemMode
- uint8_t systemMode
- uint8_t imsRegState

- uint8_t psState

### 8.142.1 Detailed Description

Structure for storing the common information for the device.

**Parameters**

| | |
|---|---|
| *temperature* | • Temperature.<br>– 8-bit signed integer<br>– 0xFF - Not Available. |
| *modemMode* | • Modem Operating Mode.<br>– 0x00 - POWERING OFF<br>– 0x01 - FACTORY TEST<br>– 0x02 - OFFLINE<br>– 0x03 - OFFLINE_AMPS<br>– 0x04 - OFFLINE_CDMA<br>– 0x05 - ONLINE<br>– 0x06 - LOW POWER MODE<br>– 0x07 - RESETTING<br>– 0x08 - NETWORK TEST<br>– 0x09 - OFFLINE REQUEST<br>– 0x0A - PSEUDO ONLINE<br>– 0x0B - RESETTING MODEM<br>– 0xFF - Unknown |
| *systemMode* | • System Acquisition Mode.<br>– 0x00 - No service<br>– 0x01 - AMPS<br>– 0x02 - CDMA<br>– 0x03 - GSM<br>– 0x04 - HDR<br>– 0x05 - WCDMA<br>– 0x06 - GPS<br>– 0x08 - WLAN<br>– 0x09 - LTE<br>– 0xFF - Unknown |
| *imsRegState* | • IMS Registration State.<br>– 0x00 - NO SRV<br>– 0x01 - IN PROG<br>– 0x02 - FAILED<br>– 0x03 - LIMITED<br>– 0x04 - FULL SRV<br>– 0xFF - Unknown |

| | |
|---|---|
| *psState* | • PS Attach State.<br>– 0x00 - Attached<br>– 0x01 - Detached<br>– 0xFF - Unknown |

### 8.142.2 Field Documentation

#### 8.142.2.1 uint8_t nas_CommInfo::imsRegState

#### 8.142.2.2 uint8_t nas_CommInfo::modemMode

#### 8.142.2.3 uint8_t nas_CommInfo::psState

#### 8.142.2.4 uint8_t nas_CommInfo::systemMode

#### 8.142.2.5 int8_t nas_CommInfo::temperature

## 8.143 nas_CSGID Struct Reference

### Data Fields

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

### 8.143.1 Detailed Description

Contain the CSGID.

**Parameters**

| | |
|---|---|
| *mcc* | • MCC value. Range 0 to 999 |
| *mnc* | • MNC value. Range 0 to 999 |
| *mncPcsDigits* | • TRUE - MNC is a three-digit value; e.g., a reported value of 90 corresponds to an MNC value of 090<br>• FALSE - MNC is a two-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90 |
| *id* | • Closed subscriber group identifier. |

| | |
|---|---|
| *rat* | • Radio interface technology of the CSG network. Values:<br>– 0x04 - RADIO_IF_GSM - GSM<br>– 0x05 - RADIO_IF_UMTS - UMTS<br>– 0x08 - RADIO_IF_LTE - LTE<br>– 0x09 - RADIO_IF_TDSCDMA - TDS |

### 8.143.2 Field Documentation

#### 8.143.2.1 uint32_t nas_CSGID::id

#### 8.143.2.2 uint16_t nas_CSGID::mcc

#### 8.143.2.3 uint16_t nas_CSGID::mnc

#### 8.143.2.4 uint8_t nas_CSGID::mncPcsDigits

#### 8.143.2.5 uint8_t nas_CSGID::rat

## 8.144 nas_currentPLMN Struct Reference

### Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t netDescrLength
- uint8_t netDescr [255]

### 8.144.1 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**

| | |
|---|---|
| *MCC* | • mobile country code<br>– A 16 bit representation of MCC<br>– Range 0 to 999 |
| *MNC* | • mobile network code<br>– A 16 bit representation of MNC<br>– Range 0 to 999 |
| *netDescrLength* | • Length of Network description field<br>• Defaults to zero |

| | |
|---|---|
| *netDescr* | • Network Description<br>– optional string containing network name or description |

### 8.144.2 Field Documentation

#### 8.144.2.1 uint16_t nas_currentPLMN::MCC

#### 8.144.2.2 uint16_t nas_currentPLMN::MNC

#### 8.144.2.3 uint8_t nas_currentPLMN::netDescr[255]

#### 8.144.2.4 uint8_t nas_currentPLMN::netDescrLength

## 8.145 nas_dataSrvCapabilities Struct Reference

### Data Fields

- uint8_t dataCapabilitiesLen
- uint8_t dataCapabilities [32]

### 8.145.1 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**

| | |
|---|---|
| *dataCapabilities-Len* | • Length of data capabilities list<br>• Defaults to zero |
| *dataCapabilities* | • List of data capabilities<br>• Values:<br>– 0x01 - GPRS<br>– 0x02 - EDGE<br>– 0x03 - HSDPA<br>– 0x04 - HSUPA<br>– 0x05 - WCDMA<br>– 0x06 - CDMA<br>– 0x07 - EV-DO Rev0<br>– 0x08 - EV-DO RevA<br>– 0x09 - GSM<br>– 0x0A - EV-DO Rev B<br>– 0x0B - LTE<br>– 0x0C - HSDPA+<br>– 0x0D - DC-HSDPA+ |

### 8.145.2 Field Documentation

#### 8.145.2.1 uint8_t nas_dataSrvCapabilities::dataCapabilities[32]

#### 8.145.2.2 uint8_t nas_dataSrvCapabilities::dataCapabilitiesLen

## 8.146 nas_DataStatusDetail Struct Reference

### Data Fields

- uint32_t IPAddress
- uint8_t LastErrCode

### 8.146.1 Detailed Description

This structure contains data status detail

**Parameters**

| | |
|---|---|
| *IPAddress* | • IP Address<br>• 0xABCDEFGH - AB.CD.EF.GH<br>• Example:<br>  – 0x12345678 - 18.52.86.120<br>  0x12=18 0x34=52 0x56=86 0x78=120<br>• 0xFFFFFFFF - NA |

| | |
|---|---|
| *LastErrCode* | • MIP Error code<br>– 0x00 - MIP_RRP_CODE_SUCCESS<br>– 0x01 - MIP_RRP_CODE_SUCCESS_NO_SIM_BINDINGS<br>– 0x40 - MIP_RRP_CODE_FAILURE_FA_REASON_UNSPECIFIED<br>– 0x41 - MIP_RRP_CODE_FAILURE_FA_ADMIN_PROHIBITED<br>– 0x42 - MIP_RRP_CODE_FAILURE_FA_INSUFFICIENT_RESOURCES<br>– 0x43 - MIP_RRP_CODE_FAILURE_FA_MOBILE_NODE_FAILED_AUTH<br>– 0x44 - MIP_RRP_CODE_FAILURE_FA_HA_FAILED_AUTH<br>– 0x45 - MIP_RRP_CODE_FAILURE_FA_REQUESTED_LIFETIME_TOO_LONG<br>– 0x46 - MIP_RRP_CODE_FAILURE_FA_MALFORMED_REQUEST<br>– 0x47 - MIP_RRP_CODE_FAILURE_FA_MALFORMED_REPLY<br>– 0x48 - MIP_RRP_CODE_FAILURE_FA_ENCAPSULATION_UNAVAILABLE<br>– 0x49 - MIP_RRP_CODE_FAILURE_FA_VJHC_UNAVAILABLE<br>– 0x4A - MIP_RRP_CODE_FAILURE_FA_CANT_REV_TUN<br>– 0x4B - MIP_RRP_CODE_FAILURE_FA_MUST_REV_TUN<br>– 0x4C - MIP_RRP_CODE_FAILURE_FA_BAD_TTL<br>– 0x4D - MIP_RRP_CODE_FAILURE_INVALID_COA<br>– 0x4F - MIP_RRP_CODE_FAILURE_FA_DELIVERY_STYLE_NOT_SUPPORTED<br>– 0x59 - MIP_RRP_CODE_FAILURE_FA_VS_REASON<br>– 0x61 - MIP_RRP_CODE_FAILURE_MISSING_NAI<br>– 0x62 - MIP_RRP_CODE_FAILURE_MISSING_HA_ADDR<br>– 0x63 - MIP_RRP_CODE_FAILURE_MISSING_HOMEADDR<br>– 0x68 - MIP_RRP_CODE_FAILURE_UNKNOWN_CHALLENGE<br>– 0x69 - MIP_RRP_CODE_FAILURE_MISSING_CHALLENGE<br>– 0x6A - MIP_RRP_CODE_FAILURE_STALE_CHALLENGE<br>– 0x6B - MIP_RRP_CODE_FAILURE_MISSING_MN_FA<br>– 0x80 - MIP_RRP_CODE_FAILURE_HA_REASON_UNSPECIFIED<br>– 0x81 - MIP_RRP_CODE_FAILURE_HA_ADMIN_PROHIBITED<br>– 0x82 - MIP_RRP_CODE_FAILURE_HA_INSUFFICIENT_RESOURCES<br>– 0x83 - MIP_RRP_CODE_FAILURE_HA_MOBILE_NODE_FAILED_AUTH<br>– 0x84 - MIP_RRP_CODE_FAILURE_HA_FA_FAILED_AUTH<br>– 0x85 - MIP_RRP_CODE_FAILURE_HA_REG_ID_MISMATCH<br>– 0x86 - MIP_RRP_CODE_FAILURE_HA_MALFORMED_REQUEST<br>– 0x88 - MIP_RRP_CODE_FAILURE_UNKNOWN_HA<br>– 0x89 - MIP_RRP_CODE_FAILURE_HA_CANT_REV_TUN<br>– 0x8A - MIP_RRP_CODE_FAILURE_HA_MUST_REV_TUN<br>– 0x8B - MIP_RRP_CODE_FAILURE_HA_ENCAPSULATION_UNAVAILABLE<br>– 0x8F - MIP_RRP_CODE_FAILURE_REDIRECTED_HA<br>– 0x90 - MIP_RRP_CODE_FAILURE_HA_BAD_AAA_AUTH<br>– 0xFF - NA |

### 8.146.2 Field Documentation

#### 8.146.2.1 uint32_t nas_DataStatusDetail::IPAddress

#### 8.146.2.2 uint8_t nas_DataStatusDetail::LastErrCode

## 8.147 nas_detailSvcInfo Struct Reference

### Data Fields

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

### 8.147.1 Detailed Description

This structure contains Detailed Service information

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

**Parameters**

| | |
|---|---|
| *srvStatus* | • Service status<br>• Values:<br>– 0x00 - No service<br>– 0x01 - Limited service<br>– 0x02 - Service available<br>– 0x03 - Limited regional service<br>– 0x04 - MS in power save or deep sleep |
| *srvCapability* | • System's service capability<br>• Values:<br>– 0x00 - No Service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>– 0x04 - MS found the right system but not yet registered/attached |
| *hdrSrvStatus* | • HDR service status<br>• Values:<br>– 0x00 - No service<br>– 0x01 - Limited service<br>– 0x02 - Service available<br>– 0x03 - Limited regional service<br>– 0x04 - MS in power save or deep sleep |
| *hdrHybrid* | • HDR hybrid information<br>• Values:<br>– 0x00 - System is not hybrid<br>– 0x01 - System is hybrid |

| | |
|---|---|
| *isSysForbidden* | • Forbidden system information<br>• Values:<br>– 0x00 - System is not a forbidden system<br>– 0x01 - System is a forbidden system |

### 8.147.2 Field Documentation

#### 8.147.2.1 uint8_t nas_detailSvcInfo::hdrHybrid

#### 8.147.2.2 uint8_t nas_detailSvcInfo::hdrSrvStatus

#### 8.147.2.3 uint8_t nas_detailSvcInfo::isSysForbidden

#### 8.147.2.4 uint8_t nas_detailSvcInfo::srvCapability

#### 8.147.2.5 uint8_t nas_detailSvcInfo::srvStatus

## 8.148 nas_DeviceConfigDetail Struct Reference

### Data Fields

- uint8_t Technology
- uint8_t QLIC
- uint8_t Chipset
- uint8_t HWVersion

### 8.148.1 Detailed Description

This structure contains device configure detail.

**Parameters**

| | |
|---|---|
| *Technology* | • Current technology in use<br>– 0x00 - 1x RTT<br>– 0x01 - EVDO Rev 0<br>– 0x02 - EVDO Rev A<br>– 0xFF - NA |
| *QLIC* | • Quasi Linear Interference Cancellation<br>– 0x00 - Not supported<br>– 0x01 - Supported |
| *Chipset* | • Qualcomm platform<br>– 0x4E - MDM6200<br>– 0x4F - MDM6600<br>– 0xFF - NA |

| | |
|---|---|
| *HWVersion* | • Hardware version<br>– 0x00 - BSHWREV0<br>– 0x01 - BSHWREV1<br>– 0x02 - BSHWREV2<br>– 0x03 - BSHWREV3<br>– 0x04 - BSHWREVMAX<br>– 0xFF - BSHWREVUNKNOWN |

### 8.148.2 Field Documentation

#### 8.148.2.1 uint8_t nas_DeviceConfigDetail::Chipset

#### 8.148.2.2 uint8_t nas_DeviceConfigDetail::HWVersion

#### 8.148.2.3 uint8_t nas_DeviceConfigDetail::QLIC

#### 8.148.2.4 uint8_t nas_DeviceConfigDetail::Technology

## 8.149 nas_dirNum Struct Reference

### Data Fields

- uint8_t dirNumLen
- uint8_t dirNum [255]

### 8.149.1 Detailed Description

This structure contains the parameters for Directory Number Information

**Parameters**

| | |
|---|---|
| *dirNumLen* | • Number of sets of the following elements:<br>– dir_num<br>• If zero(0), then no information follows. |
| *dirNum* | • Directory number in ASCII characters. |

### 8.149.2 Field Documentation

#### 8.149.2.1 uint8_t nas_dirNum::dirNum[255]

#### 8.149.2.2 uint8_t nas_dirNum::dirNumLen

## 8.150 nas_DRCParams Struct Reference

### Data Fields

- uint8_t DRCValue
- uint8_t DRCCover

### 8.150.1 Detailed Description

This structure contains DRC information.

**Parameters**

| | |
|---|---|
| *DRCValue* | • Current Data Rate Channel |
| *DRCCover* | • Current Data Rate Channel cover |

### 8.150.2 Field Documentation

#### 8.150.2.1 uint8_t nas_DRCParams::DRCCover

#### 8.150.2.2 uint8_t nas_DRCParams::DRCValue

## 8.151 nas_ecioListElement Struct Reference

### Data Fields

- int16_t ecio
- uint8_t radioIf

### 8.151.1 Detailed Description

This structure contains the ECIO Information

**Parameters**

| | |
|---|---|
| *ecio* | • ECIO value in dBm |
| *radioIf* | • Radio interface technology of the signal being measured<br>– 0x00 – RADIO_IF_NO_SVC – None (no service)<br>– 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X<br>– 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000 HRPD (1xEV-DO)<br>– 0x03 – RADIO_IF_AMPS – AMPS<br>– 0x04 – RADIO_IF_GSM – GSM<br>– 0x05 – RADIO_IF_UMTS – UMTS |

### 8.151.2 Field Documentation

#### 8.151.2.1 int16_t nas_ecioListElement::ecio

#### 8.151.2.2 uint8_t nas_ecioListElement::radioIf

## 8.152 nas_ECIOThresh Struct Reference

### Data Fields

- uint8_t ECIOThresListLen
- int16_t * pECIOThresList

### 8.152.1 Detailed Description

This sturcture contain ECIO threshold list.

**Parameters**

| | |
|---|---|
| *ECIOThresList-Len* | • Length of the ECIO threshold list parameter to follow |
| *pECIOThresList* | • Sequence of thresholds delimiting ECIO event reporting bands<br>• Every time a new ECIO value crosses a threshold value, an event report indication message with the new ECIO value is sent to the requesting control point. For this field<br>– Each ECIO threshold value is a signed 2 byte value<br>– Each ECIO threshold value increments in negative 0.5 dB, e.g., an ECIO threshold value of 2 means -1dB<br>– Maximum number of threshold values is 16<br>– At least one value must be specified<br>– Threshold values specified above are used for all RATs |

### 8.152.2 Field Documentation

#### 8.152.2.1 uint8_t nas_ECIOThresh::ECIOThresListLen

#### 8.152.2.2 int16_t* nas_ECIOThresh::pECIOThresList

## 8.153 nas_EmerModeTlv Struct Reference

### Data Fields

- uint8_t EmerMode
- uint8_t TlvPresent

### 8.153.1 Detailed Description

Contain the emergency Mode parameters for system selection preferences.

**Parameters**

| | |
|---|---|
| *EmerMode* | • Emergency Mode<br>• Values:<br>– 0 - OFF (normal)<br>– 1 - ON (Emergency) |
| *TlvPresent* | • Tlv Present |

### 8.153.2 Field Documentation

#### 8.153.2.1 uint8_t nas_EmerModeTlv::EmerMode

#### 8.153.2.2 uint8_t nas_EmerModeTlv::TlvPresent

## 8.154 nas_errorRateListElement Struct Reference

### Data Fields

- uint16_t errorRate
- uint8_t radioIf

### 8.154.1 Detailed Description

This structure contains the Error Rate Information

**Parameters**

| | |
|---|---|
| *errorRate* | • Error rate value corresponds to the RAT that is currently registered.<br>– For CDMA, the error rate reported is Frame Error Rate:<br>* 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%<br>* A value of 0xFFFF indicates that the error rate is unknown or unavailable<br>– For HDR, the error rate reported is Packet Error Rate:<br>* 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%<br>* A value of 0xFFFF indicates that the error rate is unknown or unavailable<br>– For GSM, the error rate reported is Bit Error Rate:<br>* 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.<br>* A value of 25500 indicates No Data<br>– For WCDMA, the error rate reported is Block Error Rate (BLER):<br>* Valid values are 1 to 10000<br>* The reported value divided by 100 provides the error rate in percentages, e.g., a value of 300 represents a BLER of 3%.<br>* A value of 0 indicates No Data |

| | |
|---|---|
| *radioIf* | • Radio interface technology of the signal being measured<br>– 0x00 – RADIO_IF_NO_SVC – None (no service)<br>– 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X<br>– 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000 HRPD (1xEV-DO)<br>– 0x03 – RADIO_IF_AMPS – AMPS<br>– 0x04 – RADIO_IF_GSM – GSM<br>– 0x05 – RADIO_IF_UMTS – UMTS |

### 8.154.2 Field Documentation

#### 8.154.2.1 uint16_t nas_errorRateListElement::errorRate

#### 8.154.2.2 uint8_t nas_errorRateListElement::radioIf

## 8.155 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]

### 8.155.1 Detailed Description

This structure contains information about the GERAN Network.

**Parameters**

| | |
|---|---|
| *cellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present. |
| *plmn[NAS_PLMN_LENGTH]* | • MCC/MNC information coded as octet 3, 4, and 5.<br>• This field is ignored when nmrCellID is not present. |
| *lac* | • Location area code.<br>• This field is ignored when nmrCellID is not present.<br>– 0xFFFF - Not Available |
| *arfcn* | • Absolute RF channel number.<br>– 0xFFFF - Not Available |

| | |
|---|---|
| *bsic* | • Base station identity code.<br>– 0xFF - Not Available |
| *timingAdvance* | • 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.<br>– 0xFFFFFFFF - Not Available |
| *rxLev* | • Serving Cell Rx measurement.<br>• Values range between 0 and 63.<br>• Mapped to a measured signal level:<br>– Rxlev 0 is a signal strength less than -110 dBm<br>– Rxlev 1 is -110 dBm to -109 dBm<br>– Rxlev 2 is -109 dBm to -108 dBm<br>– ...<br>– Rxlev 62 is -49 dBm to -48 dBm<br>– Rxlev 63 is greater than -48 dBm<br>– 0xFFFF - Not Available |
| *nmrInst* | • Provides the number of set of instances which follow.<br>• If 0(zero), then no information follows it. |
| *insNmrCellInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_nmrCellInfo for more information. |

### 8.155.2 Field Documentation

#### 8.155.2.1 uint16_t nas_GERANInfo::arfcn

#### 8.155.2.2 uint8_t nas_GERANInfo::bsic

#### 8.155.2.3 uint32_t nas_GERANInfo::cellID

#### 8.155.2.4 nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]

#### 8.155.2.5 uint16_t nas_GERANInfo::lac

#### 8.155.2.6 uint8_t nas_GERANInfo::nmrInst

#### 8.155.2.7 uint8_t nas_GERANInfo::plmn[3]

#### 8.155.2.8 uint16_t nas_GERANInfo::rxLev

#### 8.155.2.9 uint32_t nas_GERANInfo::timingAdvance

## 8.156 nas_geranInstInfo Struct Reference

### Data Fields

- uint16_t geranArfcn
- uint8_t geranBsicNcc
- uint8_t geranBsicBcc
- int16_t geranRssi

### 8.156.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *geranArfcn* | • Absolute RF channel number. |
| *geranBsicNcc* | • Base station identity code network color code.<br>• 0xFF indicates information is not present. |
| *geranBsicBcc* | • Base station identity code base station color code.<br>• 0xFF indicates information is not present. |
| *geranRssi* | • Received signal strength indicator. |

### 8.156.2 Field Documentation

#### 8.156.2.1 uint16_t nas_geranInstInfo::geranArfcn

#### 8.156.2.2 uint8_t nas_geranInstInfo::geranBsicBcc

#### 8.156.2.3 uint8_t nas_geranInstInfo::geranBsicNcc

#### 8.156.2.4 int16_t nas_geranInstInfo::geranRssi

## 8.157 nas_gsmCellInfo Struct Reference

### Data Fields

- uint16_t arfcn
- uint8_t band1900
- uint8_t cellIdValid
- uint8_t bsicId
- int16_t rssi
- int16_t srxlev

### 8.157.1 Detailed Description

This structure contains information about the GSM Cell.

**Parameters**

| | |
|---|---|
| *arfcn* | • GSM frequency being reported.<br>• Range: 0 to 1023. |
| *band1900* | • Band indicator for the GSM ARFCN<br>• This field is only valid if arfcn is in the overlapping region.<br>• If TRUE and the cell is in the overlapping region, the ARFCN is on the 1900 band.<br>• If FALSE, it is on the 1800 band. |
| *cellIdValid* | • Flag indicating whether the base station identity code ID is valid. |
| *bsicId* | • Base station identity code ID, including base station color code and network color code.<br>• The lower 6 bits can be set to any value. |
| *rssi* | • Measured RSSI value in 1/10 dB.<br>• Range: -200.0 dB to 0 |
| *srxlev* | • Cell selection Rx level (Srxlev) value.<br>• Range: -128 to 128.<br>• This field is only valid when ue_in_idle is TRUE. |

### 8.157.2 Field Documentation

#### 8.157.2.1 uint16_t nas_gsmCellInfo::arfcn

#### 8.157.2.2 uint8_t nas_gsmCellInfo::band1900

#### 8.157.2.3 uint8_t nas_gsmCellInfo::bsicId

#### 8.157.2.4 uint8_t nas_gsmCellInfo::cellIdValid

#### 8.157.2.5 int16_t nas_gsmCellInfo::rssi

#### 8.157.2.6 int16_t nas_gsmCellInfo::srxlev

## 8.158 nas_GSMRSSIThresh Struct Reference

### Data Fields

- uint8_t GSMRSSIThreshListLen
- int16_t * pGSMRSSIThreshList

### 8.158.1 Detailed Description

This structure contains GSM RSSI threshold related parameters.

**Parameters**

| | |
|---|---|
| *GSMRSSI-ThreshListLen* | • Length of the GSM RSSI threshold list parameter to follow |
| *pGSMRSSI-ThreshList* | • Array of RSSI thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values<br>• Range for RSSI values: -111 to -48 (in dBm) |

### 8.158.2 Field Documentation

#### 8.158.2.1 uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen

#### 8.158.2.2 int16_t∗ nas_GSMRSSIThresh::pGSMRSSIThreshList

## 8.159 nas_GSMSrvStatusInfo Struct Reference

### Data Fields

- uint8_t srvStatus
- uint8_t trueSrvStatus
- uint8_t isPrefDataPath

### 8.159.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *srvStatus* | • Service status of the system.<br>– 0x00 - No service<br>– 0x01 - Limited service<br>– 0x02 - Service<br>– 0x03 - Limited regional service<br>– 0x04 - Power save<br>– 0xFF - Not Available |
| *trueSrvStatus* | • True service status of the system.<br>• Not applicable to CDMA/HDR.<br>– 0x00 - No service<br>– 0x01 - Limited service<br>– 0x02 - Service<br>– 0x03 - Limited regional service<br>– 0x04 - Power save<br>– 0xFF - Not Available |

| | |
|---|---|
| *isPrefDataPath* | • Whether the RAT is the preferred data path.<br>– 0x00 - Not preferred<br>– 0x01 - Preferred<br>– 0xFF - Not Available |

### 8.159.2 Field Documentation

#### 8.159.2.1 uint8_t nas_GSMSrvStatusInfo::isPrefDataPath

#### 8.159.2.2 uint8_t nas_GSMSrvStatusInfo::srvStatus

#### 8.159.2.3 uint8_t nas_GSMSrvStatusInfo::trueSrvStatus

## 8.160 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

### 8.160.1 Detailed Description

Structure for storing the GSM System Information.

**Parameters**

| | |
|---|---|
| *sysInfoGSM* | • See nas_sysInfoCommon for more information. |
| *lacValid* | • Indicates whether the location area code is valid..<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |

| | |
|---|---|
| *lac* | • Location area code.<br>• Only applies to 3GPP.<br>– 0xFFFF - Not Available |
| *cellIdValid* | • Indicates whether the cell ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *cellId* | • Cell ID.<br>– 0xFFFFFFFF - Not Available |
| *regRejectInfo-Valid* | • Indicates whether the registration reject information is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *rejectSrvDomain* | • Type of service domain in which the registration is rejected.<br>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>– 0x04 - Camped<br>– 0xFF - Not Available |
| *rejCause* | • Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6 and 10.5.5.14]<br>– 0xFF - Not Available |
| *networkIdValid* | • Indicates whether the network ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *MCC[NAS_PLM-N_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters |
| *MNC[NAS_PLM-N_LENGTH]* | • Mobile Network Code.<br>• MNC digits in ASCII characters<br>• An unused byte is set to 0xFF.<br>• 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. |

| | |
|---|---|
| *egprsSuppValid* | • Indicates whether the EGPRS support is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *egprsSupp* | • EGPRS support indication.<br>• Only applicable for GSM.<br>– 0x00 - Not available<br>– 0x01 - Available<br>– 0xFF - Not Available |
| *dtmSuppValid* | • Indicates whether Dual Transfer mode support is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *dtmSupp* | • Dual Transfer mode support indication.<br>• Only applicable for GSM.<br>– 0x00 - Not available<br>– 0x01 - Available<br>– 0xFF - Not Available |

### 8.160.2 Field Documentation

#### 8.160.2.1 uint32_t nas_GSMSysInfo::cellId

#### 8.160.2.2 uint8_t nas_GSMSysInfo::cellIdValid

#### 8.160.2.3 uint8_t nas_GSMSysInfo::dtmSupp

#### 8.160.2.4 uint8_t nas_GSMSysInfo::dtmSuppValid

#### 8.160.2.5 uint8_t nas_GSMSysInfo::egprsSupp

#### 8.160.2.6 uint8_t nas_GSMSysInfo::egprsSuppValid

#### 8.160.2.7 uint16_t nas_GSMSysInfo::lac

#### 8.160.2.8 uint8_t nas_GSMSysInfo::lacValid

#### 8.160.2.9 uint8_t nas_GSMSysInfo::MCC[3]

#### 8.160.2.10 uint8_t nas_GSMSysInfo::MNC[3]

#### 8.160.2.11 uint8_t nas_GSMSysInfo::networkIdValid

#### 8.160.2.12 uint8_t nas_GSMSysInfo::regRejectInfoValid

#### 8.160.2.13 uint8_t nas_GSMSysInfo::rejCause

#### 8.160.2.14 uint8_t nas_GSMSysInfo::rejectSrvDomain

#### 8.160.2.15 nas_sysInfoCommon nas_GSMSysInfo::sysInfoGSM

## 8.161 nas_GWAcqOrderPrefTlv Struct Reference

### Data Fields

- uint32_t GWAcqOrderPref
- uint8_t TlvPresent

### 8.161.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *GWAcqOrder-Pref* | • Parameter indicating GSM/WCDMA Acquisition order Preference<br>• Values:<br>– 0x00 - Automatic<br>– 0x01 - GSM then WCDMA<br>– 0x02 - WCDMA then GSM |
| *TlvPresent* | • Tlv Present |

### 8.161.2 Field Documentation

#### 8.161.2.1 uint32_t nas_GWAcqOrderPrefTlv::GWAcqOrderPref

#### 8.161.2.2 uint8_t nas_GWAcqOrderPrefTlv::TlvPresent

## 8.162 nas_HDRECIOThresh Struct Reference

### Data Fields

- uint8_t HDRECIOThreshListLen
- int16_t * pHDRECIOThreshList

### 8.162.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

**Parameters**

| | |
|---|---|
| *HDRECIO-ThreshListLen* | • Length of the HDR ECIO threshold list parameter to follow |

| | |
|---|---|
| *pHDRECIO-ThreshList* | • Array of ECIO thresholds (in units of 0.1 dB)<br>• Maximum of 32 values<br>• Range for ECIO values: -31.5 to 0 (in dB). |

### 8.162.2 Field Documentation

#### 8.162.2.1 uint8_t nas_HDRECIOThresh::HDRECIOThreshListLen

#### 8.162.2.2 int16_t∗ nas_HDRECIOThresh::pHDRECIOThreshList

## 8.163 nas_HDRIOThresh Struct Reference

### Data Fields

- uint8_t HDRIOThreshListLen
- int16_t ∗ pHDRIOThreshList

### 8.163.1 Detailed Description

This structure contains HDR IO threshold related parameters.

**Parameters**

| | |
|---|---|
| *HDRIOThresh-ListLen* | • Length of the HDR IO threshold list parameter to follow |
| *pHDRIOThresh-List* | • Array of IO thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values<br>• Range for IO values: -128 to -13 (in dBm). |

### 8.163.2 Field Documentation

#### 8.163.2.1 uint8_t nas_HDRIOThresh::HDRIOThreshListLen

#### 8.163.2.2 int16_t∗ nas_HDRIOThresh::pHDRIOThreshList

## 8.164 nas_HDRPersonality_Ind_Data Struct Reference

### Data Fields

- uint16_t ∗ pCurrentPersonality
- uint8_t ∗ pPersonalityListLength
- nas_protocolSubtypeElement ∗ pProtocolSubtypeElement

### 8.164.1 Detailed Description

Elements for HDR Personality indication.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality* | • Current active personality index.<br>• Bit to check in ParamPresenceMask - **16** |
| *pPersonalityList-Length* | • Number of Personality Protocol Subtype contains in this response.<br>• Bit to check in ParamPresenceMask - **17** |
| *pProtocol-SubtypeElement* | • See nas_protocolSubtypeElement for more information.<br>• Bit to check in ParamPresenceMask - **17** |

### 8.164.2 Field Documentation

#### 8.164.2.1 uint16_t∗ nas_HDRPersonality_Ind_Data::pCurrentPersonality

#### 8.164.2.2 uint8_t∗ nas_HDRPersonality_Ind_Data::pPersonalityListLength

#### 8.164.2.3 nas_protocolSubtypeElement∗ nas_HDRPersonality_Ind_Data::pProtocolSubtypeElement

## 8.165 nas_HDRRSSIThresh Struct Reference

### Data Fields

- uint8_t HDRRSSIThreshListLen
- int16_t ∗ pHDRRSSIThreshList

### 8.165.1 Detailed Description

This structure contains HDR RSSI threshold related parameters.

**Parameters**

| | |
|---|---|
| *HDRRSSI-ThreshListLen* | • Length of the HDR RSSI threshold list parameter to follow |
| *pHDRRSSI-ThreshList* | • Array of RSSI thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values.<br>• Range for RSSI values: -118 to -13 (in dBm). |

### 8.165.2 Field Documentation

#### 8.165.2.1 uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen

#### 8.165.2.2 int16_t∗ nas_HDRRSSIThresh::pHDRRSSIThreshList

## 8.166 nas_HDRSINRThresh Struct Reference

### Data Fields

- uint8_t HDRSINRThresListLen
- uint8_t ∗ pHDRSINRThresList

### 8.166.1 Detailed Description

This structure contains HDR SINR threshold list.

**Parameters**

| | |
|---|---|
| *HDRSINRThres-ListLen* | • Length of the HDR SINR threshold list parameter to follow |
| *pHDRSINR-ThresList* | • Sequence of thresholds delimiting SINR event reporting bands<br>• Every time a new SINR value crosses a threshold value, an event report indication message with the new SINR value is sent to the requesting control point. For this field<br>– SINR is reported only for HDR<br>– Each SINR threshold value is an unsigned 1 byte value<br>– Maximum number of threshold values is 16<br>– At least one value must be specified |

### 8.166.2 Field Documentation

#### 8.166.2.1 uint8_t nas_HDRSINRThresh::HDRSINRThresListLen

#### 8.166.2.2 uint8_t∗ nas_HDRSINRThresh::pHDRSINRThresList

## 8.167 nas_HDRSINRThreshold Struct Reference

### Data Fields

- uint8_t HDRSINRThreshListLen
- uint16_t ∗ pHDRSINRThreshList

### 8.167.1 Detailed Description

This structure contains HDR SINR threshold related parameters.

**Parameters**

| | |
|---|---|
| *HDRSINR-ThreshListLen* | • Length of the HDR ECIO threshold list parameter to follow |

| | |
|---|---|
| *pHDRSINR-ThreshList* | • Array of SINR level thresholds (in units of 1)<br>• maximum of 32 values.<br>• Valid levels are 0 to 8<br>– 0x00 - SINR_LEVEL_0 is -9 dB<br>– 0x01 - SINR_LEVEL_1 is -6 dB<br>– 0x02 - SINR_LEVEL_2 is -4.5 dB<br>– 0x03 - SINR_LEVEL_3 is -3 dB<br>– 0x04 - SINR_LEVEL_4 is -2 dB<br>– 0x05 - SINR_LEVEL_5 is +1 dB<br>– 0x06 - SINR_LEVEL_6 is +3 dB<br>– 0x07 - SINR_LEVEL_7 is +6 dB<br>– 0x08 - SINR_LEVEL_8 is +9 dB |

### 8.167.2 Field Documentation

#### 8.167.2.1 uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen

#### 8.167.2.2 uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList

## 8.168 nas_HDRSysInfo Struct Reference

### Data Fields

- nas_sysInfoCommon sysInfoHDR
- uint8_t isSysPrlMatchValid
- uint8_t isSysPrlMatch
- uint8_t hdrPersonalityValid
- uint8_t hdrPersonality
- uint8_t hdrActiveProtValid
- uint8_t hdrActiveProt
- uint8_t is856SysIdValid
- uint8_t is856SysId [16]

### 8.168.1 Detailed Description

Structure for storing the HDR System Information.

**Parameters**

| | |
|---|---|
| *sysInfoHDR* | • See nas_sysInfoCommon for more information. |
| *isSysPrlMatch-Valid* | • Indicates whether the system PRL match is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |

| | |
|---|---|
| *isSysPrlMatch* | • Indicates whether the system is in a PRL.<br>• Only applies to CDMA/HDR.<br>– 0x00 - System is not in a PRL<br>– 0x01 - System is in a PRL<br>– 0xFF - Not Available<br>• If the system is not in a PRL, roam_status carries the value from the default roaming indicator in the PRL.<br>• If the system is in a PRL, roam_status is set to the value based on the standard specification. |
| *hdrPersonality-Valid* | • Indicates whether the HDR personality is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *hdrPersonality* | • HDR personality information.<br>• Only applicable for HDR.<br>– 0x00 - None<br>– 0x02 - HRPD<br>– 0x03 - eHRPD<br>– 0xFF - Not Available |
| *hdrActiveProt-Valid* | • Indicates whether the HDR active protocol revision information is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *hdrActiveProt* | • HDR active protocol revision information .<br>• Only applicable for HDR.<br>– 0x00 - None<br>– 0x02 - HDR Rel 0<br>– 0x03 - HDR Rel A<br>– 0x04 - HDR Rel B<br>– 0xFF - Not Available |
| *is856SysIdValid* | • Indicates whether the IS-856 system ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *is856SysId[SLQ-S_SYSTEM_ID-_SIZE]* | • IS-856 system ID.<br>• Only applicable for HDR. |

### 8.168.2 Field Documentation

#### 8.168.2.1 uint8_t nas_HDRSysInfo::hdrActiveProt

#### 8.168.2.2 uint8_t nas_HDRSysInfo::hdrActiveProtValid

#### 8.168.2.3 uint8_t nas_HDRSysInfo::hdrPersonality

#### 8.168.2.4 uint8_t nas_HDRSysInfo::hdrPersonalityValid

#### 8.168.2.5 uint8_t nas_HDRSysInfo::is856SysId[16]

#### 8.168.2.6 uint8_t nas_HDRSysInfo::is856SysIdValid

#### 8.168.2.7 uint8_t nas_HDRSysInfo::isSysPrlMatch

#### 8.168.2.8 uint8_t nas_HDRSysInfo::isSysPrlMatchValid

#### 8.168.2.9 nas_sysInfoCommon nas_HDRSysInfo::sysInfoHDR

## 8.169 nas_homeSIDNID Struct Reference

### Data Fields

- uint8_t numInstances
- nas_sidNid SidNid [255]

### 8.169.1 Detailed Description

This structure contains Home SID/NID

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of sets of the following elements:<br>– sid<br>– nid<br>• If zero(0), then no information follows. |
| *SidNid* | • See nas_sidNid for more information |

### 8.169.2 Field Documentation

#### 8.169.2.1 uint8_t nas_homeSIDNID::numInstances

#### 8.169.2.2 nas_sidNid nas_homeSIDNID::SidNid[255]

## 8.170 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]

## 8.170.1 Detailed Description

This structure contains information about the inter-frequency.

**Parameters**

| | |
|---|---|
| *earfcn* | • E-UTRA absolute radio frequency channel number of the serving cell.<br>• Range: 0 to 65535. |
| *threshXLow* | • Cell Srxlev low threshold.<br>• Range: 0 to 31.<br>• 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. |
| *threshXHigh* | • Cell Srxlev high threshold.<br>• Range: 0 to 31.<br>• 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. |
| *cell_resel_priority* | • Cell re-selection priority<br>• Range: 0 to 7.<br>• This field is only valid when ue_in_idle is TRUE. |
| *cells_len* | • Provides the number of set of cell params. |
| *cellInterFreqParams[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_cellParams for more information. |

## 8.170.2 Field Documentation

### 8.170.2.1 uint8_t nas_infoInterFreq::cell_resel_priority

### 8.170.2.2 nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]

### 8.170.2.3 uint8_t nas_infoInterFreq::cells_len

### 8.170.2.4 uint16_t nas_infoInterFreq::earfcn

#### 8.170.2.5 uint8_t nas_infoInterFreq::threshXHigh

#### 8.170.2.6 uint8_t nas_infoInterFreq::threshXLow

## 8.171 nas_IOThresh Struct Reference

### Data Fields

- uint8_t IOThresListLen
- int32_t * pIOThresList

### 8.171.1 Detailed Description

This structure contains IO threshold list.

**Parameters**

| | |
|---|---|
| *IOThresListLen* | • Length of the IO threshold list parameter to follow |
| *pIOThresList* | • Sequence of thresholds delimiting IO event reporting bands<br>• Every time a new IO value crosses a threshold value, an event report indication message with the new IO value is sent to the requesting control point. For this field<br>– IO is applicable only for HDR<br>– Each IO threshold value is a signed 4 byte value<br>– Maximum number of threshold values is 16<br>– At least one value must be specified |

### 8.171.2 Field Documentation

#### 8.171.2.1 uint8_t nas_IOThresh::IOThresListLen

#### 8.171.2.2 int32_t* nas_IOThresh::pIOThresList

## 8.172 nas_IteBandPrefExt Struct Reference

### Data Fields

- uint64_t bits_1_64
- uint64_t bits_65_128
- uint64_t bits_129_192
- uint64_t bits_193_256

### 8.172.1 Detailed Description

Contain the LTE Band Preference Extended parameters

**Parameters**

| | |
|---|---|
| *bits_1_64* | • Bits 1 to 64 of the 256-bit LTE E-UTRA Operating Band bitmask. |
| *bits_65_128* | • Bits 65 to 128 of the 256-bit LTE E-UTRA Operating Band bitmask. |
| *bits_129_192* | • Bits 129 to 192 of the 256-bit LTE E-UTRA Operating Band bitmask. |
| *bits_193_256* | • Bits 193 to 256 of the 256-bit LTE E-UTRA Operating Band bitmask. |

### 8.172.2 Field Documentation

#### 8.172.2.1 uint64_t nas_lteBandPrefExt::bits_129_192

#### 8.172.2.2 uint64_t nas_lteBandPrefExt::bits_193_256

#### 8.172.2.3 uint64_t nas_lteBandPrefExt::bits_1_64

#### 8.172.2.4 uint64_t nas_lteBandPrefExt::bits_65_128

## 8.173 nas_LTEBandPrefTlv Struct Reference

### Data Fields

- uint64_t LTEBandPref
- uint8_t TlvPresent

### 8.173.1 Detailed Description

Contain the LTE Band Preference for system selection preferences.

**Parameters**

| | |
|---|---|
| *LTEBandPref* | • Bit mask representing the LTE band preference<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 22 - E-UTRA Operating Band 23<br>– Bit 23 - E-UTRA Operating Band 24<br>– Bit 24 - E-UTRA Operating Band 25<br>– Bit 25 - E-UTRA Operating Band 26<br>– Bit 27 - E-UTRA Operating Band 28<br>– Bit 28 - E-UTRA Operating Band 29<br>– Bit 29 - E-UTRA Operating Band 32<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– Bit 40 - E-UTRA Operating Band 41<br>– Bit 41 - E-UTRA Operating Band 42<br>– Bit 42 - E-UTRA Operating Band 43<br>– Bit 60 - E-UTRA Operating Band 125<br>– All other bits are reserved |
| *TlvPresent* | • Tlv Present |

### 8.173.2 Field Documentation

#### 8.173.2.1 uint64_t nas_LTEBandPrefTlv::LTEBandPref

#### 8.173.2.2 uint8_t nas_LTEBandPrefTlv::TlvPresent

## 8.174 nas_lteEARFCN Struct Reference

### Data Fields

- uint8_t status
- uint32_t earfcn0
- uint32_t earfcn1

### 8.174.1 Detailed Description

This structure contains LTE EARFCN infromation.

**Parameters**

| | |
|---|---|
| *status* | • 0 - Disable<br>• 1 - Enable |
| *earfcn0* | • Primary DL EARFCN to which the UE is locked |
| *earfcn1* | • Secondary DL EARFCN to which the UE is locked<br>• Note : Make earfcn1 value equal to earfcn0 if only one EARFCN is desired. |

### 8.174.2 Field Documentation

#### 8.174.2.1 uint32_t nas_lteEARFCN::earfcn0

#### 8.174.2.2 uint32_t nas_lteEARFCN::earfcn1

#### 8.174.2.3 uint8_t nas_lteEARFCN::status

## 8.175 nas_lteGsmCellInfo Struct Reference

### Data Fields

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

### 8.175.1 Detailed Description

This structure contains information about the LTE GSM Cell.

**Parameters**

| | |
|---|---|
| *cellReselPriority* | • Priority of this frequency group.<br>• Range: 0 to 7.<br>• This field is only valid when ue_in_idle is TRUE. |
| *threshGsmHigh* | • Reselection threshold for high priority layers.<br>• Range: 0 to 31.<br>• This field is only valid when ue_in_idle is TRUE. |
| *threshGsmLow* | • Reselection threshold for low priority layers.<br>• Range: 0 to 31.<br>• This field is only valid when ue_in_idle is TRUE. |
| *nccPermitted* | • Bitmask specifying whether a neighbor with a specific network color code is to be reported.<br>• Range: 0 to 255.<br>• Bit n set to 1 means a neighbor with NCC n must be included in the report. This flag is synonymous with a blacklist in other RATs.<br>• This field is only valid when ue_in_idle is TRUE. |
| *cells_len* | • Provides the number of set of gsm cells. |
| *GsmCellInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_gsmCellInfo for more information. |

### 8.175.2 Field Documentation

#### 8.175.2.1 uint8_t nas_lteGsmCellInfo::cellReselPriority

#### 8.175.2.2 uint8_t nas_lteGsmCellInfo::cells_len

#### 8.175.2.3 nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]

#### 8.175.2.4 uint8_t nas_lteGsmCellInfo::nccPermitted

#### 8.175.2.5 uint8_t nas_lteGsmCellInfo::threshGsmHigh

#### 8.175.2.6 uint8_t nas_lteGsmCellInfo::threshGsmLow

## 8.176 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

### 8.176.1 Detailed Description

Structure for storing the LTE information for the device.

**Parameters**

| | |
|---|---|
| *band* | • LTE Band<br>– 1 ∼ 40 (Band in decimal)<br>– 0xFF - Invalid |
| *bandwidth* | • BandWidth.<br>– 0x00 - 1.4 MHz<br>– 0x01 - 3 MHz<br>– 0x02 - 5 MHz<br>– 0x03 - 10 MHz<br>– 0x04 - 15 MHz<br>– 0x05 - 20 MHz<br>– 0x06 - Invalid<br>– 0xFF - Unknown |
| *RXChan* | • RX channel number in decimal<br>– 0xFFFF - Not Available |
| *TXChan* | • TX channel number in decimal<br>– 0xFFFF - Not Available |
| *emmState* | • EMM State.<br>– 0x00 - Deregistered<br>– 0x01 - Reg Initiated<br>– 0x02 - Registered<br>– 0x03 - TAU Initiated<br>– 0x04 - SR Initiated<br>– 0x05 - Dereg Initiated<br>– 0x06 - Invalid<br>– 0xFF - Unknown |

| | |
|---|---|
| *emmSubState* | • EMM Sub State.<br>– 0xFF - NOT Applicable<br>• When EMM_state is 0x00:<br>– 0x00 - No IMSI<br>– 0x01 - PLMN Search<br>– 0x02 - Attach Needed<br>– 0x03 - No Cell<br>– 0x04 - Attaching<br>– 0x05 - Normal Service<br>– 0x06 - Limited Service<br>– 0x07 - Waiting for PDN<br>• When EMM_state is 0x01:<br>– 0x00 - Waiting for NW<br>– 0x01 - Waiting for ESM<br>• When EMM_state is 0x02:<br>– 0x00 - Normal Service<br>– 0x01 - Update Needed<br>– 0x02 - Attempt Update<br>– 0x03 - No Cell<br>– 0x04 - PLMN Search<br>– 0x05 - Limited Service<br>– 0x06 - MM Update<br>– 0x07 - IMSI Detach<br>– 0x08 - Waiting for ESM |
| *emmConnState* | • EMM Connected Mode State.<br>– 0x00 - RRC Idle<br>– 0x01 - Waiting RRC Cfm<br>– 0x02 - RRC Connected<br>– 0x03 - RRC Releasing<br>– 0xFF - Unknown |

## 8.176.2 Field Documentation

### 8.176.2.1 uint8_t nas_LTEInfo::band

### 8.176.2.2 uint8_t nas_LTEInfo::bandwidth

### 8.176.2.3 uint8_t nas_LTEInfo::emmConnState

### 8.176.2.4 uint8_t nas_LTEInfo::emmState

### 8.176.2.5 uint8_t nas_LTEInfo::emmSubState

### 8.176.2.6 uint16_t nas_LTEInfo::RXChan

### 8.176.2.7 uint16_t nas_LTEInfo::TXChan

## 8.177 nas_LTEInfoInterfreq Struct Reference

### Data Fields

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

### 8.177.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *ueInIdle* | • TRUE if the UE is in Idle mode, otherwise FALSE.<br>  – 0xFF - Not Available |
| *freqsLen* | • Provides the number of set of inter frequency information.<br>• If 0(zero), then no information follows it. |
| *InfoInterfreq[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_infoInterFreq for more information. |

### 8.177.2 Field Documentation

#### 8.177.2.1 uint8_t nas_LTEInfoInterfreq::freqsLen

#### 8.177.2.2 nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq[255]

#### 8.177.2.3 uint8_t nas_LTEInfoInterfreq::ueInIdle

## 8.178 nas_LTEInfoIntrafreq Struct Reference

### Data Fields

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

### 8.178.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *ueInIdle* | • TRUE if the UE is in Idle mode, otherwise FALSE.<br>  – 0xFF - Not Available |
| *plmn[NAS_PLMN_LENGTH]* | • PLMN ID coded as octet 3, 4, and 5. |
| *tac* | • Tracking area code.<br>  – 0xFFFF - Not Available |
| *globalCellId* | • Global cell ID in the system information block.<br>  – 0xFFFFFFFF - Not Available |
| *earfcn* | • E-UTRA absolute radio frequency channel number of the serving cell.<br>• Range: 0 to 65535.<br>  – 0xFFFF - Not Available |
| *servingCellId* | • LTE serving cell ID.<br>• Range: 0 to 503.<br>• This is the cell ID of the serving cell and can be found in the cell list.<br>  – 0xFFFF - Not Available |
| *cellReselPriority* | • Priority for serving frequency.<br>• Range: 0 to 7.<br>• This field is only valid when ue_in_idle is TRUE.<br>  – 0xFF - Not Available |
| *sNonIntraSearch* | • S non-intra search threshold to control non-intrafrequency searches.<br>• Range: 0 to 31.<br>• This field is only valid when ue_in_idle is TRUE.<br>  – 0xFF - Not Available |
| *threshServingLow* | • Serving cell low threshold.<br>• Range: 0 to 31.<br>• This field is only valid when ue_in_idle is TRUE.<br>  – 0xFF - Not Available |

| | |
|---|---|
| *sIntraSearch* | • S Intra search threshold.<br>• Range: 0 to 31.<br>• The current cell measurement must fall below this threshold to consider intrafrequency for reselection.<br>• This field is only valid when ue_in_idle is TRUE.<br>  – 0xFF - Not Available |
| *cellsLen* | • Provides the number of set of cell params.<br>• If 0(zero), then no information follows it. |
| *CellParams[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_cellParams for more information. |

### 8.178.2 Field Documentation

#### 8.178.2.1 nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]

#### 8.178.2.2 uint8_t nas_LTEInfoIntrafreq::cellReselPriority

#### 8.178.2.3 uint8_t nas_LTEInfoIntrafreq::cellsLen

#### 8.178.2.4 uint16_t nas_LTEInfoIntrafreq::earfcn

#### 8.178.2.5 uint32_t nas_LTEInfoIntrafreq::globalCellId

#### 8.178.2.6 uint8_t nas_LTEInfoIntrafreq::plmn[3]

#### 8.178.2.7 uint16_t nas_LTEInfoIntrafreq::servingCellId

#### 8.178.2.8 uint8_t nas_LTEInfoIntrafreq::sIntraSearch

#### 8.178.2.9 uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch

#### 8.178.2.10 uint16_t nas_LTEInfoIntrafreq::tac

#### 8.178.2.11 uint8_t nas_LTEInfoIntrafreq::threshServingLow

#### 8.178.2.12 uint8_t nas_LTEInfoIntrafreq::ueInIdle

## 8.179 nas_LTEInfoNeighboringGSM Struct Reference

### Data Fields

- uint8_t ueInIdle
- uint8_t freqsLen
- nas_lteGsmCellInfo LteGsmCellInfo [255]

### 8.179.1 Detailed Description

This structure contains information about the LTE Neighboring GSM Network.

**Parameters**

| | |
|---|---|
| *ueInIdle* | • TRUE if the UE is in Idle mode, otherwise FALSE.<br>– 0xFF - Not Available |
| *freqsLen* | • Provides the number of set of LTE GSM cell information.<br>• If 0(zero), then no information follows it. |
| *LteGsmCellInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_lteGsmCellInfo for more information. |

### 8.179.2 Field Documentation

#### 8.179.2.1 uint8_t nas_LTEInfoNeighboringGSM::freqsLen

#### 8.179.2.2 nas_lteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]

#### 8.179.2.3 uint8_t nas_LTEInfoNeighboringGSM::ueInIdle

## 8.180 nas_LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

- uint8_t ueInIdle
- uint8_t freqsLen
- nas_lteWcdmaCellInfo LTEWCDMACellInfo [255]

### 8.180.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA Network.

**Parameters**

| | |
|---|---|
| *ueInIdle* | • TRUE if the UE is in Idle mode, otherwise FALSE.<br>– 0xFF - Not Available |
| *freqsLen* | • Provides the number of set of LTE WCDMA cell information.<br>• If 0(zero), then no information follows it. |
| *LTEWCDMACellInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_lteWcdmaCellInfo for more information. |

### 8.180.2 Field Documentation

#### 8.180.2.1 uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen

#### 8.180.2.2 nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

#### 8.180.2.3 uint8_t nas_LTEInfoNeighboringWCDMA::ueInIdle

## 8.181 nas_LteM1BandPrefTlv Struct Reference

### Data Fields

- uint64_t lteM1BandPref
- uint8_t TlvPresent

### 8.181.1 Detailed Description

Contain the LTE M1 band Preference.

**Parameters**

| | |
|---|---|
| *lteM1BandPref* | • Bitmask representing the LTE M1 band preference<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 22 - E-UTRA Operating Band 23<br>– Bit 23 - E-UTRA Operating Band 24<br>– Bit 24 - E-UTRA Operating Band 25<br>– Bit 25 - E-UTRA Operating Band 26<br>– Bit 27 - E-UTRA Operating Band 28<br>– Bit 28 - E-UTRA Operating Band 29<br>– Bit 29 - E-UTRA Operating Band 32<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– Bit 40 - E-UTRA Operating Band 41<br>– Bit 41 - E-UTRA Operating Band 42<br>– Bit 42 - E-UTRA Operating Band 43<br>– Bit 60 - E-UTRA Operating Band 125<br>– Bit 61 - E-UTRA Operating Band 126<br>– Bit 62 - E-UTRA Operating Band 127 |
| *TlvPresent* | • Tlv Present |

### 8.181.2 Field Documentation

#### 8.181.2.1 uint64_t nas_LteM1BandPrefTlv::lteM1BandPref

#### 8.181.2.2 uint8_t nas_LteM1BandPrefTlv::TlvPresent

## 8.182 nas_LteNb1BandPrefTlv Struct Reference

### Data Fields

- uint64_t lteNb1BandPref
- uint8_t TlvPresent

### 8.182.1 Detailed Description

Contain the LTE NB1 band Preference.

**Parameters**

| | |
|---|---|
| *lteNb1BandPref* | • Bitmask representing the LTE NB1 band preference.<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 22 - E-UTRA Operating Band 23<br>– Bit 23 - E-UTRA Operating Band 24<br>– Bit 24 - E-UTRA Operating Band 25<br>– Bit 25 - E-UTRA Operating Band 26<br>– Bit 27 - E-UTRA Operating Band 28<br>– Bit 28 - E-UTRA Operating Band 29<br>– Bit 29 - E-UTRA Operating Band 32<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– Bit 40 - E-UTRA Operating Band 41<br>– Bit 41 - E-UTRA Operating Band 42<br>– Bit 42 - E-UTRA Operating Band 43<br>– Bit 60 - E-UTRA Operating Band 125<br>– Bit 61 - E-UTRA Operating Band 126<br>– Bit 62 - E-UTRA Operating Band 127 |
| *TlvPresent* | • Tlv Present |

### 8.182.2 Field Documentation

#### 8.182.2.1 uint64_t nas_LteNb1BandPrefTlv::lteNb1BandPref

#### 8.182.2.2 uint8_t nas_LteNb1BandPrefTlv::TlvPresent

## 8.183 nas_LTEOperationMode Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t * pLTEOperationMode

### 8.183.1 Detailed Description

This structure contains the nas_LTEOperationMode response parameters.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present |
| *pLTEOperation-Mode* | • LTE Operational Mode |

### 8.183.2 Field Documentation

#### 8.183.2.1 uint32_t* nas_LTEOperationMode::pLTEOperationMode

#### 8.183.2.2 uint8_t nas_LTEOperationMode::TlvPresent

## 8.184 nas_ltePCI Struct Reference

### Data Fields

- uint8_t status
- uint32_t earfcn
- uint32_t pci

### 8.184.1 Detailed Description

This structure contains LTE PCI infromation.

**Parameters**

| | |
|---|---|
| *status* | • 0 - Disable<br>• 1 - Enable |
| *earfcn* | • UARFCN to which UE is locked |
| *pci* | • PCI to which the UE is locked |

### 8.184.2 Field Documentation

#### 8.184.2.1 uint32_t nas_ltePCI::earfcn

#### 8.184.2.2 uint32_t nas_ltePCI::pci

#### 8.184.2.3 uint8_t nas_ltePCI::status

## 8.185 nas_lteRsrpinformation Struct Reference

### Data Fields

- int16_t rsrplevel

### 8.185.1 Detailed Description

This structure contains the LTE RSRP Information

**Parameters**

| | |
|---|---|
| *rsrplevel* | • LTE RSRP in dBm as a mesaured by L1. Range: -44 to -140(-44 means -44dBm, -140 means -140dBm). |

### 8.185.2 Field Documentation

#### 8.185.2.1 int16_t nas_lteRsrpinformation::rsrplevel

## 8.186 nas_LTERSRPThresh Struct Reference

### Data Fields

- uint8_t LTERSRPThreshListLen
- int16_t * pLTERSRPThreshList

### 8.186.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

**Parameters**

| | |
|---|---|
| *LTERSRP-ThreshListLen* | • Length of the LTE RSRP threshold list parameter to follow |
| *pLTERSRP-ThreshList* | • Array of RSRP thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values<br>• Range for RSRP values: -140 to -44 (in dBm). |

### 8.186.2 Field Documentation

#### 8.186.2.1 uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen

#### 8.186.2.2 int16_t* nas_LTERSRPThresh::pLTERSRPThreshList

## 8.187 nas_LTERSRQThresh Struct Reference

### Data Fields

- uint8_t LTERSRQThreshListLen
- int16_t * pLTERSRQThreshList

### 8.187.1 Detailed Description

This structure contains LTE RSRQ threshold related parameters.

**Parameters**

| | |
|---|---|
| *LTERSRQ-ThreshListLen* | • Length of the LTE RSRQ threshold list parameter to follow |
| *pLTERSRQ-ThreshList* | • Array of RSRQ thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values.<br>• Range for RSRQ values: -20 to -3 (in dBm) |

### 8.187.2 Field Documentation

#### 8.187.2.1 uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen

#### 8.187.2.2 int16_t* nas_LTERSRQThresh::pLTERSRQThreshList

## 8.188 nas_LTERSSIThresh Struct Reference

### Data Fields

- uint8_t LTERSSIThreshListLen
- int16_t * pLTERSSIThreshList

### 8.188.1 Detailed Description

This structure contains LTE RSSI threshold related parameters.

**Parameters**

| | |
|---|---|
| *LTERSSI-ThreshListLen* | • Length of the LTE RSSI threshold list parameter to follow |
| *pLTERSSI-ThreshList* | • Array of RSSI thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values.<br>• Range for RSSI values: -120 to 0 (in dBm) |

### 8.188.2 Field Documentation

#### 8.188.2.1 uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen

#### 8.188.2.2 int16_t* nas_LTERSSIThresh::pLTERSSIThreshList

## 8.189 nas_LTESigRptCfg Struct Reference

### Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

### 8.189.1 Detailed Description

This structure contains LTE signal report config.

**Parameters**

| | |
|---|---|
| *rptRate* | • Rate on how often the LTE signal must be checked for reporting<br>• Values:<br>– 0 - Report using the default configuration<br>– 1 - Report every 1 sec<br>– 2 - Report every 2 sec<br>– 3 - Report every 3 sec<br>– 4 - Report every 4 sec<br>– 5 - Report every 5 sec |
| *avgPeriod* | • Averaging period to be used for the LTE signal<br>• Values:<br>– 0 - Average using the default configuration<br>– 1 - Average over 1 sec<br>– 2 - Average over 2 sec<br>– 3 - Average over 3 sec<br>– 4 - Average over 4 sec<br>– 5 - Average over 5 sec<br>– 6 - Average over 6 sec<br>– 7 - Average over 7 sec<br>– 8 - Average over 8 sec<br>– 9 - Average over 9 sec<br>– 10 - Average over 10 sec |

### 8.189.2 Field Documentation

#### 8.189.2.1 uint8_t nas_LTESigRptCfg::avgPeriod

#### 8.189.2.2 uint8_t nas_LTESigRptCfg::rptRate

## 8.190 nas_LTESigRptConfig Struct Reference

### Data Fields

- uint8_t rptRate
- uint8_t avgPeriod

### 8.190.1 Detailed Description

This structure contains LTE RSRP threshold related parameters.

**Parameters**

| | |
|---|---|
| *rptRate* | • Rate on how often the LTE signal must be checked for reporting Values<br>• 0 - Report using the default configuration<br>• 1 - Report every 1 sec<br>• 2 - Report every 2 sec<br>• 3 - Report every 3 sec<br>• 4 - Report every 4 sec<br>• 5 - Report every 5 sec |
| *avgPeriod* | • Averaging period to be used for the LTE signal.<br>• Values<br>  – 0 - Average using the default configuration<br>  – 1 - Average over 1 sec<br>  – 2 - Average over 2 sec<br>  – 3 - Average over 3 sec<br>  – 4 - Average over 4 sec<br>  – 5 - Average over 5 sec<br>  – 6 - Average over 6 sec<br>  – 7 - Average over 7 sec<br>  – 8 - Average over 8 sec<br>  – 9 - Average over 9 sec<br>  – 10 - Average over 10 sec |

### 8.190.2 Field Documentation

#### 8.190.2.1 uint8_t nas_LTESigRptConfig::avgPeriod

#### 8.190.2.2 uint8_t nas_LTESigRptConfig::rptRate

## 8.191 nas_lteSnrinformation Struct Reference

### Data Fields

- int16_t snrlevel

### 8.191.1 Detailed Description

This structure contains the LTE SNR Information

**Parameters**

| | |
|---|---|
| *snrlevel* | • LTE SNR level as a scaled integer in units of 0.1dB e.g. -16dB has a value of -160 and 24.6dB has value of 246. |

### 8.191.2 Field Documentation

#### 8.191.2.1 int16_t nas_lteSnrinformation::snrlevel

## 8.192 nas_LTESNRThresh Struct Reference

### Data Fields

- uint8_t LTESNRThresListLen
- int16_t ∗ pLTESNRThresList

### 8.192.1 Detailed Description

This structure contains LTE SNR threshold list.

**Parameters**

| | |
|---|---|
| *LTESNRThres-ListLen* | • Length of the LTE SNR threshold list parameter to follow |
| *pLTESNRThres-List* | • Sequence of thresholds delimiting SNR event reporting bands<br>• Every time a SNR value crosses a threshold value, an event report indication message with the new SNR value is sent to the requesting control point. For this field<br>– For LTE, each SNR threshold value is a signed 2 Byte value<br>– Maximum number of threshold values is 16<br>– At least one value must be specified<br>– SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246 |

### 8.192.2 Field Documentation

#### 8.192.2.1 uint8_t nas_LTESNRThresh::LTESNRThresListLen

#### 8.192.2.2 int16_t∗ nas_LTESNRThresh::pLTESNRThresList

## 8.193 nas_LTESNRThreshold Struct Reference

### Data Fields

- uint8_t LTESNRThreshListLen

- int16_t * pLTESNRThreshList

### 8.193.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

**Parameters**

| | |
|---|---|
| *LTESNRThresh-ListLen* | • Length of the LTE SNR threshold list parameter to follow |
| *pLTESNR-ThreshList* | • Array of SNR thresholds (in units of 0.1 dB)<br>• Maximum of 32 values<br>• Range for SNR values: -20 to 30 (in dB). |

### 8.193.2 Field Documentation

#### 8.193.2.1 uint8_t nas_LTESNRThreshold::LTESNRThreshListLen

#### 8.193.2.2 int16_t* nas_LTESNRThreshold::pLTESNRThreshList

## 8.194 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

### 8.194.1 Detailed Description

Structure for storing the LTE System Information.

**Parameters**

| | |
|---|---|
| *sysInfoLTE* | • See nas_sysInfoCommon for more information. |

| | |
|---|---|
| *lacValid* | • Indicates whether the location area code is valid..<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *lac* | • Location area code.<br>• Only applies to 3GPP.<br>– 0xFFFF - Not Available |
| *cellIdValid* | • Indicates whether the cell ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *cellId* | • Cell ID.<br>– 0xFFFFFFFF - Not Available |
| *regRejectInfo-Valid* | • Indicates whether the registration reject information is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *rejectSrvDomain* | • Type of service domain in which the registration is rejected.<br>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>– 0x04 - Camped<br>– 0xFF - Not Available |
| *rejCause* | • Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6]. and [3GPP TS 24.301, Section 9.9.3.9]<br>– 0xFF - Not Available |
| *networkIdValid* | • Indicates whether the network ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *MCC[NAS_PLM-N_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters |

| | |
|---|---|
| *MNC[NAS_PLMN_LENGTH]* | • Mobile Network Code.<br>• MNC digits in ASCII characters<br>• An unused byte is set to 0xFF.<br>• 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. |
| *tacValid* | • Indicates whether tracking area code is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *tac* | • Tracking area code.<br>• Only applicable for LTE.<br>– 0xFFFF - Not Available |

### 8.194.2 Field Documentation

#### 8.194.2.1 uint32_t nas_LTESysInfo::cellId

#### 8.194.2.2 uint8_t nas_LTESysInfo::cellIdValid

#### 8.194.2.3 uint16_t nas_LTESysInfo::lac

#### 8.194.2.4 uint8_t nas_LTESysInfo::lacValid

#### 8.194.2.5 uint8_t nas_LTESysInfo::MCC[3]

#### 8.194.2.6 uint8_t nas_LTESysInfo::MNC[3]

#### 8.194.2.7 uint8_t nas_LTESysInfo::networkIdValid

#### 8.194.2.8 uint8_t nas_LTESysInfo::regRejectInfoValid

#### 8.194.2.9 uint8_t nas_LTESysInfo::rejCause

#### 8.194.2.10 uint8_t nas_LTESysInfo::rejectSrvDomain

#### 8.194.2.11 nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE

#### 8.194.2.12 uint16_t nas_LTESysInfo::tac

#### 8.194.2.13 uint8_t nas_LTESysInfo::tacValid

## 8.195 nas_IteWcdmaCellInfo Struct Reference

### Data Fields

- uint16_t uarfcn
- uint8_t cellReselPriority

- uint16_t threshXhigh
- uint16_t threshXlow
- uint8_t cellsLen
- nas_wcdmaCellInfo WCDMACellInfo [255]

### 8.195.1 Detailed Description

This structure contains information about the LTE WCDMA Cell.

**Parameters**

| | |
|---|---|
| *uarfcn* | • WCDMA layer frequency.<br>• Range: 0 to 16383. |
| *cellReselPriority* | • Cell re-selection priority.<br>• Range: 0 to 7.<br>• This field is only valid when ue_in_idle is TRUE. |
| *threshXhigh* | • Re-selection low threshold.<br>• Range: 0 to 31.<br>• This field is only valid when ue_in_idle is TRUE. |
| *threshXlow* | • Re-selection high threshold.<br>• Range: 0 to 31.<br>• This field is only valid when ue_in_idle is TRUE. |
| *cellsLen* | • Provides the number of set of WCDMA cells. |
| *WCDMACellInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_wcdmaCellInfo for more information. |

### 8.195.2 Field Documentation

#### 8.195.2.1 uint8_t nas_lteWcdmaCellInfo::cellReselPriority

#### 8.195.2.2 uint8_t nas_lteWcdmaCellInfo::cellsLen

#### 8.195.2.3 uint16_t nas_lteWcdmaCellInfo::threshXhigh

#### 8.195.2.4 uint16_t nas_lteWcdmaCellInfo::threshXlow

#### 8.195.2.5 uint16_t nas_lteWcdmaCellInfo::uarfcn

#### 8.195.2.6 nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]

## 8.196 nas_minBasedIMSI Struct Reference

### Data Fields

- uint8_t mccM [3]
- uint16_t imsiM1112
- uint8_t imsiMS1 [7]
- uint8_t imsiMS2 [3]

### 8.196.1 Detailed Description

This structure contains MIN-based IMSI.

**Parameters**

| | |
|---|---|
| *mccM* | • ASCII character representation of MCC_M |
| *imsiM1112* | • ASCII character representation of IMSI_M_11_12 value<br>– 0xFFFF - Not Available |
| *imsiMS1* | • ASCII character representation of IMSI_M_S1 value |
| *imsiMS2* | • ASCII character representation of IMSI_M_S2 value |

### 8.196.2 Field Documentation

#### 8.196.2.1 uint16_t nas_minBasedIMSI::imsiM1112

#### 8.196.2.2 uint8_t nas_minBasedIMSI::imsiMS1[7]

#### 8.196.2.3 uint8_t nas_minBasedIMSI::imsiMS2[3]

#### 8.196.2.4 uint8_t nas_minBasedIMSI::mccM[3]

## 8.197 nas_MNRInfo Struct Reference

### Data Fields

- uint16_t mcc
- uint16_t mnc
- uint32_t rat

### 8.197.1 Detailed Description

Structure contains Manual Network Register Information parameters

**Parameters**

| | |
|---|---|
| *mcc* | • A 16-bit integer representation of Mobile Country Code. Range - 0 to 999. |

| | |
|---|---|
| *mnc* | • A 16-bit integer representation of Mobile Network Code. Range - 0 to 999. |
| *rat* | • Radio access technology for which to register.<br>– 0x04 - RADIO_IF_GSM<br>– 0x05 - RADIO_IF_UMTS<br>– 0x08 - RADIO_IF_LTE |

### 8.197.2 Field Documentation

#### 8.197.2.1 uint16_t nas_MNRInfo::mcc

#### 8.197.2.2 uint16_t nas_MNRInfo::mnc

#### 8.197.2.3 uint32_t nas_MNRInfo::rat

## 8.198 nas_ModePrefTlv Struct Reference

### Data Fields

- uint16_t ModePref
- uint8_t TlvPresent

### 8.198.1 Detailed Description

Contain the radio technology mode preference

**Parameters**

| | |
|---|---|
| *ModePref* | • Bit Mask indicating the radio technology mode preference<br>• Bit values:<br>– Bit 0 - cdma2000 1x<br>– Bit 1 - cdma2000 HRPD(1xEV-DO)<br>– Bit 2 - GSM<br>– Bit 3 - UMTS<br>– Bit 4 - LTE<br>– Bit 5 - TD-SCDMA<br>– Bit 6 - NR5G |
| *TlvPresent* | • Tlv Present |

### 8.198.2 Field Documentation

#### 8.198.2.1 uint16_t nas_ModePrefTlv::ModePref

#### 8.198.2.2 uint8_t nas_ModePrefTlv::TlvPresent

## 8.199 nas_namName Struct Reference

### Data Fields

- uint8_t namNameLen
- uint8_t namName [12]

### 8.199.1 Detailed Description

This structure contains NAM Name.

**Parameters**

| | |
|---|---|
| *namNameLen* | • Number of sets of the following elements:<br>– nam_name<br>• If zero(0), then no information follows. |
| *namName* | • Name information in ASCII. The maximum length of nam_name is 12. |

### 8.199.2 Field Documentation

#### 8.199.2.1 uint8_t nas_namName::namName[12]

#### 8.199.2.2 uint8_t nas_namName::namNameLen

## 8.200 nas_netSelectionPref Struct Reference

### Data Fields

- uint8_t netReg
- uint16_t mcc
- uint16_t mnc

### 8.200.1 Detailed Description

Contain the network selection preference.

**Parameters**

| | |
|---|---|
| *netReg* | • specifies one of the following actions:<br>– 0x00 - Automatic registration<br>* Device registers according to its provisioning; mcc and mnc fields are ignored<br>– 0x01 - Manual Registration<br>* Device registers to specified network; mcc and mnc must contain valid values |

| | |
|---|---|
| *mcc* | • MCC value. Range 0 to 999 |
| *mnc* | • MNC value. Range 0 to 999 |

### 8.200.2 Field Documentation

#### 8.200.2.1 uint16_t nas_netSelectionPref::mcc

#### 8.200.2.2 uint16_t nas_netSelectionPref::mnc

#### 8.200.2.3 uint8_t nas_netSelectionPref::netReg

## 8.201 nas_NetSelPrefTlv Struct Reference

### Data Fields

- uint8_t NetSelPref
- uint8_t TlvPresent

### 8.201.1 Detailed Description

Contain the Network selection Preference for system selection preferences.

**Parameters**

| | |
|---|---|
| *NetSelPref* | • Parameter indicating network selection preference<br>• Values:<br>– 0x00 - Automatic network selection<br>– 0x01 - Manual network selection |
| *TlvPresent* | • Tlv Present |

### 8.201.2 Field Documentation

#### 8.201.2.1 uint8_t nas_NetSelPrefTlv::NetSelPref

#### 8.201.2.2 uint8_t nas_NetSelPrefTlv::TlvPresent

## 8.202 nas_NetworkStat1x Struct Reference

### Data Fields

- uint8_t State
- uint16_t SO
- uint32_t RX_PWR

- uint16_t RX_EC_IO
- uint32_t TX_PWR
- uint8_t ActSetCnt
- nas_ActPilotPNElement * pActPilotPNElements
- uint8_t NeighborSetCnt
- uint16_t * pNeighborSetPilotPN

### 8.202.1 Detailed Description

This structure contains network stat 1x.

**Parameters**

| | |
|---|---|
| *State* | • CDMA current access state<br>– 0x01 - Searching<br>– 0x02 - Idle<br>– 0x03 - Traffic<br>– Others - NA |
| *SO* | • CDMA service option<br>– 0xFFFF - Not in a call<br>– 0x0001 - Basic Variable Rate Voice Service(8kbps)<br>– 0x0002 - Mobile Station Loopback(8kbps)<br>– 0x0003 - Enhanced Variable Rate Codec(EVRC) Voice Service(8kbps)<br>– 0x0006 - Short message Services(Rate Set 1)<br>– 0x0009 - Mobile Station Loopback(13kbps)<br>– 0x000E - Short Message Service (Rate Set 2)<br>– 0x0011 - High Rate Voice Service(13kbps)<br>– 0x0020 - Test Data Service Option(TDSO)<br>– 0x0021 - cdma2000 High Speed Packet Data Service, Internet or ISO Protocol Stack<br>– 0x0044 - EVRC-B Voice Service(8 kbps)<br>– 0x0046 - EVRC-WB Voice Service(8 kbps)<br>– 0x0049 - Voice Echo mode supports smart blanking(EVRC-NW)<br>– 0x004B - Enhanced loopback<br>– 0x8000 - Proprietary Service Option (Qualcomm Inc.) |
| *RX_PWR* | • RX Pwr(dBm)<br>– 0xABCD00EF - -ABCD.EF dBm<br>– ABCD00EF should be transferred to decimal while displaying<br>– Example: 0x12340056 - -4660.86dBm<br>0x1234 = 4660, 0x0056 = 86<br>– 0xFFFFFFFF - NA |

| | |
|---|---|
| *RX_EC_IO* | • RX EC/IO(dB)<br>– 0xABCD - -AB.CD dB<br>– ABCD should be transferred to decimal while displaying<br>– Example: 0x1234 - -18.52dB<br>0x12 = 18, 0x34 = 52<br>– 0xFFFF - NA |
| *TX_PWR* | • TX PWR(dBm)<br>– 0xFFFFFFFF - NA<br>– Others - display actual value in decimal<br>– Example: 0x1234 - -4660dBm<br>0x1234 = 4660 |
| *ActSetCnt* | • Count of active pilot PN elements<br>• As input specifies number of sets of parameter pActPilotElements for which memory has been assigned<br>• As output specifies the actual number of sets of parameter pActPilotElements returned by device |
| *pActPilotPN-Elements* | • See nas_ActPilotPNElement for more information |
| *NeighborSetCnt* | • Count of neighbor pilot PN elements<br>• As input specifies number of sets of parameter pNeighborSetPilotPN for which memory has been assigned<br>• As output specifies the actual number of sets of parameter pNeighborSetPilotPN returned by device |
| *pNeighborSet-PilotPN* | • Neighbor pilot PN |

## 8.202.2 Field Documentation

### 8.202.2.1 uint8_t nas_NetworkStat1x::ActSetCnt

### 8.202.2.2 uint8_t nas_NetworkStat1x::NeighborSetCnt

### 8.202.2.3 nas_ActPilotPNElement* nas_NetworkStat1x::pActPilotPNElements

### 8.202.2.4 uint16_t* nas_NetworkStat1x::pNeighborSetPilotPN

### 8.202.2.5 uint16_t nas_NetworkStat1x::RX_EC_IO

### 8.202.2.6 uint32_t nas_NetworkStat1x::RX_PWR

### 8.202.2.7 uint16_t nas_NetworkStat1x::SO

### 8.202.2.8 uint8_t nas_NetworkStat1x::State

#### 8.202.2.9 uint32_t nas_NetworkStat1x::TX_PWR

## 8.203 nas_NetworkStatEVDO Struct Reference

### Data Fields

- uint8_t State
- uint8_t MACIndex
- uint8_t SectorIDLen
- uint16_t * pSectorID
- uint16_t RX_PWR
- uint16_t PER
- uint16_t PilotEnergy
- uint8_t SNR

### 8.203.1 Detailed Description

This structure contains network stat EVDO.

**Parameters**

| | |
|---|---|
| *State* | • EVDO network access state<br>– 0x00 - Sleep<br>– 0x01 - Searching<br>– 0x02 - Idle<br>– 0x03 - Active<br>– 0xFF - NA |
| *MACIndex* | • HDR Mac index<br>– 0xFF - NA<br>– Others - Display the actual value in decimal<br>– Example: 0x12 - 18<br>0x12 = 18 |
| *SectorIDLen* | (IN/OUT)<br>• Sector ID length<br>• As input specifies length of parameter pSectorID for which memory has been assigned<br>• As output specifies the actual length of parameter pSectorID returned by device |
| *pSectorID* | • Sector ID |
| *RX_PWR* | • TX PWR(dBm)<br>– 0xABCD - -ABCD dBm<br>– ABCD should be transferred to decimal while displaying<br>– Example: 0x1234 - -4660dBm<br>0x1234 = 4660<br>– 0xFFFF - NA |

| | |
|---|---|
| *PER* | • HDR Packet Error Rate<br>– 0xFFFF - Unknown<br>– Others - display the actual value in decimal<br>– Example: 0x1234 - -4660dBm<br>0x1234 = 4660 |
| *PilotEnergy* | • Pilt Energy (dB)<br>– 0xFFFF - NA<br>– 0xABCD should be transferred to decimal while displaying<br>– Example: 0x1234 - -4660dBm<br>0x1234 = 4660 |
| *SNR* | • Signal to Noise ratio (dB) |

### 8.203.2 Field Documentation

#### 8.203.2.1 uint8_t nas_NetworkStatEVDO::MACIndex

#### 8.203.2.2 uint16_t nas_NetworkStatEVDO::PER

#### 8.203.2.3 uint16_t nas_NetworkStatEVDO::PilotEnergy

#### 8.203.2.4 uint16_t* nas_NetworkStatEVDO::pSectorID

#### 8.203.2.5 uint16_t nas_NetworkStatEVDO::RX_PWR

#### 8.203.2.6 uint8_t nas_NetworkStatEVDO::SectorIDLen

#### 8.203.2.7 uint8_t nas_NetworkStatEVDO::SNR

#### 8.203.2.8 uint8_t nas_NetworkStatEVDO::State

## 8.204 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

### 8.204.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *nmrCellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present. |
| *nmrPlmn[NAS_-PLMN_LENGT-H]* | • MCC/MNC information coded as octet 3, 4, and 5.<br>• This field is ignored when nmrCellID is not present. |
| *nmrLac* | • Location area code.<br>• This field is ignored when nmrCellID is not present.<br>– 0xFFFF - Not Available |
| *nmrArfcn* | • Absolute RF channel number.<br>– 0xFFFF - Not Available |
| *nmrBsic* | • Base station identity code.<br>– 0xFF - Not Available |
| *nmrRxLev* | • Cell Rx measurement.<br>• Values range between 0 and 63.<br>• Mapped to a measured signal level:<br>– Rxlev 0 is a signal strength less than -110 dBm<br>– Rxlev 1 is -110 dBm to -109 dBm<br>– Rxlev 2 is -109 dBm to -108 dBm<br>– ...<br>– Rxlev 62 is -49 dBm to -48 dBm<br>– Rxlev 63 is greater than -48 dBm<br>– 0xFFFF - Not Available |

### 8.204.2 Field Documentation

#### 8.204.2.1 uint16_t nas_nmrCellInfo::nmrArfcn

#### 8.204.2.2 uint8_t nas_nmrCellInfo::nmrBsic

#### 8.204.2.3 uint32_t nas_nmrCellInfo::nmrCellID

#### 8.204.2.4 uint16_t nas_nmrCellInfo::nmrLac

#### 8.204.2.5 uint8_t nas_nmrCellInfo::nmrPlmn[3]

#### 8.204.2.6 uint16_t nas_nmrCellInfo::nmrRxLev

## 8.205 nas_nr5gBandPref Struct Reference

### Data Fields

- uint64_t bits_1_64
- uint64_t bits_65_128
- uint64_t bits_129_192
- uint64_t bits_193_256

### 8.205.1 Detailed Description

Contain the NR5G Band Preference.

**Parameters**

| | |
|---|---|
| *bits_1_64* | • Bits 1 to 64 of the 256-bit NR5G Operating Band bitmask. |
| *bits_65_128* | • Bits 65 to 128 of the 256-bit NR5G Operating Band bitmask. |
| *bits_129_192* | • Bits 129 to 192 of the 256-bit NR5G Operating Band bitmask. |
| *bits_193_256* | • Bits 193 to 256 of the 256-bit NR5G Operating Band bitmask. |

### 8.205.2 Field Documentation

#### 8.205.2.1 uint64_t nas_nr5gBandPref::bits_129_192

#### 8.205.2.2 uint64_t nas_nr5gBandPref::bits_193_256

#### 8.205.2.3 uint64_t nas_nr5gBandPref::bits_1_64

#### 8.205.2.4 uint64_t nas_nr5gBandPref::bits_65_128

## 8.206 nas_NR5gBandPrefTlv Struct Reference

### Data Fields

- uint64_t bits_1_64
- uint64_t bits_65_128
- uint64_t bits_129_192
- uint64_t bits_193_256
- uint8_t TlvPresent

### 8.206.1 Detailed Description

Contain the NR5G Band Preference.

**Parameters**

| | |
|---|---|
| *bits_1_64* | • Bits 1 to 64 of the 256-bit NR5G Operating Band bitmask. |
| *bits_65_128* | • Bits 65 to 128 of the 256-bit NR5G Operating Band bitmask. |
| *bits_129_192* | • Bits 129 to 192 of the 256-bit NR5G Operating Band bitmask. |
| *bits_193_256* | • Bits 193 to 256 of the 256-bit NR5G Operating Band bitmask. |
| *TlvPresent* | • Tlv Present |

### 8.206.2 Field Documentation

#### 8.206.2.1 uint64_t nas_NR5gBandPrefTlv::bits_129_192

#### 8.206.2.2 uint64_t nas_NR5gBandPrefTlv::bits_193_256

#### 8.206.2.3 uint64_t nas_NR5gBandPrefTlv::bits_1_64

#### 8.206.2.4 uint64_t nas_NR5gBandPrefTlv::bits_65_128

#### 8.206.2.5 uint8_t nas_NR5gBandPrefTlv::TlvPresent

## 8.207 nas_operatorNameString Struct Reference

### Data Fields

- uint8_t PLMNName [255]

### 8.207.1 Detailed Description

This structure contains PLMN name.

**Parameters**

| | |
|---|---|
| *PLMNName* | • PLMN name must be coded in a default 7-bit alphabet with b8 set to 0. |

### 8.207.2 Field Documentation

#### 8.207.2.1 uint8_t nas_operatorNameString::PLMNName[255]

## 8.208 nas_OperatorPLMNData Struct Reference

### Data Fields

- uint8_t mcc [3]
- uint8_t mnc [3]
- uint16_t lac1
- uint16_t lac2
- uint8_t PLMNRecID

### 8.208.1 Detailed Description

This structure contains Operator PLMN.

**Parameters**

| | |
|---|---|
| *mcc* | • MCC in ASCII string (a value of D in any of the digits is to be used to indicate a "wild" value for that corresponding digit). |
| *mnc* | • MNC in ASCII string (a value of D in any of the digits is to be used to indicate a "wild" value for that corresponding digit; digit 3 in MNC is optional and when not present, will be set as ASCII F). |
| *lac1* | • Location area code 1. |
| *lac2* | • Location area code 1. |
| *PLMNRecID* | • PLMN network name record identifier. |

### 8.208.2 Field Documentation

#### 8.208.2.1 uint16_t nas_OperatorPLMNData::lac1

#### 8.208.2.2 uint16_t nas_OperatorPLMNData::lac2

#### 8.208.2.3 uint8_t nas_OperatorPLMNData::mcc[3]

#### 8.208.2.4 uint8_t nas_OperatorPLMNData::mnc[3]

#### 8.208.2.5 uint8_t nas_OperatorPLMNData::PLMNRecID

## 8.209 nas_operatorPLMNList Struct Reference

### Data Fields

- uint16_t numInstance
- nas_OperatorPLMNData PLMNData [255]

### 8.209.1 Detailed Description

This structure contains Operator PLMN List.

**Parameters**

| | |
|---|---|
| *numInstance* | • Number of sets of the elements. |
| *PLMNData* | • Refer OperatorPLMNLData for details (Optional). |

### 8.209.2 Field Documentation

#### 8.209.2.1 uint16_t nas_operatorPLMNList::numInstance

#### 8.209.2.2 nas_OperatorPLMNData nas_operatorPLMNList::PLMNData[255]

## 8.210 nas_PhyCaAggPcellInfo Struct Reference

### Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- int iLTEbandValue
- uint8_t TlvPresent

### 8.210.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *freq* | • Frequency of the absolute cell Range.<br>• Range for ID values: 0 to 65535. |
| *dl_bw_value* | • Downlink Bandwidth Values.<br>• See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information. |
| *scell_state* | • Scell state Values.<br>• See NAS_LTE_CPHY_SCELL_STATE_LITE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.210.2 Field Documentation

#### 8.210.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggPcellInfo::dl_bw_value

#### 8.210.2.2 int nas_PhyCaAggPcellInfo::freq

#### 8.210.2.3 int nas_PhyCaAggPcellInfo::iLTEbandValue

#### 8.210.2.4 int nas_PhyCaAggPcellInfo::pci

#### 8.210.2.5 uint8_t nas_PhyCaAggPcellInfo::TlvPresent

## 8.211 nas_PhyCaAggScellDlBw Struct Reference

### Data Fields

- NAS_LTE_CPHY_CA_BW_NRB_LITE dl_bw_value
- uint8_t TlvPresent

### 8.211.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *dl_bw_value* | • Downlink Bandwidth Values.<br>• See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information. |

### 8.211.2 Field Documentation

#### 8.211.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellDlBw::dl_bw_value

#### 8.211.2.2 uint8_t nas_PhyCaAggScellDlBw::TlvPresent

## 8.212 nas_PhyCaAggScellIndex Struct Reference

### Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

### 8.212.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *scell_idx* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |

### 8.212.2 Field Documentation

#### 8.212.2.1 uint8_t nas_PhyCaAggScellIndex::scell_idx

#### 8.212.2.2 uint8_t nas_PhyCaAggScellIndex::TlvPresent

## 8.213 nas_PhyCaAggScellIndType Struct Reference

### Data Fields

- int pci
- int freq
- NAS_LTE_CPHY_SCELL_STATE_LITE scell_state
- uint8_t TlvPresent

### 8.213.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *freq* | • Frequency of the absolute cell Range.<br>• Range for ID values: 0 to 65535. |
| *scell_state* | • Scell state Values.<br>• See NAS_LTE_CPHY_SCELL_STATE_LITE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.213.2 Field Documentation

#### 8.213.2.1 int nas_PhyCaAggScellIndType::freq

#### 8.213.2.2 int nas_PhyCaAggScellIndType::pci

#### 8.213.2.3 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellIndType::scell_state

#### 8.213.2.4 uint8_t nas_PhyCaAggScellIndType::TlvPresent

## 8.214 nas_PhyCaAggScellInfo Struct Reference

### Data Fields

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

### 8.214.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *freq* | • Frequency of the absolute cell Range.<br>• Range for ID values: 0 to 65535. |
| *dl_bw_value* | • Downlink Bandwidth Values.<br>• See NAS_LTE_CPHY_CA_BW_NRB_LITE for more information. |

| | |
|---|---|
| *iLTEbandValue* | • Band value.<br>• Range for LTE Band class 120 to 160.<br>– 120 - LTE E-UTRA Operating Band 1<br>– 121 - LTE E-UTRA Operating Band 2<br>– 122 - LTE E-UTRA Operating Band 3<br>– 123 - LTE E-UTRA Operating Band 4<br>– 124 - LTE E-UTRA Operating Band 5<br>– 125 - LTE E-UTRA Operating Band 6<br>– 126 - LTE E-UTRA Operating Band 7<br>– 127 - LTE E-UTRA Operating Band 8<br>– 128 - LTE E-UTRA Operating Band 9<br>– 129 - LTE E-UTRA Operating Band 10<br>– 130 - LTE E-UTRA Operating Band 11<br>– 131 - LTE E-UTRA Operating Band 12<br>– 132 - LTE E-UTRA Operating Band 13<br>– 133 - LTE E-UTRA Operating Band 14<br>– 134 - LTE E-UTRA Operating Band 17<br>– 135 - LTE E-UTRA Operating Band 33<br>– 136 - LTE E-UTRA Operating Band 34<br>– 137 - LTE E-UTRA Operating Band 35<br>– 138 - LTE E-UTRA Operating Band 36<br>– 139 - LTE E-UTRA Operating Band 37<br>– 140 - LTE E-UTRA Operating Band 38<br>– 141 - LTE E-UTRA Operating Band 39<br>– 142 - LTE E-UTRA Operating Band 40<br>– 143 - LTE E-UTRA Operating Band 18<br>– 144 - LTE E-UTRA Operating Band 19<br>– 145 - LTE E-UTRA Operating Band 20<br>– 146 - LTE E-UTRA Operating Band 21<br>– 147 - LTE E-UTRA Operating Band 24<br>– 148 - LTE E-UTRA Operating Band 25<br>– 149 - LTE E-UTRA Operating Band 41<br>– 150 - LTE E-UTRA Operating Band 42<br>– 151 - LTE E-UTRA Operating Band 43<br>– 152 - LTE E-UTRA Operating Band 23<br>– 153 - LTE E-UTRA Operating Band 26<br>– 154 - LTE E-UTRA Operating Band 32<br>– 155 - LTE E-UTRA Operating Band 125<br>– 156 - LTE E-UTRA Operating Band 126<br>– 157 - LTE E-UTRA Operating Band 127<br>– 158 - LTE E-UTRA Operating Band 28<br>– 159 - LTE E-UTRA Operating Band 29<br>– 160 - LTE E-UTRA Operating Band 30 |
| *scell_state* | • Scell state Values.<br>• See NAS_LTE_CPHY_SCELL_STATE_LITE for more information. |

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |

### 8.214.2 Field Documentation

#### 8.214.2.1 NAS_LTE_CPHY_CA_BW_NRB_LITE nas_PhyCaAggScellInfo::dl_bw_value

#### 8.214.2.2 int nas_PhyCaAggScellInfo::freq

#### 8.214.2.3 uint16_t nas_PhyCaAggScellInfo::iLTEbandValue

#### 8.214.2.4 int nas_PhyCaAggScellInfo::pci

#### 8.214.2.5 NAS_LTE_CPHY_SCELL_STATE_LITE nas_PhyCaAggScellInfo::scell_state

#### 8.214.2.6 uint8_t nas_PhyCaAggScellInfo::TlvPresent

## 8.215 nas_PilotSetData Struct Reference

### Data Fields

- uint8_t NumPilots
- nas_PilotSetParams * pPilotSetInfo

### 8.215.1 Detailed Description

This structure contains pilot sets data.

**Parameters**

| | |
|---|---|
| *NumPilots* | • Number of Pilot Sets<br>• As input specifies number of sets of parameter pPilotSetInfo for which memory has been assigned<br>• As output specifies the actual number of sets of parameter pPilotSetInfo returned by device |
| *pPilotSetInfo* | • Pilot Set Parameters<br>• See nas_PilotSetParams for more information. |

note A buffer under sized error is returned if the number of sets of pPilotSetInfo returned by the device is greater than the value in NumPilots input parameter.

### 8.215.2 Field Documentation

#### 8.215.2.1 uint8_t nas_PilotSetData::NumPilots

#### 8.215.2.2 nas_PilotSetParams* nas_PilotSetData::pPilotSetInfo

## 8.216 nas_PilotSetParams Struct Reference

### Data Fields

- uint32_t PilotType
- uint16_t PilotPN
- uint16_t PilotStrength

### 8.216.1 Detailed Description

This structure contains pilot sets parameters.

**Parameters**

| | |
|---|---|
| *PilotType* | • 0x00 - NAS_HRPD_PILOT_CURR_ACT_PLT Current Active Pilot<br>• 0x01 - NAS_HRPD_PILOT_NEIGHBOR_PLT Neighbor pilot information |
| *PilotPN* | • Pilot PN sequence offset index |
| *PilotStrength* | • Strength of the pilot (in dB) |

### 8.216.2 Field Documentation

#### 8.216.2.1 uint16_t nas_PilotSetParams::PilotPN

#### 8.216.2.2 uint16_t nas_PilotSetParams::PilotStrength

#### 8.216.2.3 uint32_t nas_PilotSetParams::PilotType

## 8.217 nas_PLMNNetworkName Struct Reference

### Data Fields

- uint8_t numInstance
- nas_PLMNNetworkNameData PLMNNetName [255]

### 8.217.1 Detailed Description

This structure contains NITZ Information list.

**Parameters**

| | |
|---|---|
| *numInstance* | • Number of sets of the elements. |
| *PLMNNetName* | • Refer nas_PLMNNetworkNameData for details (Optional). |

### 8.217.2 Field Documentation

#### 8.217.2.1 uint8_t nas_PLMNNetworkName::numInstance

#### 8.217.2.2 nas_PLMNNetworkNameData nas_PLMNNetworkName::PLMNNetName[255]

## 8.218 nas_PLMNNetworkNameData Struct Reference

### Data Fields

- uint8_t codingScheme
- uint8_t countryInitials
- uint8_t longNameSpareBits
- uint8_t shortNameSpareBits
- uint8_t longNameLen
- uint8_t longName [255]
- uint8_t shortNameLen
- uint8_t shortName [255]

### 8.218.1 Detailed Description

This structure contains NITZ Information.

**Parameters**

| | |
|---|---|
| *codingScheme* | • Coding scheme:<br>– 0 - CODING_SCHEME_CELL_BROADCAST_GSM - Cell broadcast data coding scheme, GSM default alphabet, language unspecified;defined in 3GPP TS 23.038.<br>– 1 - CODING_SCHEME_UCS2 - UCS2 (16 bit);defined in ISO/IEC 10646 |
| *countryInitials* | • Country's initials:<br>– 0 - COUNTRY_INITIALS_DO_NOT_ADD - MS should not add the letters for the country's initials to the text string.<br>– 1 - COUNTRY_INITIALS_ADD - MS should add the letters for the country's initials and a separator, e.g., a space, to the text string. |
| *longNameSpare-Bits* | • Long Name Spare Bits:<br>– 1 - SPARE_BITS_8 - Bit 8 is spare and set to 0 in octet n<br>– 2 - SPARE_BITS_7_TO_8 - Bits 7 and 8 are spare and set to 0 in octet n.<br>– 3 - SPARE_BITS_6_TO_8 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 4 - SPARE_BITS_5_TO_8 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 5 - SPARE_BITS_4_TO_8 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 6 - SPARE_BITS_3_TO_8 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 7 - SPARE_BITS_2_TO_8 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 0 - SPARE_BITS_UNKNOWN - Carries no information about the number of spare bits in octet n. |

| | |
|---|---|
| *shortName-SpareBits* | • Short Name Spare Bits:<br>– 1 - SPARE_BITS_8 - Bit 8 is spare and set to 0 in octet n.<br>– 2 - SPARE_BITS_7_TO_8 - Bits 7 and 8 are spare and set to 0 in octet n.<br>– 3 - SPARE_BITS_6_TO_8 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 4 - SPARE_BITS_5_TO_8 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 5 - SPARE_BITS_4_TO_8 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 6 - SPARE_BITS_3_TO_8 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 7 - SPARE_BITS_2_TO_8 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n.<br>– 0 - SPARE_BITS_UNKNOWN - Carries no information about the number of spare bits in octet n. |
| *longNameLen* | • It provides the length of long name. |
| *longName* | • Long name string in coding_scheme. |
| *shortNameLen* | • It provides the length of short name. |
| *shortName* | • Short name string in coding_scheme. |

### 8.218.2 Field Documentation

#### 8.218.2.1 uint8_t nas_PLMNNetworkNameData::codingScheme

#### 8.218.2.2 uint8_t nas_PLMNNetworkNameData::countryInitials

#### 8.218.2.3 uint8_t nas_PLMNNetworkNameData::longName[255]

#### 8.218.2.4 uint8_t nas_PLMNNetworkNameData::longNameLen

#### 8.218.2.5 uint8_t nas_PLMNNetworkNameData::longNameSpareBits

#### 8.218.2.6 uint8_t nas_PLMNNetworkNameData::shortName[255]

#### 8.218.2.7 uint8_t nas_PLMNNetworkNameData::shortNameLen

#### 8.218.2.8 uint8_t nas_PLMNNetworkNameData::shortNameSpareBits

## 8.219 nas_PRLPrefTlv Struct Reference

### Data Fields

- uint16_t PRLPref
- uint8_t TlvPresent

### 8.219.1 Detailed Description

Contain the PRL preference for system selection preferences.

**Parameters**

| | |
|---|---|
| *PRLPref* | • Parameter indicating the CDMA PRL Preference<br>• Values:<br>– 0x0001 - Acquire available system only on the A side<br>– 0x0002 - Acquire available system only on the B side<br>– 0x3FFF - Acquire any available systems |
| *TlvPresent* | • Tlv Present |

### 8.219.2 Field Documentation

#### 8.219.2.1 uint16_t nas_PRLPrefTlv::PRLPref

#### 8.219.2.2 uint8_t nas_PRLPrefTlv::TlvPresent

## 8.220 nas_protocolSubtypeElement Struct Reference

### Data Fields

- uint16_t PhysicalLayer
- uint16_t ControlMac
- uint16_t AccessMac
- uint16_t ForwardMac
- uint16_t ReverseMac
- uint16_t KeyExchange
- uint16_t AuthProt
- uint16_t EncryptProt
- uint16_t SecProt
- uint16_t IdleState
- uint16_t MultDisc
- uint16_t VirtStream

### 8.220.1 Detailed Description

This structure contains protocol subtype element.

**Parameters**

| | |
|---|---|
| *PhysicalLayer* | • Specifies Physical Layer Protocol subtype |
| *ControlMac* | • Specifies Control Channel MAC Protocol subtype |
| *AccessMac* | • Specifies Access Channel MAC Protocol subtype |

| | |
|---|---|
| *ForwardMac* | • Specifies Forward Traffic Channel MAC Protocol subtype |
| *ReverseMac* | • Specifies Reverse Traffic Channel MAC Protocol subtype |
| *KeyExchange* | • Specifies Key exchange Protocol subtype |
| *AuthProt* | • Specifies Authentication Protocol subtype |
| *EncryptProt* | • Specifies Encryption Protocol subtype |
| *SecProt* | • Specifies Security Protocol subtype |
| *IdleState* | • Specifies Idle state Protocol subtype |
| *MultDisc* | • Specifies Generic multimode capability discovery Protocol subtype |
| *VirtStream* | • Specifies Generic Virtual Stream Protocol subtype |

### 8.220.2 Field Documentation

#### 8.220.2.1 uint16_t nas_protocolSubtypeElement::AccessMac

#### 8.220.2.2 uint16_t nas_protocolSubtypeElement::AuthProt

#### 8.220.2.3 uint16_t nas_protocolSubtypeElement::ControlMac

#### 8.220.2.4 uint16_t nas_protocolSubtypeElement::EncryptProt

#### 8.220.2.5 uint16_t nas_protocolSubtypeElement::ForwardMac

#### 8.220.2.6 uint16_t nas_protocolSubtypeElement::IdleState

#### 8.220.2.7 uint16_t nas_protocolSubtypeElement::KeyExchange

#### 8.220.2.8 uint16_t nas_protocolSubtypeElement::MultDisc

#### 8.220.2.9 uint16_t nas_protocolSubtypeElement::PhysicalLayer

#### 8.220.2.10 uint16_t nas_protocolSubtypeElement::ReverseMac

#### 8.220.2.11 uint16_t nas_protocolSubtypeElement::SecProt

#### 8.220.2.12 uint16_t nas_protocolSubtypeElement::VirtStream

## 8.221 nas_qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- uint8_t leapSeconds
- uint8_t localTimeOffset
- uint8_t daylightSavings

### 8.221.1 Detailed Description

This structure contains the 3GPP2TimeZone parameters

**Parameters**

| | |
|---|---|
| *leapSeconds* | • leap seconds - Number of leap seconds since the start of CDMA system time. |
| *localTimeOffset* | • 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. |
| *daylightSavings* | • Day Light Savings Indicator<br>– 0x00 - OFF (daylight savings not in effect)<br>– 0x01 - ON (daylight savings in effect) |

### 8.221.2 Field Documentation

#### 8.221.2.1 uint8_t nas_qaQmi3Gpp2TimeZone::daylightSavings

#### 8.221.2.2 uint8_t nas_qaQmi3Gpp2TimeZone::leapSeconds

#### 8.221.2.3 uint8_t nas_qaQmi3Gpp2TimeZone::localTimeOffset

## 8.222 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 Desription [255]

### 8.222.1 Detailed Description

Contain the 3GPP network information.

**Parameters**

| | |
|---|---|
| *MCC* | • Mobile Country Code |
| *MNC* | • Mobile Network Code |
| *InUse* | • Is the Network the current serving Network<br>– 0 - Unknown<br>– 1 - Current serving network<br>– 2 - Not current serving network, available |
| *Roaming* | • Home/Roam Status of the Network<br>– 0 - Unknown<br>– 1 - Home<br>– 2 - Roam |
| *Forbidden* | • Is the Network in the forbidden network list<br>– 0 - Unknown<br>– 1 - Forbidden<br>– 2 - Not Forbidden |
| *Preferred* | • Is the Network in the Preferred network list<br>– 0 - Unknown<br>– 1 - Preferred<br>– 2 - Not Preferred |
| *Description* | • Network Name/Description<br>• This is a NULL terminated string. |

## 8.222.2 Field Documentation

### 8.222.2.1 char nas_QmiNas3GppNetworkInfo::Desription[255]

### 8.222.2.2 uint32_t nas_QmiNas3GppNetworkInfo::Forbidden

### 8.222.2.3 uint32_t nas_QmiNas3GppNetworkInfo::InUse

### 8.222.2.4 uint16_t nas_QmiNas3GppNetworkInfo::MCC

### 8.222.2.5 uint16_t nas_QmiNas3GppNetworkInfo::MNC

### 8.222.2.6 uint32_t nas_QmiNas3GppNetworkInfo::Preferred

### 8.222.2.7 uint32_t nas_QmiNas3GppNetworkInfo::Roaming

## 8.223 nas_QmiNas3GppNetworkRAT Struct Reference

### Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t RAT

### 8.223.1 Detailed Description

Contain the 3GPP radio access technology information.

**Parameters**

| | |
|---|---|
| *MCC* | • Mobile Country Code |
| *MNC* | • Mobile Network Code |
| *RAT* | • Radio Access Technology<br>– 0x04 - GERAN<br>– 0x05 - UMTS<br>– 0x08 - LTE<br>– 0x09 - TD-SCDMA |

### 8.223.2 Field Documentation

#### 8.223.2.1 uint16_t nas_QmiNas3GppNetworkRAT::MCC

#### 8.223.2.2 uint16_t nas_QmiNas3GppNetworkRAT::MNC

#### 8.223.2.3 uint8_t nas_QmiNas3GppNetworkRAT::RAT

## 8.224 nas_QmisNasPcsDigit Struct Reference

### Data Fields

- uint16_t MCC
- uint16_t MNC
- uint8_t includes_pcs_digit

### 8.224.1 Detailed Description

Contain the PCS Digit information

**Parameters**

| | |
|---|---|
| *MCC* | • Mobile Country Code |

| | |
|---|---|
| *MNC* | • Mobile Network Code |
| *includes_pcs_digit* | • this field is use to interpret the length of corresponding MNC reported<br>• 0x01 - MNC is a three-digit value<br>• 0x00 - MNC is a two-digit value |

### 8.224.2 Field Documentation

#### 8.224.2.1 uint8_t nas_QmisNasPcsDigit::includes_pcs_digit

#### 8.224.2.2 uint16_t nas_QmisNasPcsDigit::MCC

#### 8.224.2.3 uint16_t nas_QmisNasPcsDigit::MNC

## 8.225 nas_RankIndicatorTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t count0
- uint32_t count1

### 8.225.1 Detailed Description

Elements for Rank indicator TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if TLV is present, otherwise 0 |
| *count0* | • count of Rank1 |
| *count1* | • count of Rank2 |

### 8.225.2 Field Documentation

#### 8.225.2.1 uint32_t nas_RankIndicatorTlv::count0

#### 8.225.2.2 uint32_t nas_RankIndicatorTlv::count1

#### 8.225.2.3 uint8_t nas_RankIndicatorTlv::TlvPresent

## 8.226 nas_RatDisabledMaskTlv Struct Reference

### Data Fields

- uint16_t ratDisabledMask
- uint8_t TlvPresent

### 8.226.1 Detailed Description

Contain the RAT disabled mask

**Parameters**

| | |
|---|---|
| *ratDisabledMask* | • Bitmask representing the radio technologies that are disabled.<br>• Bit values:<br>– Bit 0 (0x01) - cdma2000 ® 1X<br>– Bit 1 (0x02) - cdma2000 ® HRPD(1xEV-DO)<br>– Bit 2 (0x04) - GSM<br>– Bit 3 (0x08) - UMTS<br>– Bit 4 (0x10) - LTE<br>– Bit 5 (0x20) - TD-SCDMA<br>– Bit 6 (0x40) - NR5G All unlisted bits are reserved for future use and the service point ignores them if used. |
| *TlvPresent* | • Tlv Present |

### 8.226.2 Field Documentation

#### 8.226.2.1 uint16_t nas_RatDisabledMaskTlv::ratDisabledMask

#### 8.226.2.2 uint8_t nas_RatDisabledMaskTlv::TlvPresent

## 8.227 nas_RejectReasonTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t serviceDomain
- uint32_t rejectCause

### 8.227.1 Detailed Description

This structure contains paramaters reject reason.

**Parameters**

| | |
|---|---|
| *TlvPresent* | indicating the presence of the TLV in the QMI ind |
| *serviceDomain* | service domain |
| *rejectCause* | cause of reject |

### 8.227.2 Field Documentation

#### 8.227.2.1 uint32_t nas_RejectReasonTlv::rejectCause

#### 8.227.2.2 uint32_t nas_RejectReasonTlv::serviceDomain

#### 8.227.2.3 uint8_t nas_RejectReasonTlv::TlvPresent

## 8.228 nas_RFBandInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t activeBandClass
- uint16_t activeChannel

### 8.228.1 Detailed Description

This structure contains the RFBandInfo response parameters.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology<br>– See Tables for Radio Interface |
| *activeBandClass* | • Active Band Class<br>– See Tables for Band Classes |
| *activeChannel* | • Active channel (0 if channel is not relevant to the reported technology) |

### 8.228.2 Field Documentation

#### 8.228.2.1 uint16_t nas_RFBandInfoElements::activeBandClass

#### 8.228.2.2 uint16_t nas_RFBandInfoElements::activeChannel

#### 8.228.2.3 uint8_t nas_RFBandInfoElements::radioInterface

## 8.229 nas_RfBandInfoExtFormat Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t * pInstancesSize
- struct nas_RfBandInfoExtFormatElements * pRfBandInfoExtFormatParam

### 8.229.1 Detailed Description

This structure contains the nas_RfBandInfoExtFormat response parameters.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present |
| *pInstancesSize[-IN/OUT]* | • Number of sets of the RFBandInfoElements<br>• Upon input, maximum number of elements that the RF info instances array can contain.<br>• Upon successful output, actual number of elements in RF info instances array. |
| *pRfBandInfoExt-FormatParam* | • See nas_RfBandInfoExtFormatElements for more information |

### 8.229.2 Field Documentation

#### 8.229.2.1 uint8_t* nas_RfBandInfoExtFormat::pInstancesSize

#### 8.229.2.2 struct nas_RfBandInfoExtFormatElements* nas_RfBandInfoExtFormat::pRfBandInfoExtFormatParam

#### 8.229.2.3 uint8_t nas_RfBandInfoExtFormat::TlvPresent

## 8.230 nas_RfBandInfoExtFormatElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t activeBand
- uint32_t activeChannel

### 8.230.1 Detailed Description

This structure contains the RF Band Information, Extended Format.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology<br>  – See Tables for Radio Interface |
| *activeBand* | • Active Band Class<br>  – See Tables for Band Classes |
| *activeChannel* | • Active channel (0 if channel is not relevant to the reported technology) |

### 8.230.2 Field Documentation

#### 8.230.2.1 uint16_t nas_RfBandInfoExtFormatElements::activeBand

#### 8.230.2.2 uint32_t nas_RfBandInfoExtFormatElements::activeChannel

#### 8.230.2.3 uint8_t nas_RfBandInfoExtFormatElements::radioInterface

## 8.231 nas_RfBandInfoList Struct Reference

### Data Fields

- uint8_t * pInstanceSize
- struct nas_RFBandInfoElements * pRFBandInfoParam

### 8.231.1 Detailed Description

This structure contains the RfBandInfoList response parameters.

**Parameters**

| | |
|---|---|
| *pInstanceSize[IN/OUT]* | • Number of sets of the RFBandInfoElements<br>• Upon input, maximum number of elements that the RF info instances array can contain.<br>• Upon successful output, actual number of elements in RF info instances array. |
| *pRFBandInfoParam* | • See nas_RFBandInfoElements for more information |

### 8.231.2 Field Documentation

#### 8.231.2.1 uint8_t* nas_RfBandInfoList::pInstanceSize

#### 8.231.2.2 struct nas_RFBandInfoElements* nas_RfBandInfoList::pRFBandInfoParam

## 8.232 nas_RfBandwidthInfo Struct Reference

### Data Fields

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

### 8.232.1 Detailed Description

This structure contains the nas_RfBandwidthInfo response parameters.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present |
| *pInstancesSize[IN/OUT]* | • Number of sets of the RFBandInfoElements<br>• Upon input, maximum number of elements that the RF info instances array can contain.<br>• Upon successful output, actual number of elements in RF info instances array. |

| | |
|---|---|
| *pRfBandwidth-InfoParam* | • See nas_RfBandwidthInfoElements for more information |

### 8.232.2 Field Documentation

#### 8.232.2.1 uint8_t* nas_RfBandwidthInfo::pInstancesSize

#### 8.232.2.2 struct nas_RfBandwidthInfoElements* nas_RfBandwidthInfo::pRfBandwidthInfoParam

#### 8.232.2.3 uint8_t nas_RfBandwidthInfo::TlvPresent

## 8.233 nas_RfBandwidthInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint32_t bandwidth

### 8.233.1 Detailed Description

This structure contains the RF Bandwidth Information List parameters.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology<br>  – See Tables for Radio Interface |
| *bandwidth* | • Bandwidth<br>  – Values<br>  * NAS_LTE_BW_NRB_6 (0) - LTE 1.4 MHz bandwidth<br>  * NAS_LTE_BW_NRB_15 (1) - LTE 3 MHz bandwidth<br>  * NAS_LTE_BW_NRB_25 (2) - LTE 5 MHz bandwidth<br>  * NAS_LTE_BW_NRB_50 (3) - LTE 10 MHz bandwidth<br>  * NAS_LTE_BW_NRB_75 (4) - LTE 15 MHz bandwidth<br>  * NAS_LTE_BW_NRB_100 (5) - LTE 20 MHz bandwidth<br>  * NAS_NR5G_BW_NRB_6 (6) - NR5G 1.4 MHz bandwidth<br>  * NAS_NR5G_BW_NRB_15 (7) - NR5G 3 MHz bandwidth<br>  * NAS_NR5G_BW_NRB_25 (8) - NR5G 5 MHz bandwidth<br>  * NAS_NR5G_BW_NRB_50 (9) - NR5G 10 MHz bandwidth<br>  * NAS_NR5G_BW_NRB_75 (10) - NR5G 15 MHz bandwidth<br>  * NAS_NR5G_BW_NRB_100 (11) - NR5G 20 MHz bandwidth |

### 8.233.2 Field Documentation

#### 8.233.2.1 uint32_t nas_RfBandwidthInfoElements::bandwidth

#### 8.233.2.2 uint8_t nas_RfBandwidthInfoElements::radioInterface

## 8.234 nas_RfDedicatedBandInfo Struct Reference

### Data Fields

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

### 8.234.1 Detailed Description

This structure contains the nas_RfDedicatedBandInfo response parameters.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present |
| *pInstancesSize[-IN/OUT]* | • Number of sets of the RFBandInfoElements<br>• Upon input, maximum number of elements that the RF info instances array can contain.<br>• Upon successful output, actual number of elements in RF info instances array. |
| *pRfDedicated-BandInfoParam* | • See nas_RfDedicatedBandInfoElements for more information |

### 8.234.2 Field Documentation

#### 8.234.2.1 uint8_t* nas_RfDedicatedBandInfo::pInstancesSize

#### 8.234.2.2 struct nas_RfDedicatedBandInfoElements* nas_RfDedicatedBandInfo::pRfDedicatedBandInfoParam

#### 8.234.2.3 uint8_t nas_RfDedicatedBandInfo::TlvPresent

## 8.235 nas_RfDedicatedBandInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t dedicatedBand

### 8.235.1 Detailed Description

This structure contains the RF Dedicated Band Info.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology<br>  – See Tables for Radio Interface |

| | |
|---|---|
| *dedicatedBand* | • Dedicated Band Class<br>– See Tables for Band Classes |

### 8.235.2 Field Documentation

#### 8.235.2.1 uint16_t nas_RfDedicatedBandInfoElements::dedicatedBand

#### 8.235.2.2 uint8_t nas_RfDedicatedBandInfoElements::radioInterface

## 8.236 nas_RFInfoTlv Struct Reference

### Data Fields

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

### 8.236.1 Detailed Description

This structure contains paramaters RF Band Information List.

**Parameters**

| | |
|---|---|
| *TlvPresent* | indicating the presence of the TLV in the QMI ind |
| *radioInterface-Size* | Number of sets of the following elements<br>• radioInterface<br>• activeBandClass<br>• activeChannel |
| *radioInterface* | radio interface technology of the signal being measured<br>• 0x01 - cdma2000 1X<br>• 0x02 - cdma2000 HRPD (1xEV-DO)<br>• 0x03 - AMPS<br>• 0x04 - GSM<br>• 0x05 - UMTS<br>• 0x08 - LTE<br>• 0x09 - TD-SCDMA |
| *activeBandClass* | active band class<br>• 00 to 39 - CDMA band classes<br>• 40 to 79 - GSM band classes<br>• 80 to 91 - WCDMA band classes<br>• 120 to 160 - LTE band classes<br>• 200 to 205 - TD-SCDMA band classes |
| *activeChannel* | active channel<br>• Active channel. If the channel is not relevant to the technology, a value of 0 is returned. |

### 8.236.2 Field Documentation

#### 8.236.2.1 uint32_t nas_RFInfoTlv::activeBandClass[255]

#### 8.236.2.2 uint32_t nas_RFInfoTlv::activeChannel[255]

#### 8.236.2.3 uint32_t nas_RFInfoTlv::radioInterface[255]

#### 8.236.2.4 uint8_t nas_RFInfoTlv::radioInterfaceSize

#### 8.236.2.5 uint8_t nas_RFInfoTlv::TlvPresent

## 8.237 nas_roamIndList Struct Reference

### Data Fields

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

### 8.237.1 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**

| | |
|---|---|
| *numInstances* | • number of sets of radio interface currently in use and roaming indicator<br>– defaults to zero |
| *radioInterface* | • Radio Interface currently in use<br>• Values:<br>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X<br>– 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)<br>– 0x03 - RADIO_IF_AMPS - AMPS<br>– 0x04 - RADIO_IF_GSM - GSM<br>– 0x05 - RADIO_IF_UMTS - UMTS<br>– 0x08 - RADIO_IF_LTE - LTE |
| *roamIndicator* | • Roaming Indicator<br>• Values:<br>– 0x00 - Roaming<br>– 0x01 - Home |

### 8.237.2 Field Documentation

#### 8.237.2.1 uint8_t nas_roamIndList::numInstances

#### 8.237.2.2 uint8_t nas_roamIndList::radioInterface[32]

#### 8.237.2.3 uint8_t nas_roamIndList::roamIndicator[32]

## 8.238 nas_RoamPrefTlv Struct Reference

### Data Fields

- uint16_t RoamPref
- uint8_t TlvPresent

### 8.238.1 Detailed Description

Contain the roaming preference for system selection preferences.

**Parameters**

| | |
|---|---|
| *RoamPref* | • Parameter indicating the roaming Preference<br>• Values:<br>– 0x01 - Acquire only systems for which the roaming indicator is off<br>– 0x02 - Acquire a system as long as its roaming indicator is not off<br>– 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only<br>– 0xFF - Acquire systems, regardless of their roaming indicator |
| *TlvPresent* | • Tlv Present |

### 8.238.2 Field Documentation

#### 8.238.2.1 uint16_t nas_RoamPrefTlv::RoamPref

#### 8.238.2.2 uint8_t nas_RoamPrefTlv::TlvPresent

## 8.239 nas_RSRPThresh Struct Reference

### Data Fields

- uint8_t RSRPThresListLen
- int16_t * pRSRPThresList

### 8.239.1 Detailed Description

This structure contains RSRP threshold list.

**Parameters**

| | |
|---|---|
| *RSRPThresList-Len* | • Length of the RSRP threshold list parameter to follow |
| *pRSRPThresList* | • Sequence of thresholds delimiting current RSRP event reporting bands<br>• Every time a new RSRP value crosses a specified threshold value, an event report indication message with the new RSRQ value is sent to the requesting control point. For this field<br>– RSRP values are applicable only for LTE<br>– RSRP values are measured in dBm, with a range of -44 dBm to -140 dBm<br>– Each RSRP threshold value is a signed byte value<br>– Maximum number of threshold values is 16<br>– At least one value must be specified |

### 8.239.2 Field Documentation

#### 8.239.2.1 int16_t∗ nas_RSRPThresh::pRSRPThresList

#### 8.239.2.2 uint8_t nas_RSRPThresh::RSRPThresListLen

## 8.240 nas_rsrqInformation Struct Reference

### Data Fields

- int8_t rsrq
- uint8_t radioIf

### 8.240.1 Detailed Description

This structure contains the RSRQ Information

**Parameters**

| | |
|---|---|
| *rsrq* | • RSRQ value in dB (signed integer value); valid range is -3 to -20 (-3 means -3 dB, -20 means -20 dB) |
| *radioIf* | • Radio interface technology of the signal being measured<br>– 0x08 – LTE |

### 8.240.2 Field Documentation

#### 8.240.2.1 uint8_t nas_rsrqInformation::radioIf

#### 8.240.2.2 int8_t nas_rsrqInformation::rsrq

## 8.241 nas_RSRQThresh Struct Reference

### Data Fields

- uint8_t RSRQThresListLen
- int8_t * pRSRQThresList

### 8.241.1 Detailed Description

This structure contains RSRQ threshold list.

**Parameters**

| | |
|---|---|
| *RSRQThresList-Len* | • Length of the RSRQ threshold list parameter to follow |
| *pRSRQThresList* | • Sequence of thresholds delimiting current RSRQ event reporting bands<br>• Every time a new RSRQ value crosses a threshold value, an event report indication message with the new RSRQ value is sent to the requesting control point. For this field<br>– RSRQ values are applicable only for LTE<br>– RSRQ values are measured in dBm, with a range of -20 dBm to -3 dBm<br>– Each RSRQ threshold value is a signed byte value<br>– Maximum number of threshold values is 16<br>– At least one value must be specified |

### 8.241.2 Field Documentation

#### 8.241.2.1 int8_t* nas_RSRQThresh::pRSRQThresList

#### 8.241.2.2 uint8_t nas_RSRQThresh::RSRQThresListLen

## 8.242 nas_RSSIThresh Struct Reference

### Data Fields

- uint8_t RSSIThresListLen
- int8_t * pRSSIThresList

### 8.242.1 Detailed Description

This sturcture contain RSSI threshold list.

**Parameters**

| | |
|---|---|
| *RSSIThresList-Len* | • Length of the RSSI threshold list parameter to follow |
| *pRSSIThresList* | • RSSI in dBm( signed bytes )<br>• A value of -125 dBm or lower is used to indicate No Signal<br>• RSSI values have the following ranges (in dBm)<br>– CDMA is -105 to -21<br>– HDR is -118 to -13<br>– GSM is -111 to -48<br>– WCDMA is -121 to 0<br>– LTE is -120 to 0<br>• Threshold values specified above are used for all RATs<br>• The maximum number of threshold values is 16, each a signed byte value. |

### 8.242.2 Field Documentation

#### 8.242.2.1 int8_t∗ nas_RSSIThresh::pRSSIThresList

#### 8.242.2.2 uint8_t nas_RSSIThresh::RSSIThresListLen

## 8.243 nas_rxInfo Struct Reference

### Data Fields

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

### 8.243.1 Detailed Description

This structure contains rx radio infomation.

**Parameters**

| | |
|---|---|
| *isRadioTuned* | • Whether Rx is tuned to a channel:<br>– 0x00 - Not tuned<br>– 0x01 - Tuned<br>– 0xFF - Not Available<br>• If the radio is tuned, instantaneous values are set for the signal information fields below.<br>• If the radio is not tuned, or is delayed or invalid, the values are set depending on each technology. |
| *rx_pwr* | • Rx power value in 1/10 dbm resolution. |

| | |
|---|---|
| *ecio* | • ECIO in 1/10 dB; valid for CDMA, HDR, GSM, WCDMA, and LTE. |
| *rscp* | • Received signal code power in 1/10 dbm.<br>• Valid for WCDMA. |
| *rsrp* | • Current reference signal received power in 1/10 dbm valid for LTE. |
| *phase* | • Phase in 1/100 degrees; valid for LTE.<br>• When the phase is unknown, 0xFFFFFFFF is used. |

### 8.243.2 Field Documentation

#### 8.243.2.1 int32_t nas_rxInfo::ecio

#### 8.243.2.2 uint8_t nas_rxInfo::isRadioTuned

#### 8.243.2.3 uint32_t nas_rxInfo::phase

#### 8.243.2.4 int32_t nas_rxInfo::rscp

#### 8.243.2.5 int32_t nas_rxInfo::rsrp

#### 8.243.2.6 int32_t nas_rxInfo::rxPower

## 8.244 nas_RxSigInfo Struct Reference

### Data Fields

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

### 8.244.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

**Parameters**

| | |
|---|---|
| *rxChainIndex* | • Rx antenna path<br>• Valid Values<br>  – 0 - Primary Rx<br>  – 1 - Diversity Rx |

| | |
|---|---|
| *isRadioTuned* | • Rx path is tuned to a channel or Not<br>• Values<br>– 0x00 - Not tuned<br>– 0x01 - Tuned |

**Note**

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

**Parameters**

| | |
|---|---|
| *rxPower* | • Rx power value in 1/10 dBm resolution |
| *rsrp* | • Current reference signal received power in 1/10 dBm resolution |

### 8.244.2 Field Documentation

#### 8.244.2.1 uint8_t nas_RxSigInfo::isRadioTuned

#### 8.244.2.2 int32_t nas_RxSigInfo::rsrp

#### 8.244.2.3 uint8_t nas_RxSigInfo::rxChainIndex

#### 8.244.2.4 int32_t nas_RxSigInfo::rxPower

## 8.245 nas_rxSignalStrengthListElement Struct Reference

### Data Fields

- int16_t rxSignalStrength
- uint8_t radioIf

### 8.245.1 Detailed Description

This structure contains the Received Signal Strength Information

**Parameters**

| | |
|---|---|
| *rxSignalStrength* | • Received signal strength in dBm<br>– For CDMA and UMTS, this indicates forward link pilotEc.<br>– For GSM, the received signal strength.<br>– For LTE, this indicates the total received wideband power observed by UE. |

| | |
|---|---|
| *radioIf* | • Radio interface technology of the signal being radio_if measured<br>– 0x00 – RADIO_IF_NO_SVC – None (no service)<br>– 0x01 – RADIO_IF_CDMA_1X – cdma2000® 1X<br>– 0x02 – RADIO_IF_CDMA_1XEVDO – cdma2000 HRPD (1xEV-DO)<br>– 0x03 – RADIO_IF_AMPS – AMPS<br>– 0x04 – RADIO_IF_GSM – GSM<br>– 0x05 – RADIO_IF_UMTS – UMTS<br>– 0x08 – RADIO_IF_LTE – LTE |

**Note**

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

### 8.245.2 Field Documentation

#### 8.245.2.1 uint8_t nas_rxSignalStrengthListElement::radioIf

#### 8.245.2.2 int16_t nas_rxSignalStrengthListElement::rxSignalStrength

## 8.246 nas_SccRxInfo Struct Reference

### Data Fields

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

### 8.246.1 Detailed Description

This structure contains information about the SccRxInfo parameters.

**Parameters**

| | |
|---|---|
| *rsrq* | • Current reference signal<br>• Receive quality in 1/10 dB resolution |
| *snr* | • Reference signal signal-to-noise ratio in dB.<br>• Range -10 to 30 |
| *numInstances* | • Number of sets of the following<br>– rxChainIndex<br>– isRadioTuned<br>– rxPower<br>– rsrp |

| | |
|---|---|
| *sigInfo* | • See nas_RxSigInfo for more information |
| *TlvPresent* | • Tlv Present. |

### 8.246.2 Field Documentation

#### 8.246.2.1 uint8_t nas_SccRxInfo::numInstances

#### 8.246.2.2 int32_t nas_SccRxInfo::rsrq

#### 8.246.2.3 nas_RxSigInfo nas_SccRxInfo::sigInfo[255]

#### 8.246.2.4 int16_t nas_SccRxInfo::snr

#### 8.246.2.5 uint8_t nas_SccRxInfo::TlvPresent

## 8.247 nas_serviceProviderName Struct Reference

### Data Fields

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

### 8.247.1 Detailed Description

This structure contains service provider name.

**Parameters**

| | |
|---|---|
| *displayCondition* | • Display condition. |
| *spnLength* | • It provides length of spn. |
| *spn* | • Service provider name string must use: The SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 9. |

### 8.247.2 Field Documentation

#### 8.247.2.1 uint8_t nas_serviceProviderName::displayCondition

#### 8.247.2.2 uint8_t nas_serviceProviderName::spn[255]

#### 8.247.2.3 uint8_t nas_serviceProviderName::spnLength

## 8.248 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]

### 8.248.1 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**

| | |
|---|---|
| *regState* | • Registration state - Registration state of the mobile<br>• Values:<br>– 0 - Not Registered; mobile is not currently searching for a new network to provide service<br>– 1 - Registered with a network<br>– 2 - Not registered, but mobile is currently searching for a new network to provide service<br>– 3 - Registration denied by visible network<br>– 4 - Registration state is unknown |
| *csAttachState* | • CS Attach State - Circuit-switched domain attach state of the mobile<br>• Values:<br>– 0 - Unknown or not applicable<br>– 1 - Attached<br>– 2 - Detached |
| *psAttachState* | • PS Attach State - Packet-switched domain attach state of the mobile<br>• Values:<br>– 0 - Unknown or not applicable<br>– 1 - Attached<br>– 2 - Detached |
| *selNetwork* | • Selected Network - Type of selected radio access network<br>• Values:<br>– 0 - Unknown<br>– 1 - 3GPP2 network<br>– 2 - 3GPP network |
| *numRadio-Interfaces* | • In Use Radio Interfaces Number<br>– Number of radio interfaces currently in use<br>– defaults to zero |

| | |
|---|---|
| *radioInterface* | • Radio Interface currently in use<br>• Values:<br>– 0x00 - RADIO_IF_NO_SVC - None(no service)<br>– 0x01 - RADIO_IF_CDMA_1X - cdma2000 1X<br>– 0x02 - RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)<br>– 0x03 - RADIO_IF_AMPS - AMPS<br>– 0x04 - RADIO_IF_GSM - GSM<br>– 0x05 - RADIO_IF_UMTS - UMTS<br>– 0x08 - RADIO_IF_LTE - LTE |

### 8.248.2 Field Documentation

#### 8.248.2.1 uint8_t nas_servSystem::csAttachState

#### 8.248.2.2 uint8_t nas_servSystem::numRadioInterfaces

#### 8.248.2.3 uint8_t nas_servSystem::psAttachState

#### 8.248.2.4 uint8_t nas_servSystem::radioInterface[32]

#### 8.248.2.5 uint8_t nas_servSystem::regState

#### 8.248.2.6 uint8_t nas_servSystem::selNetwork

## 8.249 nas_sidNid Struct Reference

### Data Fields

- uint16_t nid
- uint16_t sid

### 8.249.1 Detailed Description

This structure contains the parameters for SidNid Information

**Parameters**

| | |
|---|---|
| *nid* | • Network ID |
| *sid* | • System ID |

### 8.249.2 Field Documentation

#### 8.249.2.1 uint16_t nas_sidNid::nid

#### 8.249.2.2 uint16_t nas_sidNid::sid

## 8.250 nas_SignalStrengthTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- int8_t signalStrength
- uint32_t radioInterface

### 8.250.1 Detailed Description

This structure contains paramaters Signal Strength.

**Parameters**

| | |
|---|---|
| *TlvPresent* | indicating the presence of the TLV in the QMI ind |
| *signalStrength* | signal strength |
| *radioInterface* | radio interface technology of the signal being measured |

### 8.250.2 Field Documentation

#### 8.250.2.1 uint32_t nas_SignalStrengthTlv::radioInterface

#### 8.250.2.2 int8_t nas_SignalStrengthTlv::signalStrength

#### 8.250.2.3 uint8_t nas_SignalStrengthTlv::TlvPresent

## 8.251 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

### 8.251.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *rxSignal-StrengthDelta* | • RSSI delta(in dBm) at which an event report indication, including the current RSSI, will be sent to the requesting control point. |

| | |
|---|---|
| *ecioDelta* | • ECIO delta at which an event report indication, ecioDelta including the current ECIO, will be sent to the requesting control point.<br>• ECIO delta is an unsigned 1 byte value that increments in negative 0.5 dBm, e.g., ecio_delta of 2 means a change of -1 dBm. |
| *ioDelta* | • IO delta (in dBm) at which an event report indication, ioDelta including the current IO, will be sent to the requesting control point. |
| *sinrDelta* | • SINR delta level at which an event report indication, sinrDelta including the current SINR, will be sent to the requesting control point. |
| *rsrqDelta* | • RSRQ delta level at which an event report indication, including the current RSRQ, will be sent to the requesting control point. |
| *ecioThreshold-ListLen* | • Number of elements in the ECIO threshold list. |
| *ecioThreshold-List* | • A sequence of thresholds delimiting Ecio event reporting bands. Every time a new Ecio value crosses a threshold value, an event report indication message with the new ECIO value is sent to the requesting control point. For this field:<br>– Maximum number of threshold values is 10<br>– At least one value must be specified. |
| *sinrThreshold-ListLen* | • Number of elements in the SINR threshold list. |
| *sinrThreshold-List* | • A sequence of thresholds delimiting SINR event reporting bands. Every time a new SINR value crosses a threshold value, an event report indication message with the new sinr value is sent to the requesting control point. For this field:<br>– Maximum number of threshold values is 5<br>– At least one value must be specified. |
| *ltesnrdelta* | • LTE SNR delta level at which an event report indication, including the current SNR, will be sent to the requesting control point. LTE SNR delta level is an unsigned 2 byte value, representing the delta in units of 0.1 dB, e.g., lte_snr_delta of 3 means a change 0.3dB. |
| *ltersrpdelta* | -LTE RSRP delta level at which an event report -indication, including the current RSRP, will be sent -to the requesting control point. LTE RSRP delta -level is an unsigned 1 byte value, representing the -delta in dB. |

### 8.251.2 Field Documentation

#### 8.251.2.1 uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta

#### 8.251.2.2 int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]

#### 8.251.2.3 uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen

#### 8.251.2.4 uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta

#### 8.251.2.5 uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta

#### 8.251.2.6 uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta

#### 8.251.2.7 uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta

#### 8.251.2.8 uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

#### 8.251.2.9 uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta

#### 8.251.2.10 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

#### 8.251.2.11 uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen

## 8.252 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

### 8.252.1 Detailed Description

This structure contains paramaters signal strengths information.

**Parameters**

| | |
|---|---|
| *rxSignal-StrengthInfo* | • signal strength info list<br>• Bit to check in ParamPresenceMask - **19** |
| *ecioInfo* | • ecio info list<br>• Bit to check in ParamPresenceMask - **20** |
| *io* | • received IO in dBm; IO is only applicable for 1xEV-DO<br>• Bit to check in ParamPresenceMask - **21** |
| *sinr* | • SINR level<br>• Bit to check in ParamPresenceMask - **22** |

| | |
|---|---|
| *errorRateInfo* | • error rate info<br>• Bit to check in ParamPresenceMask - **23** |
| *rsrqInfo* | • rsrq info<br>• Bit to check in ParamPresenceMask - **24** |
| *lteSnrinfo* | • lte Snr information<br>• Bit to check in ParamPresenceMask - **25** |
| *lteRsrpinfo* | • lte rsrp info<br>• Bit to check in ParamPresenceMask - **26** |

### 8.252.2 Field Documentation

#### 8.252.2.1 nas_ecioListElement nas_SLQSSignalStrengthsInformation::ecioInfo

#### 8.252.2.2 nas_errorRateListElement nas_SLQSSignalStrengthsInformation::errorRateInfo

#### 8.252.2.3 uint32_t nas_SLQSSignalStrengthsInformation::io

#### 8.252.2.4 nas_lteRsrpinformation nas_SLQSSignalStrengthsInformation::lteRsrpinfo

#### 8.252.2.5 nas_lteSnrinformation nas_SLQSSignalStrengthsInformation::lteSnrinfo

#### 8.252.2.6 nas_rsrqInformation nas_SLQSSignalStrengthsInformation::rsrqInfo

#### 8.252.2.7 nas_rxSignalStrengthListElement nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo

#### 8.252.2.8 uint8_t nas_SLQSSignalStrengthsInformation::sinr

## 8.253 nas_SLQSSignalStrengthsTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- nas_SLQSSignalStrengthsInformation sSLQSSignalStrengthsInfo

### 8.253.1 Detailed Description

This structure contains paramaters SLQS Signal Strength.

**Parameters**

| | |
|---|---|
| *TlvPresent* | indicating the presence of the TLV in the QMI ind |
| *sSLQSSignal-StrengthsInfo* | signal strength info |

### 8.253.2 Field Documentation

#### 8.253.2.1 nas_SLQSSignalStrengthsInformation nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo

#### 8.253.2.2 uint8_t nas_SLQSSignalStrengthsTlv::TlvPresent

## 8.254 nas_SrvDomainPrefTlv Struct Reference

### Data Fields

- uint32_t SrvDomainPref
- uint8_t TlvPresent

### 8.254.1 Detailed Description

Contain the Service domain Preference for system selection preferences.

**Parameters**

| | |
|---|---|
| *SrvDomainPref* | • Parameter indicating Service domain preference<br>• Values:<br>– 0x00 - Circuit switched only<br>– 0x01 - Packet switched only<br>– 0x02 - Circuit switched and packet switched<br>– 0x03 - Packet switched attach<br>– 0x04 - Packet switched detach |
| *TlvPresent* | • Tlv Present |

### 8.254.2 Field Documentation

#### 8.254.2.1 uint32_t nas_SrvDomainPrefTlv::SrvDomainPref

#### 8.254.2.2 uint8_t nas_SrvDomainPrefTlv::TlvPresent

## 8.255 nas_SrvStatusInfo Struct Reference

### Data Fields

- uint8_t srvStatus
- uint8_t isPrefDataPath

### 8.255.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *srvStatus* | • Service status of the system.<br>– 0x00 - No service<br>– 0x01 - Limited service<br>– 0x02 - Service<br>– 0x03 - Limited regional service<br>– 0x04 - Power save<br>– 0xFF - Not Available |
| *isPrefDataPath* | • Whether the RAT is the preferred data path.<br>– 0x00 - Not preferred<br>– 0x01 - Preferred<br>– 0xFF - Not Available |

### 8.255.2 Field Documentation

#### 8.255.2.1 uint8_t nas_SrvStatusInfo::isPrefDataPath

#### 8.255.2.2 uint8_t nas_SrvStatusInfo::srvStatus

## 8.256 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

### 8.256.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *srvDomainValid* | • Indicates whether the service domain is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |

| | |
|---|---|
| *srvDomain* | • Service domain registered on the system.<br>– 0x00 - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>– 0x04 - Camped<br>– 0xFF - Not Available |
| *srvCapability-Valid* | • Indicates whether the service capability is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *srvCapability* | • Current system's service capability.<br>– 0x00 - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>– 0x04 - Camped<br>– 0xFF - Not Available |
| *roamStatusValid* | • Indicates whether the roaming status is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |

| | |
|---|---|
| *roamStatus* | • Current roaming status.<br>– 0x00 - Off<br>– 0x01 - On<br>– 0x02 - Blinking<br>– 0x03 - Out of the neighborhood<br>– 0x04 - Out of the building<br>– 0x05 - Preferred system<br>– 0x06 - Available system<br>– 0x07 - Alliance partner<br>– 0x08 - Premium partner<br>– 0x09 - Full service<br>– 0x0A - Partial service<br>– 0x0B - Banner is on<br>– 0x0C - Banner is off<br>– 0x0D to 0x3F - Reserved for Standard Enhanced Roaming Indicator Numbers<br>– 0x40 to 0x7F - Reserved for Non-Standard Enhanced Roaming Indicator Numbers<br>– 0x40 to 0xFF - Reserved.<br>– 0xFF - Not Available<br>• Values from 0x02 onward are only applicable for 3GPP2 |
| *isSysForbidden-Valid* | • Indicates whether the forbidden system is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *isSysForbidden* | • Whether the system is forbidden.<br>– 0x00 - Not forbidden<br>– 0x01 - Forbidden<br>– 0xFF - Not Available |

### 8.256.2 Field Documentation

#### 8.256.2.1 uint8_t nas_sysInfoCommon::isSysForbidden

#### 8.256.2.2 uint8_t nas_sysInfoCommon::isSysForbiddenValid

#### 8.256.2.3 uint8_t nas_sysInfoCommon::roamStatus

#### 8.256.2.4 uint8_t nas_sysInfoCommon::roamStatusValid

#### 8.256.2.5 uint8_t nas_sysInfoCommon::srvCapability

#### 8.256.2.6 uint8_t nas_sysInfoCommon::srvCapabilityValid

#### 8.256.2.7 uint8_t nas_sysInfoCommon::srvDomain

#### 8.256.2.8 uint8_t nas_sysInfoCommon::srvDomainValid

## 8.257 nas_TDSCDMAECIOThresh Struct Reference

### Data Fields

- uint8_t TDSCDMAECIOThreshListLen
- float * pTDSCDMAECIOThreshList

### 8.257.1 Detailed Description

This structure contains TDSCDMA ECIO threshold related parameters.

**Parameters**

| | |
|---|---|
| *TDSCDMAECIOThreshListLen* | • Length of the TDSCDMA ECIO threshold list parameter to follow |
| *pTDSCDMAECIOThreshList* | • Array of ECIO thresholds (in dB) used by TD-SCDMA<br>• Maximum of 32 values. |

### 8.257.2 Field Documentation

#### 8.257.2.1 float* nas_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

#### 8.257.2.2 uint8_t nas_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.258 nas_TDSCDMARSCPThresh Struct Reference

### Data Fields

- uint8_t TDSCDMARSCPThreshListLen
- int16_t * pTDSCDMARSCPThreshList

### 8.258.1 Detailed Description

This structure contains TDSCDMA RSCP threshold related parameters.

**Parameters**

| | |
|---|---|
| *TDSCDMARSCPThreshListLen* | • Length of the TDSCDMA RSCP threshold list parameter to follow |
| *pTDSCDMARSCPThreshList* | • Array of RSCP thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values<br>• Range for RSCP values: -120 to -25 (in dBm). |

### 8.258.2 Field Documentation

#### 8.258.2.1 int16_t* nas_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

#### 8.258.2.2 uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.259 nas_TDSCDMARSSIThresh Struct Reference

### Data Fields

- uint8_t TDSCDMARSSIThreshListLen
- float * pTDSCDMARSSIThreshList

### 8.259.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

**Parameters**

| | |
|---|---|
| *TDSCDMARSSIThreshListLen* | • Length of the TDSCDMA RSSI threshold list parameter to follow |
| *pTDSCDMARSSIThreshList* | • Array of RSSI thresholds (in dBm) used by TD-SCDMA<br>• Maximum of 32 values. |

### 8.259.2 Field Documentation

#### 8.259.2.1 float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

#### 8.259.2.2 uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.260 nas_TDSCDMASINRCONFThresh Struct Reference

### Data Fields

- uint8_t TDSCDMASINRCONFThreshListLen
- float * pTDSCDMASINRCONFThreshList

### 8.260.1 Detailed Description

This structure contains TDSCDMA SINR threshold list.

**Parameters**

| | |
|---|---|
| *TDSCDMASINRCONFThreshListLen* | • Length of the TDSCDMA SINR threshold list parameter to follow |
| *pTDSCDMASINRCONFThreshList* | • Array of SINR thresholds (in dB) used by TD-SCDMA<br>• Maximum of 32 values |

### 8.260.2 Field Documentation

#### 8.260.2.1 float∗ nas_TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

#### 8.260.2.2 uint8_t nas_TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

## 8.261 nas_TDSCDMASINRThresh Struct Reference

### Data Fields

- uint8_t TDSCDMASINRThreshListLen
- float ∗ pTDSCDMASINRThreshList

### 8.261.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

**Parameters**

| | |
|---|---|
| *TDSCDMASINRThreshListLen* | • Length of the TDSCDMA SINR threshold list parameter to follow |
| *pTDSCDMASINRThreshList* | • Array of SINR thresholds (in dB) used by TD-SCDMA<br>• Maximum of 32 values |

### 8.261.2 Field Documentation

#### 8.261.2.1 float∗ nas_TDSCDMASINRThresh::pTDSCDMASINRThreshList

#### 8.261.2.2 uint8_t nas_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.262 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

### 8.262.1 Detailed Description

This structure contains the parameters for Network Time.

**Parameters**

| | |
|---|---|
| *year* | • Year |
| *month* | • Month<br>• 1 is January and 12 is December |
| *day* | • Day<br>• Range - 1 to 31 |
| *hour* | • Hour<br>• Range - 0 to 59 |
| *minute* | • Minute<br>• Range - 0 to 59 |
| *second* | • Second<br>• Range - 0 to 59 |
| *dayOfWeek* | • Day of the week<br>• 0 is Monday and 6 is Sunday |
| *timeZone* | • Offset from Universal time<br>• The difference between local time and Universal time, in increments of 15 min<br>• Signed Value |
| *dayLtSavingAdj* | • Daylight saving adjustment in hours<br>• Possible values - 0, 1, and 2.<br>• This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVDO |
| *radioInterface* | • Radio interface from which the information comes<br>• Values<br>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X<br>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)<br>– 0x04 - NAS_RADIO_IF_GSM - GSM<br>– 0x05 - NAS_RADIO_IF_UMTS - UMTS<br>– 0x08 - NAS_RADIO_IF_LTE - LTE<br>– 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA |
| *TlvPresent* | • Tlv Present. |

### 8.262.2 Field Documentation

#### 8.262.2.1 uint8_t nas_timeInfo::day

#### 8.262.2.2 uint8_t nas_timeInfo::dayLtSavingAdj

#### 8.262.2.3 uint8_t nas_timeInfo::dayOfWeek

#### 8.262.2.4 uint8_t nas_timeInfo::hour

#### 8.262.2.5 uint8_t nas_timeInfo::minute

#### 8.262.2.6 uint8_t nas_timeInfo::month

#### 8.262.2.7 uint8_t nas_timeInfo::radioInterface

#### 8.262.2.8 uint8_t nas_timeInfo::second

#### 8.262.2.9 int8_t nas_timeInfo::timeZone

#### 8.262.2.10 uint8_t nas_timeInfo::TlvPresent

#### 8.262.2.11 uint16_t nas_timeInfo::year

## 8.263 nas_trueIMSI Struct Reference

### Data Fields

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

### 8.263.1 Detailed Description

This structure contains True IMSI.

**Parameters**

| | |
|---|---|
| *mccT* | • ASCII character representation of MCC_T |
| *imsiT1112* | • ASCII character representation of IMSI_T_11_12 value<br>– 0xFFFF - Not Available |
| *imsiTS1* | • ASCII character representation of IMSI_T_S1 value |
| *imsiTS2* | • ASCII character representation of IMSI_T_S2 value |

| | |
|---|---|
| *imsiTaddrNum* | • Value of IMSI_T_ADDR_NUM<br>– 0xFF - Not Available |

### 8.263.2 Field Documentation

#### 8.263.2.1 uint16_t nas_truelMSI::imsiT1112

#### 8.263.2.2 uint8_t nas_truelMSI::imsiTaddrNum

#### 8.263.2.3 uint8_t nas_truelMSI::imsiTS1[7]

#### 8.263.2.4 uint8_t nas_truelMSI::imsiTS2[3]

#### 8.263.2.5 uint8_t nas_truelMSI::mccT[3]

## 8.264 nas_txInfo Struct Reference

### Data Fields

- uint8_t isInTraffic
- int32_t txPower

### 8.264.1 Detailed Description

This structure contains tx radio infomation.

**Parameters**

| | |
|---|---|
| *isInTraffic* | • Whether the device is in traffic.<br>– 0x00 - not in traffic<br>– 0x01 - in traffic<br>• The txPower field is only meaningful when in the device is in traffic. |
| *txPower* | • Tx power value in 1/10 dbm. |

### 8.264.2 Field Documentation

#### 8.264.2.1 uint8_t nas_txInfo::isInTraffic

#### 8.264.2.2 int32_t nas_txInfo::txPower

## 8.265 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]

### 8.265.1 Detailed Description

This structure contains information about the UMTS Network.

**Parameters**

| | |
|---|---|
| *cellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present. |
| *plmn[NAS_PLMN_LENGTH]* | • MCC/MNC information coded as octet 3, 4, and 5.<br>• This field is ignored when nmrCellID is not present. |
| *lac* | • Location area code.<br>• This field is ignored when nmrCellID is not present.<br>  – 0xFFFF - Not Available |
| *uarfcn* | • UTRA absolute RF channel number.<br>  – 0xFFFF - Not Available |
| *psc* | • Primary scrambling code.<br>  – 0xFFFF - Not Available |
| *rscp* | • Received signal code power.<br>  – 0xFFFF - Not Available |
| *ecio* | • ECIO(Signal-to-Interference-ratio).<br>  – 0xFFFF - Not Available |
| *umtsInst* | • Provides the number of set of UMTS info instances.<br>• If 0(zero), then no information follows it. |
| *UMTSInstInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_UMTSinstInfo for more information. |

| | |
|---|---|
| *geranInst* | • Provides the number of set of GERAN info instances.<br>• If 0(zero), then no information follows it. |
| *GeranInstInfo[NAS_MAX_DESCRIPTION_LENGTH]* | • See nas_geranInstInfo for more information. |

### 8.265.2 Field Documentation

#### 8.265.2.1 uint16_t nas_UMTSInfo::cellID

#### 8.265.2.2 int16_t nas_UMTSInfo::ecio

#### 8.265.2.3 uint8_t nas_UMTSInfo::geranInst

#### 8.265.2.4 nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]

#### 8.265.2.5 uint16_t nas_UMTSInfo::lac

#### 8.265.2.6 uint8_t nas_UMTSInfo::plmn[3]

#### 8.265.2.7 uint16_t nas_UMTSInfo::psc

#### 8.265.2.8 int16_t nas_UMTSInfo::rscp

#### 8.265.2.9 uint16_t nas_UMTSInfo::uarfcn

#### 8.265.2.10 uint8_t nas_UMTSInfo::umtsInst

#### 8.265.2.11 nas_UMTSinstInfo nas_UMTSInfo::UMTSInstInfo[255]

## 8.266 nas_UMTSinstInfo Struct Reference

### Data Fields

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

### 8.266.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *umtsUarfcn* | • UTRA absolute RF channel number. |
| *umtsPsc* | • Primary scrambling code. |

| | |
|---|---|
| *umtsRscp* | • Received signal code power. |
| *umtsEcio* | • ECIO(Signal-to-Interference-ratio). |

### 8.266.2 Field Documentation

#### 8.266.2.1 int16_t nas_UMTSinstInfo::umtsEcio

#### 8.266.2.2 uint16_t nas_UMTSinstInfo::umtsPsc

#### 8.266.2.3 int16_t nas_UMTSinstInfo::umtsRscp

#### 8.266.2.4 uint16_t nas_UMTSinstInfo::umtsUarfcn

## 8.267 nas_umtsLTENbrCell Struct Reference

### Data Fields

- uint16_t earfcn
- uint16_t pci
- uint32_t rsrp
- uint32_t rsrq
- int16_t srxlev
- uint8_t cellIsTDD

### 8.267.1 Detailed Description

This structure contains information about the UMTS LTE neighbour Cell.

**Parameters**

| | |
|---|---|
| *earfcn* | • E-UTRA absolute RF channel number of the detected cell. |
| *pci* | • Physical cell ID of the detected cell.<br>• Range is defined in 3GPP TS 36.211 |
| *rsrp* | • Current received signal strength indication (in dBm) of the detected cell. |
| *rsrq* | • Current reference signal received quality (in dB) of the detected cell. |
| *srxlev* | • Cell selection Rx level (Srxlev) value of the detected cell in linear scale.<br>• This field is only valid when wcdma_rrc_state is not NAS_WCDMA_RRC_STATE_CEL_FACH or NAS_WCDMA_RRC_STATE_CELL_DCH. |

| | |
|---|---|
| *cellIsTDD* | • TRUE if the cell is TDD; FALSE if the cell is FDD. |

### 8.267.2 Field Documentation

#### 8.267.2.1 uint8_t nas_umtsLTENbrCell::cellIsTDD

#### 8.267.2.2 uint16_t nas_umtsLTENbrCell::earfcn

#### 8.267.2.3 uint16_t nas_umtsLTENbrCell::pci

#### 8.267.2.4 uint32_t nas_umtsLTENbrCell::rsrp

#### 8.267.2.5 uint32_t nas_umtsLTENbrCell::rsrq

#### 8.267.2.6 int16_t nas_umtsLTENbrCell::srxlev

## 8.268 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

### 8.268.1 Detailed Description

This structure contains the parameters for Universal Time Information.

**Parameters**

| | |
|---|---|
| *year* | • Year. |
| *month* | • Month.<br>  – 1 is January and 12 is December. |
| *day* | • Day.<br>  – Range 1 to 31. |
| *hour* | • Hour.<br>  – Range 0 to 59. |

| | |
|---|---|
| *minute* | • Minute.<br>– Range 0 to 59. |
| *second* | • Second.<br>– Range 0 to 59. |
| *dayOfWeek* | • Day of the Week.<br>– 0 is Monday and 6 is Sunday. |

### 8.268.2 Field Documentation

#### 8.268.2.1 uint8_t nas_UniversalTime::day

#### 8.268.2.2 uint8_t nas_UniversalTime::dayOfWeek

#### 8.268.2.3 uint8_t nas_UniversalTime::hour

#### 8.268.2.4 uint8_t nas_UniversalTime::minute

#### 8.268.2.5 uint8_t nas_UniversalTime::month

#### 8.268.2.6 uint8_t nas_UniversalTime::second

#### 8.268.2.7 uint16_t nas_UniversalTime::year

## 8.269 nas_wcdmaCellInfo Struct Reference

### Data Fields

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

### 8.269.1 Detailed Description

This structure contains information about the WCDMA Cell.

**Parameters**

| | |
|---|---|
| *psc* | • Primary scrambling code.<br>• Range: 0 to 511. |
| *cpich_rscp* | • Absolute power level (in 1/10 dBm) of the common pilot channel as received by the UE.<br>• Range: -120.0 dBm to -25.0 dBm |

| | |
|---|---|
| *cpich_ecno* | • 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.<br>• Range: -50.0 dB to 0. |
| *srxlev* | • Cell selection Rx level (Srxlev) value.<br>• Range: -128 to 128.<br>• This field is only valid when ue_in_idle is TRUE. |

### 8.269.2 Field Documentation

#### 8.269.2.1 int16_t nas_wcdmaCellInfo::cpich_ecno

#### 8.269.2.2 int16_t nas_wcdmaCellInfo::cpich_rscp

#### 8.269.2.3 uint16_t nas_wcdmaCellInfo::psc

#### 8.269.2.4 int16_t nas_wcdmaCellInfo::srxlev

## 8.270 nas_WCDMAECIOThresh Struct Reference

### Data Fields

- uint8_t WCDMAECIOThreshListLen
- int16_t ∗ pWCDMAECIOThreshList

### 8.270.1 Detailed Description

This structure contains WCDMA ECIO threshold related parameters.

**Parameters**

| | |
|---|---|
| *WCDMAECIO-ThreshListLen* | • Length of the WCDMA ECIO threshold list parameter to follow |
| *pWCDMAECIO-ThreshList* | • Array of ECIO thresholds (in units of 0.1 dB)<br>• Maximum of 32 values<br>• Range for ECIO values: -31.5 to 0 (in dB) |

### 8.270.2 Field Documentation

#### 8.270.2.1 int16_t∗ nas_WCDMAECIOThresh::pWCDMAECIOThreshList

#### 8.270.2.2 uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen

## 8.271 nas_WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- uint32_t wcdmaRRCState
- uint8_t umtsLTENbrCellLen
- nas_umtsLTENbrCell UMTSLTENbrCell [255]

### 8.271.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *wcdmaRRC-State* | • WCDMA RRC states.<br>• Defined in 3GPP TS 25.331<br>• Values:<br>– 0x00 - NAS_WCDMA_RRC_STATE_DISCONNECTED<br>* WCDMA RRC State is IDLE<br>– 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH<br>* WCDMA RRC state is CELL_PCH<br>– 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH<br>* WCDMA RRC state is URA_PCH<br>– 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH<br>* WCDMA RRC state is CELL_FACH<br>– 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH<br>* WCDMA RRC state is CELL_DCH |
| *umtsLTENbr-CellLen* | • Number of sets of UMTS LTE Neighbors. |
| *UMTSLTENbr-Cell* | • See nas_umtsLTENbrCell for more information. |

### 8.271.2 Field Documentation

#### 8.271.2.1 nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

#### 8.271.2.2 uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

#### 8.271.2.3 uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCState

## 8.272 nas_WCDMARSSIThresh Struct Reference

### Data Fields

- uint8_t WCDMARSSIThreshListLen
- int16_t * pWCDMARSSIThreshList

### 8.272.1 Detailed Description

This structure contains WCDMA RSSI threshold related parameters.

**Parameters**

| | |
|---|---|
| *WCDMARSSI-ThreshListLen* | • Length of the WCDMA RSSI threshold list parameter to follow |
| *pWCDMARSSI-ThreshList* | • Array of RSSI thresholds (in units of 0.1 dBm)<br>• Maximum of 32 values.<br>• Range for RSSI values: -121 to 0 (in dBm) |

### 8.272.2 Field Documentation

#### 8.272.2.1 int16_t* nas_WCDMARSSIThresh::pWCDMARSSIThreshList

#### 8.272.2.2 uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen

## 8.273 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

### 8.273.1 Detailed Description

Structure for storing the WCDMA System Information.

**Parameters**

| | |
|---|---|
| *sysInfoWCDMA* | • See nas_sysInfoCommon for more information. |
| *lacValid* | • Indicates whether the location area code is valid..<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |

| | |
|---|---|
| *lac* | • Location area code.<br>• Only applies to 3GPP.<br>– 0xFFFF - Not Available |
| *cellIdValid* | • Indicates whether the cell ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *cellId* | • Cell ID.<br>– 0xFFFFFFFF - Not Available |
| *regRejectInfo-Valid* | • Indicates whether the registration reject information is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *rejectSrvDomain* | • Type of service domain in which the registration is rejected.<br>– 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>– 0x04 - Camped<br>– 0xFF - Not Available |
| *rejCause* | • Reject cause values sent are specified in [3GPP TS 24.008, Section 10.5.3.6].<br>– 0xFF - Not Available |
| *networkIdValid* | • Indicates whether the network ID is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *MCC[NAS_PLM-N_LENGTH]* | • Mobile Country Code.<br>• MCC digits in ASCII characters |
| *MNC[NAS_PLM-N_LENGTH]* | • Mobile Network Code.<br>• MNC digits in ASCII characters<br>• An unused byte is set to 0xFF.<br>• 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. |

| | |
|---|---|
| *hsCallStatus-Valid* | • Indicates whether the high-speed call status is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *hsCallStatus* | • Call status on high speed.<br>• Only applicable for WCDMA.<br>– 0x00 - HSDPA and HSUPA are unsupported<br>– 0x01 - HSDPA is supported<br>– 0x02 - HSUPA is supported<br>– 0x03 - HSDPA and HSUPA are supported<br>– 0x04 - HSDPA+ is supported<br>– 0x05 - HSDPA+ and HSUPA are supported<br>– 0x06 - Dual-cell HSDPA+ is supported<br>– 0x07 - Dual-cell HSDPA+ and HSUPA are supported<br>– 0xFF - Not Available |
| *hsIndValid* | • Indicates whether high-speed service indication is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *hsInd* | • High-speed service indication<br>• Only applicable for WCDMA.<br>– 0x00 - HSDPA and HSUPA are unsupported<br>– 0x01 - HSDPA is supported<br>– 0x02 - HSUPA is supported<br>– 0x03 - HSDPA and HSUPA are supported<br>– 0x04 - HSDPA+ is supported<br>– 0x05 - HSDPA+ and HSUPA are supported<br>– 0x06 - Dual-cell HSDPA+ is supported<br>– 0x07 - Dual-cell HSDPA+ and HSUPA are supported<br>– 0xFF - Not Available |
| *pscValid* | • Indicates whether primary scrambling code is valid.<br>– 0x00 - Invalid<br>– 0x01 - Valid<br>– 0xFF - Not Available |
| *psc* | • Primary scrambling code.<br>– 0xFFFF - Not Available |

### 8.273.2 Field Documentation

#### 8.273.2.1 uint32_t nas_WCDMASysInfo::cellId

#### 8.273.2.2 uint8_t nas_WCDMASysInfo::cellIdValid

#### 8.273.2.3 uint8_t nas_WCDMASysInfo::hsCallStatus

#### 8.273.2.4 uint8_t nas_WCDMASysInfo::hsCallStatusValid

#### 8.273.2.5 uint8_t nas_WCDMASysInfo::hsInd

#### 8.273.2.6 uint8_t nas_WCDMASysInfo::hsIndValid

#### 8.273.2.7 uint16_t nas_WCDMASysInfo::lac

#### 8.273.2.8 uint8_t nas_WCDMASysInfo::lacValid

#### 8.273.2.9 uint8_t nas_WCDMASysInfo::MCC[3]

#### 8.273.2.10 uint8_t nas_WCDMASysInfo::MNC[3]

#### 8.273.2.11 uint8_t nas_WCDMASysInfo::networkIdValid

#### 8.273.2.12 uint16_t nas_WCDMASysInfo::psc

#### 8.273.2.13 uint8_t nas_WCDMASysInfo::pscValid

#### 8.273.2.14 uint8_t nas_WCDMASysInfo::regRejectInfoValid

#### 8.273.2.15 uint8_t nas_WCDMASysInfo::rejCause

#### 8.273.2.16 uint8_t nas_WCDMASysInfo::rejectSrvDomain

#### 8.273.2.17 nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA

## 8.274 nas_wcdmaUARFCN Struct Reference

### Data Fields

- uint8_t status
- uint32_t uarfcn

### 8.274.1 Detailed Description

This structure contains WCDMA UARFCN infromation.

**Parameters**

| | |
|---|---|
| *status* | • 0 - Disable<br>• 1 - Enable |
| *uarfcn* | • UARFCN to which UE is locked |

### 8.274.2 Field Documentation

#### 8.274.2.1 uint8_t nas_wcdmaUARFCN::status

#### 8.274.2.2 uint32_t nas_wcdmaUARFCN::uarfcn

## 8.275 NASBandPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t band_pref

### 8.275.1 Detailed Description

Structure used to store all Band Preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *band_pref* | - Roaming Indication Value |

### 8.275.2 Field Documentation

#### 8.275.2.1 uint64_t NASBandPreferenceTlv::band_pref

#### 8.275.2.2 uint8_t NASBandPreferenceTlv::TlvPresent

## 8.276 NASEmergencyModeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t EmerMode

### 8.276.1 Detailed Description

Structure used to store Emergency Mode TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *EmerMode* | - Emergency Mode Value |

### 8.276.2 Field Documentation

#### 8.276.2.1 uint8_t NASEmergencyModeTlv::EmerMode

#### 8.276.2.2 uint8_t NASEmergencyModeTlv::TlvPresent

## 8.277 NasGetLTECphyCaInfo Struct Reference

### Data Fields

- NASPhyCaAggScellIndType PhyCaAggScellIndType
- NASPhyCaAggScellDlBw PhyCaAggScellDlBw
- NASPhyCaAggScellInfo PhyCaAggScellInfo
- NASPhyCaAggPcellInfo PhyCaAggPcellInfo
- NASPhyCaAggScellIndex PhyCaAggScellIndex

### 8.277.1 Detailed Description

This structure contains the parameters for aggregation event information.

**Parameters**

| | |
|---|---|
| *PhyCaAggScell-IndType* | • see NASPhyCaAggScellIndType<br>• Bit to check in ParamPresenceMask - **16** |
| *PhyCaAggScell-DlBw* | • see NASPhyCaAggScellDlBw<br>• Bit to check in ParamPresenceMask - **17** |
| *PhyCaAggScell-Info* | • see NASPhyCaAggScellInfo<br>• Bit to check in ParamPresenceMask - **18** |
| *PhyCaAggPcell-Info* | • see NASPhyCaAggPcellInfo<br>• Bit to check in ParamPresenceMask - **19** |
| *PhyCaAggScell-Index* | • see NASPhyCaAggScellIndex<br>• Bit to check in ParamPresenceMask - **20** |

### 8.277.2 Field Documentation

#### 8.277.2.1 NASPhyCaAggPcellInfo NasGetLTECphyCaInfo::PhyCaAggPcellInfo

#### 8.277.2.2 NASPhyCaAggScellDlBw NasGetLTECphyCaInfo::PhyCaAggScellDlBw

#### 8.277.2.3 NASPhyCaAggScellIndex NasGetLTECphyCaInfo::PhyCaAggScellIndex

#### 8.277.2.4 NASPhyCaAggScellIndType NasGetLTECphyCaInfo::PhyCaAggScellIndType

#### 8.277.2.5 NASPhyCaAggScellInfo NasGetLTECphyCaInfo::PhyCaAggScellInfo

## 8.278 NASGWAcqOrderPrefTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t GWAcqOrderPref

### 8.278.1 Detailed Description

Structure used to store GSM/WCDMA acquisition order preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *GWAcqOrder-Pref* | - GSM/WCDMA acquisition order Preference Value |

### 8.278.2 Field Documentation

#### 8.278.2.1 uint32_t NASGWAcqOrderPrefTlv::GWAcqOrderPref

#### 8.278.2.2 uint8_t NASGWAcqOrderPrefTlv::TlvPresent

## 8.279 NASLTEBandPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t LTEBandPref

### 8.279.1 Detailed Description

Structure used to store LTE Band Preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *LTEBandPref* | - LTE Band Preference Value |

### 8.279.2 Field Documentation

#### 8.279.2.1 uint64_t NASLTEBandPreferenceTlv::LTEBandPref

#### 8.279.2.2 uint8_t NASLTEBandPreferenceTlv::TlvPresent

## 8.280 NASLteNasReleaseInfoTlv Struct Reference

### Data Fields

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

### 8.280.1 Detailed Description

This structure contains LTE Nas Release Infomation

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack result |
| *nas_release* | • LTE NAS release |
| *nas_major* | • LTE NAS version major |
| *nas_minor* | • LTE NAS version minor |

### 8.280.2 Field Documentation

#### 8.280.2.1 uint8_t NASLteNasReleaseInfoTlv::nas_major

#### 8.280.2.2 uint8_t NASLteNasReleaseInfoTlv::nas_minor

#### 8.280.2.3 uint8_t NASLteNasReleaseInfoTlv::nas_release

#### 8.280.2.4 uint8_t NASLteNasReleaseInfoTlv::TlvPresent

## 8.281 NASModePreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t ModePref

### 8.281.1 Detailed Description

Structure used to store Mode Preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *ModePref* | - Mode Preference Value |

### 8.281.2 Field Documentation

#### 8.281.2.1 uint16_t NASModePreferenceTlv::ModePref

#### 8.281.2.2 uint8_t NASModePreferenceTlv::TlvPresent

## 8.282 NASNetSelPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t NetSelPref

### 8.282.1 Detailed Description

Structure used to store Network Selection Preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *NetSelPref* | - Network Selection Preference Value |

### 8.282.2 Field Documentation

#### 8.282.2.1 uint8_t NASNetSelPreferenceTlv::NetSelPref

#### 8.282.2.2 uint8_t NASNetSelPreferenceTlv::TlvPresent

## 8.283 NASOTAMessageTlv Struct Reference

### Data Fields

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

### 8.283.1 Detailed Description

This structure contains OTA message

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack result |
| *message_type* | • message type<br>– 0 - LTE ESM uplink<br>– 1 - LTE ESM downlink<br>– 2 - LTE EMM uplink<br>– 3 - LTE EMM downlink<br>– 4 - GSM/UMTS uplink<br>– 5 - GSM/UMTS downlink |
| *data_len* | • OTA Message Content Length |
| *data_buf* | • OTA Message Content |

### 8.283.2 Field Documentation

#### 8.283.2.1 uint8_t NASOTAMessageTlv::data_buf[2048]

#### 8.283.2.2 uint16_t NASOTAMessageTlv::data_len

#### 8.283.2.3 uint32_t NASOTAMessageTlv::message_type

#### 8.283.2.4 uint8_t NASOTAMessageTlv::TlvPresent

## 8.284 NASPhyCaAggPcellInfo Struct Reference

### Data Fields

- uint32_t pci
- uint32_t freq
- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint32_t iLTEbandValue
- uint8_t TlvPresent

### 8.284.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *freq* | • Frequency of the absolute cell Range.<br>• Range for ID values: 0 to 65535. |
| *dl_bw_value* | • Downlink Bandwidth Values.<br>• See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information. |
| *scell_state* | • Scell state Values.<br>• See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.284.2 Field Documentation

#### 8.284.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggPcellInfo::dl_bw_value

#### 8.284.2.2 uint32_t NASPhyCaAggPcellInfo::freq

#### 8.284.2.3 uint32_t NASPhyCaAggPcellInfo::iLTEbandValue

#### 8.284.2.4 uint32_t NASPhyCaAggPcellInfo::pci

#### 8.284.2.5 uint8_t NASPhyCaAggPcellInfo::TlvPresent

## 8.285 NASPhyCaAggScellDlBw Struct Reference

### Data Fields

- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint8_t TlvPresent

### 8.285.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *dl_bw_value* | • Downlink Bandwidth Values.<br>• See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information. |

### 8.285.2 Field Documentation

#### 8.285.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellDlBw::dl_bw_value

#### 8.285.2.2 uint8_t NASPhyCaAggScellDlBw::TlvPresent

## 8.286 NASPhyCaAggScellIndex Struct Reference

### Data Fields

- uint8_t scell_idx
- uint8_t TlvPresent

### 8.286.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *scell_idx* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *TlvPresent* | • Tlv Present. |

### 8.286.2 Field Documentation

#### 8.286.2.1 uint8_t NASPhyCaAggScellIndex::scell_idx

#### 8.286.2.2 uint8_t NASPhyCaAggScellIndex::TlvPresent

## 8.287 NASPhyCaAggScellIndType Struct Reference

### Data Fields

- uint32_t pci
- uint32_t freq
- LITEQMI_NAS_LTE_CPHY_SCELL_STATE scell_state
- uint8_t TlvPresent

### 8.287.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *freq* | • Frequency of the absolute cell Range.<br>• Range for ID values: 0 to 65535. |
| *scell_state* | • Scell state Values.<br>• See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.287.2 Field Documentation

#### 8.287.2.1 uint32_t NASPhyCaAggScellIndType::freq

#### 8.287.2.2 uint32_t NASPhyCaAggScellIndType::pci

#### 8.287.2.3 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::scell_state

#### 8.287.2.4 uint8_t NASPhyCaAggScellIndType::TlvPresent

## 8.288 NASPhyCaAggScellInfo Struct Reference

### Data Fields

- uint32_t pci
- uint32_t freq
- LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value
- uint32_t iLTEbandValue
- LITEQMI_NAS_LTE_CPHY_SCELL_STATE scell_state
- uint8_t TlvPresent

### 8.288.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pci* | • Physical cell ID of the SCell Range.<br>• Range for ID values: 0 to 503. |
| *freq* | • Frequency of the absolute cell Range.<br>• Range for ID values: 0 to 65535. |
| *dl_bw_value* | • Downlink Bandwidth Values.<br>• See LITEQMI_NAS_LTE_CPHY_CA_BW_NRB for more information. |
| *iLTEbandValue* | • Band value.<br>• Range for LTE Band class 120 to 160. |
| *scell_state* | • Scell state Values.<br>• See LITEQMI_NAS_LTE_CPHY_SCELL_STATE for more information. |
| *TlvPresent* | • Tlv Present. |

### 8.288.2 Field Documentation

#### 8.288.2.1 LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellInfo::dl_bw_value

#### 8.288.2.2 uint32_t NASPhyCaAggScellInfo::freq

#### 8.288.2.3 uint32_t NASPhyCaAggScellInfo::iLTEbandValue

#### 8.288.2.4 uint32_t NASPhyCaAggScellInfo::pci

#### 8.288.2.5 LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellInfo::scell_state

#### 8.288.2.6 uint8_t NASPhyCaAggScellInfo::TlvPresent

## 8.289 NASPRLPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t PRLPref

### 8.289.1 Detailed Description

Structure used to store CDMA PRL Preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *PRLPref* | - CDMA PRL Preference Value |

### 8.289.2 Field Documentation

#### 8.289.2.1 uint16_t NASPRLPreferenceTlv::PRLPref

#### 8.289.2.2 uint8_t NASPRLPreferenceTlv::TlvPresent

## 8.290 NASQmiCbkNasSwiOTAMessageInd Struct Reference

### Data Fields

- NASOTAMessageTlv otaMsgTlv
- NASLteNasReleaseInfoTlv nasRelInfoTlv
- NASTimeInfoTlv timeTlv

### 8.290.1 Detailed Description

This structure contains OTA message

**Parameters**

| | |
|---|---|
| *otaMsgTlv* | • See NASOTAMessageTlv for more details<br>• Bit to check in ParamPresenceMask - **1** |
| *nasRelInfoTlv* | • See NASLteNasReleaseInfoTlv for more details<br>• Bit to check in ParamPresenceMask - **16** |
| *timeTlv* | • See NASTimeInfoTlv for more details<br>• Bit to check in ParamPresenceMask - **17** |

### 8.290.2 Field Documentation

#### 8.290.2.1 NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

#### 8.290.2.2 NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

#### 8.290.2.3 NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv

## 8.291 NASQmiCbkNasSystemSelPrefInd Struct Reference

### Data Fields

- NASEmergencyModeTlv EMTlv
- NASModePreferenceTlv MPTlv
- NASBandPreferenceTlv BPTlv

- NASPRLPreferenceTlv PRLPTlv
- NASRoamPreferenceTlv RPTlv
- NASLTEBandPreferenceTlv LBPTlv
- NASNetSelPreferenceTlv NSPTlv
- NASServDomainPrefTlv SDPTlv
- NASGWAcqOrderPrefTlv GWAOPTlv

### 8.291.1 Detailed Description

Structure used to store all QMI Notification parameters.

**Parameters**

| | |
|---|---|
| *EMTlv* | - Emergency mode Tlv<br>• See NASEmergencyModeTlv for more details<br>• Bit to check in ParamPresenceMask - **16** |
| *MPTlv* | - Mode preference Tlv<br>• See NASModePreferenceTlv for more details<br>• Bit to check in ParamPresenceMask - **17** |
| *BPTlv* | - Band preference Tlv<br>• See NASBandPreferenceTlv for more details<br>• Bit to check in ParamPresenceMask - **18** |
| *PRLPTlv* | - PRL preference Tlv<br>• See NASPRLPreferenceTlv for more details<br>• Bit to check in ParamPresenceMask - **19** |
| *RPTlv* | - Roaming preference Tlv<br>• See NASRoamPreferenceTlv for more details<br>• Bit to check in ParamPresenceMask - **20** |
| *LBPTlv* | - LTE band preference Tlv<br>• See NASLTEBandPreferenceTlv for more details<br>• Bit to check in ParamPresenceMask - **21** |
| *NSPTlv* | - Network selection preference Tlv<br>• See NASNetSelPreferenceTlv for more details<br>• Bit to check in ParamPresenceMask - **22** |
| *SDPTlv* | - Service domain preference Tlv<br>• See NASServDomainPrefTlv for more details<br>• Bit to check in ParamPresenceMask - **24** |
| *GWAOPTlv* | - GSM WCDMA Acquisition Order Preference Tlv<br>• See NASGWAcqOrderPrefTlv for more details<br>• Bit to check in ParamPresenceMask - **25** |

Note: None

### 8.291.2 Field Documentation

#### 8.291.2.1 NASBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::BPTlv

#### 8.291.2.2 NASEmergencyModeTlv NASQmiCbkNasSystemSelPrefInd::EMTlv

#### 8.291.2.3 NASGWAcqOrderPrefTlv NASQmiCbkNasSystemSelPrefInd::GWAOPTlv

#### 8.291.2.4 NASLTEBandPreferenceTlv NASQmiCbkNasSystemSelPrefInd::LBPTlv

#### 8.291.2.5 NASModePreferenceTlv NASQmiCbkNasSystemSelPrefInd::MPTlv

#### 8.291.2.6 NASNetSelPreferenceTlv NASQmiCbkNasSystemSelPrefInd::NSPTlv

#### 8.291.2.7 NASPRLPreferenceTlv NASQmiCbkNasSystemSelPrefInd::PRLPTlv

#### 8.291.2.8 NASRoamPreferenceTlv NASQmiCbkNasSystemSelPrefInd::RPTlv

#### 8.291.2.9 NASServDomainPrefTlv NASQmiCbkNasSystemSelPrefInd::SDPTlv

## 8.292 NASRoamPreferenceTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t RoamPref

### 8.292.1 Detailed Description

Structure used to store Roaming Preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *RoamPref* | - Roaming Preference Value |

### 8.292.2 Field Documentation

#### 8.292.2.1 uint16_t NASRoamPreferenceTlv::RoamPref

#### 8.292.2.2 uint8_t NASRoamPreferenceTlv::TlvPresent

## 8.293 NASServDomainPrefTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t SrvDomainPref

### 8.293.1 Detailed Description

Structure used to store Service domain preference TLV Value.

**Parameters**

| | |
|---|---|
| *TlvPresent* | - Boolean indicating the presence of the TLV in the QMI response |
| *SrvDomainPref* | - Service Domain Preference Value |

### 8.293.2 Field Documentation

#### 8.293.2.1 uint32_t NASServDomainPrefTlv::SrvDomainPref

#### 8.293.2.2 uint8_t NASServDomainPrefTlv::TlvPresent

## 8.294 NASServingSystemInfo Struct Reference

### Data Fields

- uint8_t registrationState
- uint8_t csAttachState
- uint8_t psAttachState
- uint8_t selectedNetwork
- uint8_t radioInterfaceNo
- uint8_t radioInterfaceList [255]
- uint8_t hdrPersonality

### 8.294.1 Detailed Description

This structure will hold the serving system parameters information

**Parameters**

| | |
|---|---|
| *registrationState* | - Registration state of the mobile<br>• 0 - QMI_NAS_NOT_REGISTERED Not registered;mobile is not currently searching for a new network to provide service<br>• 1 - QMI_NAS_REGISTERED Registered with a network<br>• 2 - QMI_NAS_NOT_REGISTERED_SEARCHING Not registered, but mobile is currently searching for a new network to provide service<br>• 3 - QMI_NAS_REGISTRATION_DENIED Registration denied by the visible network<br>• 4 - QMI_NAS_REGISTRATION_UNKNOWN Registration state is unknown |
| *csAttachState* | - Circuit Switch domain attach state of the mobile<br>• 0 - Unknown or not applicable<br>• 1 - Attached<br>• 2 - Detached |
| *psAttachState* | - Packet domain attach state of the mobile<br>• 0 - Unknown or not applicable<br>• 1 - Attached<br>• 2 - Detached |
| *selectedNetwork* | - Type of selected radio access network<br>• 0x00 - Unknown<br>• 0x01 - 3GPP2 network<br>• 0x02 - 3GPP network |

| | |
|---|---|
| *radioInterfaceNo* | - Number of radio interfaces currently in use; this indicates how many radio_if identifiers follow this field |
| *radioInterface-List* | - Radio interface currently in use (each is 1 byte)<br>• 0x00 - None (no service)<br>• 0x01 - cdma2000 1X<br>• 0x02 - cdma2000 HRPD (1xEV-DO)<br>• 0x03 - AMPS<br>• 0x04 - GSM<br>• 0x05 - UMTS<br>• 0x08 - LTE |
| *hdrPersonality* | - HDR personality information (valid only for EVDO)<br>• 0x00 - Unknown<br>• 0x01 - HRPD<br>• 0x02 - eHRPD |

Note: None

### 8.294.2 Field Documentation

#### 8.294.2.1 uint8_t NASServingSystemInfo::csAttachState

#### 8.294.2.2 uint8_t NASServingSystemInfo::hdrPersonality

#### 8.294.2.3 uint8_t NASServingSystemInfo::psAttachState

#### 8.294.2.4 uint8_t NASServingSystemInfo::radioInterfaceList[255]

#### 8.294.2.5 uint8_t NASServingSystemInfo::radioInterfaceNo

#### 8.294.2.6 uint8_t NASServingSystemInfo::registrationState

#### 8.294.2.7 uint8_t NASServingSystemInfo::selectedNetwork

## 8.295 NASTimeInfoTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t time

### 8.295.1 Detailed Description

This structure contains LTE Nas time Infomation

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack result |

| | |
|---|---|
| *time* | • Seconds in local time since Jan. 6th 1980 00:00:00 UTC |

### 8.295.2 Field Documentation

#### 8.295.2.1 uint64_t NASTimeInfoTlv::time

#### 8.295.2.2 uint8_t NASTimeInfoTlv::TlvPresent

## 8.296 newMTMessageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSMTMessageInfo MTMessageInfo

### 8.296.1 Detailed Description

This structure contains MT message TLV information.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *MTMessageInfo* | • MT Message<br>• See sMSMTMessageInfo for more information |

### 8.296.2 Field Documentation

#### 8.296.2.1 sMSMTMessageInfo newMTMessageTlv::MTMessageInfo

#### 8.296.2.2 uint8_t newMTMessageTlv::TlvPresent

## 8.297 pack_audio_SLQSGetAudioPathConfig_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t Item

### 8.297.1 Detailed Description

This structure contains the pack parameters for Get Audio Path Config.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-9 |
| *Item* | • Item<br>– 0 - AV_EC<br>– 1 - AV_NS<br>– 2 - AV_TXVOL<br>– 3 - AV_DTMFTXG<br>– 4 - AV_CODECSTG<br>– 5 - AV_TXPCMIIRFLTR<br>– 6 - AV_RXPCMIIRFLTR<br>– 7 - AV_MICGAIN<br>– 8 - AV_RXAGC<br>– 9 - AV_TXAGC<br>– 10 - AV_RXAGCLIST<br>– 11 - AV_RXAVCLIST<br>– 12 - AV_TXAGCLIST |

### 8.297.2 Field Documentation

#### 8.297.2.1 uint8_t pack_audio_SLQSGetAudioPathConfig_t::Item

#### 8.297.2.2 uint8_t pack_audio_SLQSGetAudioPathConfig_t::Profile

## 8.298 pack_audio_SLQSGetAudioProfile_t Struct Reference

### Data Fields

- uint8_t Generator

### 8.298.1 Detailed Description

This structure contains the pack parameters to Get Audio Profile.

**Parameters**

| | |
|---|---|
| *Generator* | • Audio Generator<br>– 0 - Voice<br>– 1 - Key Beep<br>– 2 - MIDI |

### 8.298.2 Field Documentation

#### 8.298.2.1 uint8_t pack_audio_SLQSGetAudioProfile_t::Generator

## 8.299 pack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t Generator
- uint8_t Volume
- uint8_t Item

### 8.299.1 Detailed Description

This structure contains the pack parameters for SLQSGetAudioVolTLBConfig.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-9 |
| *Generator* | • Audio Generator<br>– 0-2 |
| *Volume* | • Audio Volume Level<br>– 0-7 |
| *Item* | • Item<br>– 13 - AV_RXVOLDB<br>– 14 - AV_DTMFVOLDB<br>– 15 - AV_PAD |

### 8.299.2 Field Documentation

#### 8.299.2.1 uint8_t pack_audio_SLQSGetAudioVolTLBConfig_t::Generator

#### 8.299.2.2 uint8_t pack_audio_SLQSGetAudioVolTLBConfig_t::Item

#### 8.299.2.3 uint8_t pack_audio_SLQSGetAudioVolTLBConfig_t::Profile

#### 8.299.2.4 uint8_t pack_audio_SLQSGetAudioVolTLBConfig_t::Volume

## 8.300 pack_audio_SLQSSetAudioPathConfig_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t * pECMode
- uint8_t * pNSEnable
- uint16_t * pTXGain
- uint16_t * pDTMFTXGain

- uint16_t * pCodecSTGain
- audio_TXPCMIIRFltr * pTXPCMIIRFltr
- audio_RXPCMIIRFltr * pRXPCMIIRFltr
- uint8_t * pRXAVCAGCSwitch
- uint8_t * pTXAVCSwitch
- audio_RXAGCList * pRXAGCList
- audio_RXAVCList * pRXAVCList
- audio_TXAGCList * pTXAGCList

### 8.300.1 Detailed Description

This structure contains the pack parameters for SLQSSetAudioPathConfig .

**Parameters**

| | |
|---|---|
| *Profile* | [Mandatory]<br>• Audio Profile<br>– 0-9 |
| *pECMode* | [Optional]<br>• AV_EC<br>– 0 - Echo cancellation off<br>– 1 - Handset echo mode<br>– 2 - Headset mode<br>– 3 - Car kit mode<br>– 4 - Speaker Mode |
| *pNSEnable* | [Optional]<br>• Noise Suppression<br>– 0 - Noise suppression off<br>– 1 - Noise suppression on |
| *pTXGain* | [Optional]<br>• TX Voice volume<br>– 0x0000 - 0xffff |
| *pDTMFTXGain* | [Optional]<br>• AV_DTMFTXG<br>– 0x0000 - 0xffff |
| *pCodecSTGain* | [Optional]<br>• AV_CODECSTG<br>– 0x0000 - 0xffff |
| *pTXPCMIIRFltr* | [Optional]<br>• See audio_TXPCMIIRFltr for more information |
| *pRXPCMIIRFltr* | [Optional]<br>• See audio_RXPCMIIRFltr for more information |
| *pRXAVCAGC-Switch* | [Optional]<br>• RX AVC/AGC Switch |

| | |
|---|---|
| *pTXAVCSwitch* | [Optional]<br>• TX AVC Switch |
| *pRXAGCList* | [Optional]<br>• See audio_RXAGCList for more information |
| *pRXAVCList* | [Optional]<br>• See audio_RXAVCList for more information |
| *pTXAGCList* | [Optional]<br>• See audio_TXAGCList for more information |

### 8.300.2 Field Documentation

#### 8.300.2.1 uint16_t* pack_audio_SLQSSetAudioPathConfig_t::pCodecSTGain

#### 8.300.2.2 uint16_t* pack_audio_SLQSSetAudioPathConfig_t::pDTMFTXGain

#### 8.300.2.3 uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pECMode

#### 8.300.2.4 uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pNSEnable

#### 8.300.2.5 uint8_t pack_audio_SLQSSetAudioPathConfig_t::Profile

#### 8.300.2.6 audio_RXAGCList* pack_audio_SLQSSetAudioPathConfig_t::pRXAGCList

#### 8.300.2.7 uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pRXAVCAGCSwitch

#### 8.300.2.8 audio_RXAVCList* pack_audio_SLQSSetAudioPathConfig_t::pRXAVCList

#### 8.300.2.9 audio_RXPCMIIRFItr* pack_audio_SLQSSetAudioPathConfig_t::pRXPCMIIRFItr

#### 8.300.2.10 audio_TXAGCList* pack_audio_SLQSSetAudioPathConfig_t::pTXAGCList

#### 8.300.2.11 uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pTXAVCSwitch

#### 8.300.2.12 uint16_t* pack_audio_SLQSSetAudioPathConfig_t::pTXGain

#### 8.300.2.13 audio_TXPCMIIRFItr* pack_audio_SLQSSetAudioPathConfig_t::pTXPCMIIRFItr

## 8.301 pack_audio_SLQSSetAudioProfile_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t EarMute
- uint8_t MicMute
- uint8_t Generator
- uint8_t Volume

### 8.301.1 Detailed Description

This structure contains pack parameters to Set Audio Profile.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0 - Handset<br>– 1 - Headset<br>– 2 - Car Kit<br>– 3 - Speaker phone<br>– 4 - Auxiliary<br>– 5 - TTY<br>– 6 - Auxiliary external PCM<br>– 7 - Primary external PCM<br>– 8 - External slave PCM<br>– 9 - I2S |
| *EarMute* | • Ear Mute Setting<br>– 0 - unmuted<br>– 1 - muted |
| *MicMute* | • MIC Mute Setting<br>– 0 - unmuted<br>– 1 - muted |
| *Generator* | • Audio Generator<br>– 0 - Voice<br>– 1 - Key Beep<br>– 2 - MIDI |
| *Volume* | • Audio Volume Level<br>– 0 to 7 |

### 8.301.2 Field Documentation

#### 8.301.2.1 uint8_t pack_audio_SLQSSetAudioProfile_t::EarMute

#### 8.301.2.2 uint8_t pack_audio_SLQSSetAudioProfile_t::Generator

#### 8.301.2.3 uint8_t pack_audio_SLQSSetAudioProfile_t::MicMute

#### 8.301.2.4 uint8_t pack_audio_SLQSSetAudioProfile_t::Profile

#### 8.301.2.5 uint8_t pack_audio_SLQSSetAudioProfile_t::Volume

## 8.302 pack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t Generator
- uint8_t Volume
- uint8_t Item
- uint16_t VolValue

### 8.302.1 Detailed Description

This structure contains the pack parameters for SLQSSetAudioVolTLBConfig.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-9 |
| *Generator* | • Audio Generator<br>– 0-2 |
| *Volume* | • Audio Volume Level<br>– 0-7 |
| *Item* | • Item<br>– 13 - AV_RXVOLDB<br>– 14 - AV_DTMFVOLDB<br>– 15 - AV_PAD |
| *Value* | • Value to be set to the volume table |

### 8.302.2 Field Documentation

#### 8.302.2.1 uint8_t pack_audio_SLQSSetAudioVolTLBConfig_t::Generator

#### 8.302.2.2 uint8_t pack_audio_SLQSSetAudioVolTLBConfig_t::Item

#### 8.302.2.3 uint8_t pack_audio_SLQSSetAudioVolTLBConfig_t::Profile

#### 8.302.2.4 uint8_t pack_audio_SLQSSetAudioVolTLBConfig_t::Volume

#### 8.302.2.5 uint16_t pack_audio_SLQSSetAudioVolTLBConfig_t::VolValue

## 8.303 pack_cat_CATSendEnvelopeCommand_t Struct Reference

### Data Fields

- uint32_t cmdID
- uint32_t dataLen
- uint8_t * pData

### 8.303.1 Detailed Description

Structure used to store envelope command data.

**Parameters**

| | |
|---|---|
| *cmdID* | • Envelope command type<br>– 0x01 - Menu Selection<br>– 0x02 - Event DL User activity<br>– 0x03 - Event DL Idle Screen Available<br>– 0x04 - Event DL Language Selection |
| *dataLen* | • Length of pData in bytes |
| *pData* | • Encoded envelope data as defined in ETSI TS 102 223, section 7 [Smart Cards: Card Application Toolkit (CAT) – Release 4] |

### 8.303.2 Field Documentation

#### 8.303.2.1 uint32_t pack_cat_CATSendEnvelopeCommand_t::cmdID

#### 8.303.2.2 uint32_t pack_cat_CATSendEnvelopeCommand_t::dataLen

#### 8.303.2.3 uint8_t* pack_cat_CATSendEnvelopeCommand_t::pData

## 8.304 pack_cat_CATSendTerminalResponse_t Struct Reference

### Data Fields

- uint32_t refID
- uint32_t dataLen
- uint8_t * pData

### 8.304.1 Detailed Description

Structure used to terminal response data.

**Parameters**

| | |
|---|---|
| *refID* | • Proactive command reference ID. The value should be the same as indicated in the CAT event callback data for the relevant proactive command. |

| | |
|---|---|
| *dataLen* | • Terminal response data length |
| *pData* | • Terminal response for the relevant proactive command encoded as per ETSI TS 102 223, section 6.8<br>[Smart Cards: Card Application Toolkit (CAT) – Release 4] |

### 8.304.2 Field Documentation

#### 8.304.2.1 uint32_t pack_cat_CATSendTerminalResponse_t::dataLen

#### 8.304.2.2 uint8_t* pack_cat_CATSendTerminalResponse_t::pData

#### 8.304.2.3 uint32_t pack_cat_CATSendTerminalResponse_t::refID

## 8.305 pack_cat_SetCATEventCallback_t Struct Reference

### Data Fields

- uint32_t eventMask

### 8.305.1 Detailed Description

This structure contains sEnables/disables the CAT event callback pack variable.

**Parameters**

| | |
|---|---|
| *eventMask* | • bitmask of CAT events to register for<br>– 0x00000001 - Display Text<br>– 0x00000002 - Get In-Key<br>– 0x00000004 - Get Input<br>– 0x00000008 - Setup Menu<br>– 0x00000010 - Select Item<br>– 0x00000020 - Send SMS - Alpha Identifier<br>– 0x00000040 - Setup Event: User Activity<br>– 0x00000080 - Setup Event: Idle Screen Notify<br>– 0x00000100 - Setup Event: Language Sel Notify<br>– 0x00000200 - Setup Idle Mode Text<br>– 0x00000400 - Language Notification<br>– 0x00000800 - Refresh<br>– 0x00001000 - End Proactive Session |

### 8.305.2 Field Documentation

#### 8.305.2.1 uint32_t pack_cat_SetCATEventCallback_t::eventMask

## 8.306 pack_dms_ActivateAutomatic_t Struct Reference

### Data Fields

- uint8_t actCode [81]

### 8.306.1 Detailed Description

This structure contains UIM activate automation information

**Parameters**

| | |
|---|---|
| *actCode* | • NULL-terminated string representing activation code (maximum string length of 12).<br>• Specific carrier requirements may dictate actual activation code that is applicable, e.g., "∗22899" |

### 8.306.2 Field Documentation

#### 8.306.2.1 uint8_t pack_dms_ActivateAutomatic_t::actCode[81]

## 8.307 pack_dms_GetCustFeaturesV2_t Struct Reference

### Data Fields

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

### 8.307.1 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**

| | |
|---|---|
| *cust_id* | • Customization ID (Maximum 64 bytes)<br>• NULL terminated ASCII string. |
| *list_type* | • list type requested |
| *Tlvresult* | • Pack Result |

### 8.307.2 Field Documentation

#### 8.307.2.1 uint8_t pack_dms_GetCustFeaturesV2_t::cust_id[64+1]

#### 8.307.2.2 uint8_t pack_dms_GetCustFeaturesV2_t::list_type

#### 8.307.2.3 uint16_t pack_dms_GetCustFeaturesV2_t::Tlvresult

## 8.308 pack_dms_ResetToFactoryDefaults_t Struct Reference

### Data Fields

- uint8_t spc [6]

### 8.308.1 Detailed Description

This structure contains UIM reset to factory default information

**Parameters**

| | |
|---|---|
| *spc[IN]* | • Service programming code in ASCII format (digits 0 to 9 only) |

### 8.308.2 Field Documentation

#### 8.308.2.1 uint8_t pack_dms_ResetToFactoryDefaults_t::spc[6]

## 8.309 pack_dms_SetActivationStatusCallback_t Struct Reference

### Data Fields

- uint8_t activationState

### 8.309.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter pack.

**Parameters**

| | |
|---|---|
| *activationState* | • Service activation state.<br>• Values<br>  – 0 - Do not report<br>  – 1 - Report activation state changes |

### 8.309.2 Field Documentation

#### 8.309.2.1 uint8_t pack_dms_SetActivationStatusCallback_t::activationState

## 8.310 pack_dms_SetCrashAction_t Struct Reference

### Data Fields

- uint8_t crashAction

### 8.310.1 Detailed Description

This structure is used to store pack_dms_SetCrashAction parameters.

**Parameters**

| | |
|---|---|
| *crashAction* | • 0 - USB Memory Download. Modem will reset after a crash and will stay in USB download mode with only DM port enumerated.<br>• 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<br>• 2 - No Action |

### 8.310.2 Field Documentation

#### 8.310.2.1 uint8_t pack_dms_SetCrashAction_t::crashAction

## 8.311 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

### 8.311.1 Detailed Description

This structure contains current settings of custom features pack parameters

**Parameters**

| | |
|---|---|
| *GpsEnable* | • describes if GPS is enabled or disabled<br>• values:<br>– 0x00 - GPS is disabled<br>– 0x01 - GPS is enabled |
| *DisableIMSI* | • optional 1 byte parameter<br>• describes if IMSI display is enabled or disabled<br>• values:<br>– 0x00 - Allow display of IMSI<br>– 0x01 - Do not display IMSI |

| | |
|---|---|
| *IPFamSupport* | • optional 2 byte BitMask<br>• bitmask representing the IP families supported<br>• values:<br>– 0x01 - IPv4<br>– 0x02 - IPv6<br>– 0x04 - IPv4v6 |
| *RMAutoConnect* | • optional 1 byte parameter<br>• QMI Mode RM Net Auto Connect Support<br>• values:<br>– 0x00 - Not Supported<br>– 0x01 - Supported |
| *GPSSel* | • optional 1 byte parameter<br>• GPS Antenna Select<br>• values:<br>– 0x00 - Dedicated GPS Port<br>– 0x01 - GPS Rx over AUX Port<br>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied |
| *SMSSupport* | • optional 1 byte parameter<br>• SMS support<br>• values:<br>– 0x00 - Not supported<br>– 0x01 - supported<br>• Used to determine whether or not to hide SMS from user |
| *IsVoiceEnabled* | • optional 1 byte parameter<br>• Voice support<br>• values:<br>– 0x00 - Enable voice on both AT and QMI interface (default)<br>– 0x01 - Reserved<br>– 0x02 - Disable voice on both AT and QMI interface |
| *DHCPRelay-Enabled* | • optional 1 byte parameter<br>• DHCP Relay support<br>• values:<br>– 0x00 - Disable DHCP relay<br>– 0x01 - Enable DHCP relay |

| | |
|---|---|
| *GPSLPM* | • optional 1 byte parameter<br>• GPSLPM support<br>• values:<br>– 0x00 - Enable GPS in Low Power Mode<br>– 0x01 - Disable GPS in Low Power Mode |

### 8.311.2 Field Documentation

#### 8.311.2.1 uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled

#### 8.311.2.2 uint8_t pack_dms_SetCustFeature_t::DisableIMSI

#### 8.311.2.3 uint32_t pack_dms_SetCustFeature_t::GpsEnable

#### 8.311.2.4 uint8_t pack_dms_SetCustFeature_t::GPSLPM

#### 8.311.2.5 uint8_t pack_dms_SetCustFeature_t::GPSSel

#### 8.311.2.6 uint16_t pack_dms_SetCustFeature_t::IPFamSupport

#### 8.311.2.7 uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled

#### 8.311.2.8 uint8_t pack_dms_SetCustFeature_t::RMAutoConnect

#### 8.311.2.9 uint8_t pack_dms_SetCustFeature_t::SMSSupport

## 8.312 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

### 8.312.1 Detailed Description

This structure contains customization settings set to modem pack

**Parameters**

| | |
|---|---|
| *cust_id* | • Customization ID (Maximum 64 bytes)<br>• NULL terminated ASCII string. |
| *value_length* | • length of cust_value field |
| *cust_value* | • Customization Setting Value (Maximum 8 bytes) |

| | |
|---|---|
| *Tlvresult* | • Pack Result |

### 8.312.2 Field Documentation

#### 8.312.2.1 uint8_t pack_dms_SetCustFeaturesV2_t::cust_id[64+1]

#### 8.312.2.2 uint8_t pack_dms_SetCustFeaturesV2_t::cust_value[8+1]

#### 8.312.2.3 uint16_t pack_dms_SetCustFeaturesV2_t::Tlvresult

#### 8.312.2.4 uint16_t pack_dms_SetCustFeaturesV2_t::value_length

## 8.313 pack_dms_SetEventReport_t Struct Reference

### Data Fields

- uint8_t mode

### 8.313.1 Detailed Description

This structure is used to store pack_dms_SetEventReport parameters.

**Parameters**

| | |
|---|---|
| *mode* | • Operating Mode<br>• 0 - reset<br>• 1 - set<br>• 2 - no change |

### 8.313.2 Field Documentation

#### 8.313.2.1 uint8_t pack_dms_SetEventReport_t::mode

## 8.314 pack_dms_SetIndicationRegister_t Struct Reference

### Data Fields

- uint8_t PSMStatus
- uint8_t PSMCfgChangeInfo
- uint8_t RptIMSCapability

### 8.314.1 Detailed Description

This structure contains the DMS Register Indication request parameters

**Parameters**

| | |
|---|---|
| *PSMStatus* | • Power Save Mode Status<br>• Values<br>– 0 - Do not report<br>– 1 - Report PSM status<br>– 2 - No Change |
| *PSMCfgChange-Info* | • Power Save Mode Configuration Change Information<br>• Values<br>– 0 - Do not report<br>– 1 - Report Power Save Mode configuration changes<br>– 2 - No Change |
| *RptIMS-Capability* | • Report IMS Capability<br>• Values<br>– 0 - Do not report<br>– 1 - Report IMS capability<br>– 2 - No Change |

### 8.314.2 Field Documentation

#### 8.314.2.1 uint8_t pack_dms_SetIndicationRegister_t::PSMCfgChangeInfo

#### 8.314.2.2 uint8_t pack_dms_SetIndicationRegister_t::PSMStatus

#### 8.314.2.3 uint8_t pack_dms_SetIndicationRegister_t::RptIMSCapability

## 8.315 pack_dms_SetPower_t Struct Reference

### Data Fields

- uint32_t mode
- uint16_t Tlvresult

### 8.315.1 Detailed Description

Structure to store pack the operating mode of the device.

**Parameters**

| | |
|---|---|
| *mode* | • Selected operating mode<br>– See qaGobiApiTablePowerModes.h for power modes |
| *Tlvresult* | • pack Tlv Result. |

### 8.315.2 Field Documentation

#### 8.315.2.1 uint32_t pack_dms_SetPower_t::mode

#### 8.315.2.2 uint16_t pack_dms_SetPower_t::Tlvresult

## 8.316 pack_dms_SetUSBComp_t Struct Reference

### Data Fields

- uint8_t USBComp
- uint16_t Tlvresult

### 8.316.1 Detailed Description

This structure is used to store USB composition information pack paramters

**Parameters**

| | |
|---|---|
| *USBComp* | • Current USB Composition<br>• Values:<br>– 0..5 - Reserved (non-QMI)<br>– 6 - DM NMEA AT QMI<br>– 7 - DM NMEA AT QMI1 QMI2 QMI3<br>– 8 - DM NMEA AT MBIM<br>– 9 - MBIM<br>– 10 - NMEA MBIM<br>– 11 - DM MBIM<br>– 12 - DM NMEA MBIM<br>13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces<br>– 13 - 6 for QMI, 8 for MBIM<br>– 14 - 6 for QMI, 9 for MBIM<br>– 15 - 6 for QMI, 10 for MBIM<br>– 16 - 6 for QMI, 11 for MBIM<br>– 17 - 6 for QMI, 12 for MBIM<br>– 18 - 7 for QMI, 8 for MBIM<br>– 19 - 7 for QMI, 9 for MBIM<br>– 20 - 7 for QMI, 10 for MBIM<br>– 21 - 7 for QMI, 11 for MBIM<br>– 22 - 7 for QMI, 12 for MBIM |
| *Tlvresult* | • pack Tlv Result. |

### 8.316.2 Field Documentation

#### 8.316.2.1 uint16_t pack_dms_SetUSBComp_t::Tlvresult

#### 8.316.2.2 uint8_t pack_dms_SetUSBComp_t::USBComp

## 8.317 pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

### Data Fields

- uint8_t resetInfoInd

### 8.317.1 Detailed Description

This structure contains set registration state for different indication pack

**Parameters**

| | |
|---|---|
| *resetInfoInd[IN]* | • Values<br>– 0 - Disable<br>– 1 - Enable |

### 8.317.2 Field Documentation

#### 8.317.2.1 uint8_t pack_dms_SLQSDmsSwiIndicationRegister_t::resetInfoInd

## 8.318 pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference

### Data Fields

- uint8_t * pPsmEnableState
- uint32_t * pActiveTimer
- uint32_t * pPeriodicUpdateTimer

### 8.318.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

**Parameters**

| | |
|---|---|
| *pPsmEnable-State* | [Optional]<br>• PSM Enable State<br>• Values<br>– 0 - PSM is not enabled<br>– 1 - PSM is enabled |
| *pActiveTimer* | [Optional]<br>• Active Timer<br>• PSM active timer value (in seconds).<br>• ActiveTimer encoding is the GPRS timer 2 encoding specified in Table 10.5.172 of 3GPP TS 24.008. |

| | |
|---|---|
| *pPeriodic-UpdateTimer* | [Optional]<br>• Periodic Update Timer<br>• PSM periodic update timer value (in seconds)<br>• Periodic Update Timer encoding is the GPRS timer 3 encoding specified in Table 10.5.163a of 3GPP TS 24.008. |

### 8.318.2 Field Documentation

#### 8.318.2.1 uint32_t∗ pack_dms_SLQSSetPowerSaveModeConfig_t::pActiveTimer

#### 8.318.2.2 uint32_t∗ pack_dms_SLQSSetPowerSaveModeConfig_t::pPeriodicUpdateTimer

#### 8.318.2.3 uint8_t∗ pack_dms_SLQSSetPowerSaveModeConfig_t::pPsmEnableState

## 8.319 pack_dms_SLQSSwiGetCrashInfo_t Struct Reference

### Data Fields

- uint8_t clear

### 8.319.1 Detailed Description

This structure contains pack_dms_SLQSSwiGetCrashInfo parameters

**Parameters**

| | |
|---|---|
| *clear* | • 0 - do not clear crash data after response<br>• 1 - clear crash data after response |

### 8.319.2 Field Documentation

#### 8.319.2.1 uint8_t pack_dms_SLQSSwiGetCrashInfo_t::clear

## 8.320 pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

### Data Fields

- uint8_t ∗ pDestSMSNum
- uint8_t ∗ pDestSMSContent

### 8.320.1 Detailed Description

This structure is used to store pack_dms_SLQSSwiSetDyingGaspCfg parameters

**Parameters**

| | |
|---|---|
| *pDestSMSNum[IN]* | • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.<br>• Optional parameter. |
| *pDestSMSContent[IN]* | • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars.<br>• Optional parameter. |

### 8.320.2 Field Documentation

#### 8.320.2.1 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSContent

#### 8.320.2.2 uint8_t* pack_dms_SLQSSwiSetDyingGaspCfg_t::pDestSMSNum

## 8.321 pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

### Data Fields

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

### 8.321.1 Detailed Description

This structure contains SWI set host device info unpack information

**Parameters**

| | |
|---|---|
| *manString[IN]* | • Host Device Manufacturer Name<br>• Null terminated ASCII String |
| *modelString[IN]* | • Host Device Model String<br>• Null terminated ASCII string. |
| *swVerString[IN]* | • Host Device Software Version String<br>• Null terminated ASCII string |
| *plasmaIDString[IN]* | • Host Device Plasma ID String<br>• Null terminated alphanumeric ASCII string |

### 8.321.2 Field Documentation

#### 8.321.2.1 char pack_dms_SLQSSwiSetHostDevInfo_t::manString[255]

#### 8.321.2.2 char pack_dms_SLQSSwiSetHostDevInfo_t::modelString[255]

#### 8.321.2.3 char pack_dms_SLQSSwiSetHostDevInfo_t::plasmaIDString[255]

#### 8.321.2.4 char pack_dms_SLQSSwiSetHostDevInfo_t::swVerString[255]

## 8.322 pack_dms_SLQSSwiSetOSInfo_t Struct Reference

### Data Fields

- char nameString [255]
- char versionString [255]

### 8.322.1 Detailed Description

This structure contains SWI set host OS info pack information

**Parameters**

| | |
|---|---|
| *nameString[IN]* | Host device manufacture String<br>• NULL terminated ASCII string. |
| *VersionString[IN]* | Host device model String<br>• NULL terminated ASCII string. |

### 8.322.2 Field Documentation

#### 8.322.2.1 char pack_dms_SLQSSwiSetOSInfo_t::nameString[255]

#### 8.322.2.2 char pack_dms_SLQSSwiSetOSInfo_t::versionString[255]

## 8.323 pack_dms_SwiSetEventReport_t Struct Reference

### Data Fields

- uint8_t * pTempReport
- uint8_t * pVoltReport

### 8.323.1 Detailed Description

This structure is used to store pack_dms_SwiSetEventReport parameters.

**Parameters**

| | |
|---|---|
| *pTempReport* | (optional)<br>• Temperature indicator.<br>– 0 - Do not report<br>– 1 - Report when state changes |

| | |
|---|---|
| *pVoltReport* | (optional)<br>• Voltage Indicator.<br>– 0 - Do not report<br>– 1 - Report when state changes |

### 8.323.2 Field Documentation

#### 8.323.2.1 uint8_t* pack_dms_SwiSetEventReport_t::pTempReport

#### 8.323.2.2 uint8_t* pack_dms_SwiSetEventReport_t::pVoltReport

## 8.324 pack_dms_UIMChangePIN_t Struct Reference

### Data Fields

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

### 8.324.1 Detailed Description

This structure contains UIM Unblock PIN Information

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *oldValue[IN]* | • Old PIN value of PIN to change |
| *newValue[IN]* | • New PIN value of PIN to change |

### 8.324.2 Field Documentation

#### 8.324.2.1 uint8_t pack_dms_UIMChangePIN_t::id

#### 8.324.2.2 uint8_t pack_dms_UIMChangePIN_t::newValue[255]

#### 8.324.2.3 uint8_t pack_dms_UIMChangePIN_t::oldValue[255]

## 8.325 pack_dms_UIMGetControlKeyStatus_t Struct Reference

### Data Fields

- uint8_t facility

### 8.325.1 Detailed Description

This structure contains UIM get control key status information

**Parameters**

| | |
|---|---|
| *facility[IN]* | • MT or network facility<br>– 0 - Network Personalization (PN)<br>– 1 - Network Subset Personalization (PU)<br>– 2 - Service Provider Personalization (PP)<br>– 3 - Corporate Personalization (PC)<br>– 4 - UIM Personalization (PF) |

### 8.325.2 Field Documentation

#### 8.325.2.1 uint8_t pack_dms_UIMGetControlKeyStatus_t::facility

## 8.326 pack_dms_UIMGetICCID_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.326.1 Detailed Description

This structure contains UIM Get ICCID pack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Pack result. |

### 8.326.2 Field Documentation

#### 8.326.2.1 swi_uint256_t pack_dms_UIMGetICCID_t::ParamPresenceMask

#### 8.326.2.2 uint16_t pack_dms_UIMGetICCID_t::Tlvresult

## 8.327 pack_dms_UIMSetControlKeyProtection_t Struct Reference

### Data Fields

- uint8_t facility
- uint8_t facilityState
- uint8_t facilityCk [8]

### 8.327.1 Detailed Description

This structure contains UIM Set control key protection information

**Parameters**

| | |
|---|---|
| *facility[IN]* | • MT or network facility<br>– 0 - Network Personalization (PN)<br>– 1 - Network Subset Personalization (PU)<br>– 2 - Service Provider Personalization (PP)<br>– 3 - Corporate Personalization (PC)<br>– 4 - UIM Personalization (PF) |
| *facilityState[IN]* | • UIM facility state<br>– 0 - Deactivated<br>– 1 - Activated<br>– 2 - Blocked |
| *facilityCk[IN]* | • Facility depersonalization control key, string in ASCII text (maximum 8 bytes) |

### 8.327.2 Field Documentation

#### 8.327.2.1 uint8_t pack_dms_UIMSetControlKeyProtection_t::facility

#### 8.327.2.2 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityCk[8]

#### 8.327.2.3 uint8_t pack_dms_UIMSetControlKeyProtection_t::facilityState

## 8.328 pack_dms_UIMSetPINProtection_t Struct Reference

### Data Fields

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

### 8.328.1 Detailed Description

This structure contains PIN Protection Information

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *bEnable[IN]* | • Enable/disable PIN protection, 0 = Disable |

| | |
|---|---|
| *pValue[IN]* | • PIN value of the PIN to be enabled/disabled |

### 8.328.2 Field Documentation

#### 8.328.2.1 uint8_t pack_dms_UIMSetPINProtection_t::bEnable

#### 8.328.2.2 uint8_t pack_dms_UIMSetPINProtection_t::id

#### 8.328.2.3 uint8_t pack_dms_UIMSetPINProtection_t::value[255]

## 8.329 pack_dms_UIMUnblockControlKey_t Struct Reference

### Data Fields

- uint8_t facility
- uint8_t facilityCk [8]

### 8.329.1 Detailed Description

This structure contains UIM unblock Control Key information

**Parameters**

| | |
|---|---|
| *facility[IN]* | • MT or network facility<br>– 0 - Network Personalization (PN)<br>– 1 - Network Subset Personalization (PU)<br>– 2 - Service Provider Personalization (PP)<br>– 3 - Corporate Personalization (PC)<br>– 4 - UIM Personalization (PF) |
| *facilityCk[IN]* | • Facility depersonalization control key, string in ASCII text (maximum 8 bytes) |

### 8.329.2 Field Documentation

#### 8.329.2.1 uint8_t pack_dms_UIMUnblockControlKey_t::facility

#### 8.329.2.2 uint8_t pack_dms_UIMUnblockControlKey_t::facilityCk[8]

## 8.330 pack_dms_UIMUnblockPIN_t Struct Reference

### Data Fields

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

### 8.330.1 Detailed Description

This structure contains UIM Unblock PIN Information

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *pukValue[IN]* | • PUK value of PIN to be unblocked |
| *newPin[IN]* | • New PIN value for the PIN to be unblocked |

### 8.330.2 Field Documentation

#### 8.330.2.1 uint8_t pack_dms_UIMUnblockPIN_t::id

#### 8.330.2.2 uint8_t pack_dms_UIMUnblockPIN_t::newPin[255]

#### 8.330.2.3 uint8_t pack_dms_UIMUnblockPIN_t::pukValue[255]

## 8.331 pack_dms_UIMVerifyPIN_t Struct Reference

### Data Fields

- uint8_t id
- uint8_t value [255]

### 8.331.1 Detailed Description

This structure contains PIN Value Information

**Parameters**

| | |
|---|---|
| *id[IN]* | • PIN ID<br>– 1 ( PIN1 / CHV1 )<br>– 2 ( PIN2 / CHV2 ) |
| *value[IN]* | • PIN value of the PIN to be enabled/disabled |

### 8.331.2 Field Documentation

#### 8.331.2.1 uint8_t pack_dms_UIMVerifyPIN_t::id

#### 8.331.2.2 uint8_t pack_dms_UIMVerifyPIN_t::value[255]

## 8.332 pack_fms_GetImagesPreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.332.1 Detailed Description

This structure contains the Get Image Preference information pack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Pack result |

### 8.332.2 Field Documentation

#### 8.332.2.1 uint16_t pack_fms_GetImagesPreference_t::Tlvresult

## 8.333 pack_fms_GetStoredImages_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.333.1 Detailed Description

This structure contains the Get Stored Images pack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Pack result |

### 8.333.2 Field Documentation

#### 8.333.2.1 uint16_t pack_fms_GetStoredImages_t::Tlvresult

## 8.334 pack_fms_SetImagesPreference_t Struct Reference

### Data Fields

- uint32_t imageListSize
- FMSPrefImageList * pImageList
- uint32_t bForceDownload
- uint8_t modemindex
- uint16_t Tlvresult

### 8.334.1 Detailed Description

This structure contains the Set Images Preference pack

**Parameters**

| | |
|---|---|
| *imageListSize* | • Image List Size |
| *pImageList* | • Image List<br>• See FMSPrefImageList |
| *bForceDownload* | • 0 - Not Force Donwload.<br>• 1 - Focrce Download. |
| *modemindex* | • Modem Index. |
| *Tlvresult* | • Unpack result |

### 8.334.2 Field Documentation

#### 8.334.2.1 uint32_t pack_fms_SetImagesPreference_t::bForceDownload

#### 8.334.2.2 uint32_t pack_fms_SetImagesPreference_t::imageListSize

#### 8.334.2.3 uint8_t pack_fms_SetImagesPreference_t::modemindex

#### 8.334.2.4 FMSPrefImageList* pack_fms_SetImagesPreference_t::pImageList

#### 8.334.2.5 uint16_t pack_fms_SetImagesPreference_t::Tlvresult

## 8.335 pack_ims_SLQSImsConfigIndicationRegister_t Struct Reference

### Data Fields

- uint8_t * pSIPConfigEvents
- uint8_t * pRegMgrConfigEvents
- uint8_t * pSMSConfigEvents
- uint8_t * pUserConfigEvents
- uint8_t * pVoIPConfigEvents

### 8.335.1 Detailed Description

This structure contains parameters of IMS Config Indication Register

**Parameters**

| | |
|---|---|
| *pSIPConfig-Events(optional)* | • Registration Indication For SIP Configuration Events.<br>• When this registration is enabled, the device learns of SIP config events via the QMI_IMS_SIP-_CONFIG_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pRegMgrConfig-Events(optional)* | • Registration Indication For Registration Manager Configuration Events.<br>• When this registration is enabled, the device learns of Reg Mgr config events via the QMI_IMS-_REG_MGR_CONFIG_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSMSConfig-Events(optional)* | • Registration Indication For SMS Configuration Events.<br>• When this registration is enabled, the device learns of SMS config events via the QMI_IMS_S-MS_CONFIG_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pUserConfig-Events(optional)* | • Registration Indication For User Configuration Events.<br>• When this registration is enabled, the device learns of user config events via the QMI_IMS_US-ER_CONFIG_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pVoIPConfig-Events(optional)* | • Registration Indication For VoIP Configuration Events.<br>• When this registration is enabled, the device learns of VOIP config events via the QMI_IMS_V-OIP_CONFIG_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |

**Note**

One of the optional parameter is mandatory to be present in the request.

### 8.335.2 Field Documentation

#### 8.335.2.1 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pRegMgrConfigEvents

#### 8.335.2.2 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pSIPConfigEvents

#### 8.335.2.3 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pSMSConfigEvents

#### 8.335.2.4 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pUserConfigEvents

#### 8.335.2.5 uint8_t* pack_ims_SLQSImsConfigIndicationRegister_t::pVoIPConfigEvents

## 8.336 pack_ims_SLQSSetIMSSMSConfig_t Struct Reference

### Data Fields

- uint8_t * pSMSFormat
- uint8_t * pSMSOverIPNwInd
- uint8_t * pPhoneCtxtURILen
- uint8_t * pPhoneCtxtURI

### 8.336.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig pack parameters.

**Parameters**

| | |
|---|---|
| *pSMSFormat* | • SMS format<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *pSMSOverIPNw-Ind* | • SMS over IP Network Indication Flag<br>– TRUE - Turn on mobile-originated SMS<br>– FALSE - Turn off mobile-originated SMS |
| *pPhoneCtxtURI-Len* | • Length of Phone context Universal Resource Identifier to follow |
| *pPhoneCtxtURI* | • Phone context universal resource identifier<br>• Length of this string must be specified in pPhoneCtxtURILen parameter |

### 8.336.2 Field Documentation

#### 8.336.2.1 uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pPhoneCtxtURI

#### 8.336.2.2 uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pPhoneCtxtURILen

#### 8.336.2.3 uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pSMSFormat

#### 8.336.2.4 uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pSMSOverIPNwInd

## 8.337 pack_ims_SLQSSetIMSUserConfig_t Struct Reference

### Data Fields

- uint8_t * pIMSDomainLen
- uint8_t * pIMSDomain

### 8.337.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig pack parameters.

**Parameters**

| | |
|---|---|
| *pIMSDomainLen* | • Length of IMS Domain Name to follow |
| *pIMSDomain* | • IMS domain name |

### 8.337.2 Field Documentation

#### 8.337.2.1 uint8_t∗ pack_ims_SLQSSetIMSUserConfig_t::pIMSDomain

#### 8.337.2.2 uint8_t∗ pack_ims_SLQSSetIMSUserConfig_t::pIMSDomainLen

## 8.338 pack_ims_SLQSSetIMSVoIPConfig_t Struct Reference

### Data Fields

- uint16_t ∗ pSessionExpiryTimer
- uint16_t ∗ pMinSessionExpiryTimer
- uint8_t ∗ pAmrWbEnable
- uint8_t ∗ pScrAmrEnable
- uint8_t ∗ pScrAmrWbEnable
- uint8_t ∗ pAmrMode
- uint16_t ∗ pAmrWBMode
- uint8_t ∗ pAmrOctetAligned
- uint8_t ∗ pAmrWBOctetAligned
- uint16_t ∗ pRingingTimer
- uint16_t ∗ pRingBackTimer
- uint16_t ∗ pRTPRTCPInactTimer

### 8.338.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig pack parameters.

**Parameters**

| | |
|---|---|
| *pSessionExpiry-Timer* | • Session duration, in seconds |
| *pMinSession-ExpiryTimer* | • Minimum allowed value for session expiry timer, in seconds |
| *pAmrWbEnable* | • Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio<br>• Values:<br>– True - Enable<br>– False - Disable |

| | |
|---|---|
| *pScrAmrEnable* | • Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB)<br>• Values:<br>– True - Enable<br>– False - Disable |
| *pScrAmrWb-Enable* | • Flag to enable/disable SCR for AMR WB Audio<br>• Values:<br>– True - Enable<br>– False - Disable |
| *pAmrMode* | • BitMask for AMR NB modes allowed<br>• Values:<br>– 0x1 - 4.75 kbps<br>– 0x2 - 5.15 kbps<br>– 0x4 - 5.9 kbps<br>– 0x8 - 6.17 kbps<br>– 0x10 - 7.4 kbps<br>– 0x20 - 7.95 kbps<br>– 0x40 - 10.2 kbps<br>– 0x80 - 12.2 kbps |
| *pAmrWBMode* | • BitMask for AMR WB modes allowed<br>• Values:<br>– 0x1 - 6.60 kbps<br>– 0x2 - 8.85 kbps<br>– 0x4 - 12.65 kbps<br>– 0x8 - 14.25 kbps<br>– 0x10 - 15.85 kbps<br>– 0x20 - 18.25 kbps<br>– 0x40 - 19.85 kbps<br>– 0x80 - 23.05 kbps<br>– 0x100 - 23.85 kbps |
| *pAmrOctet-Aligned* | • Flag to indicate if the octet is aligned for AMR NB Audio<br>• Values:<br>– True - Aligned<br>– False - Not aligned, Bandwidth Efficient mode |
| *pAmrWBOctet-Aligned* | • Flag to indicate if the octet is aligned for AMR WB Audio<br>• Values:<br>– True - Aligned<br>– False - Not aligned, Bandwidth Efficient mode |

| | |
|---|---|
| *pRingingTimer* | • Duration of ringing timer, in seconds. The ringing timer starts on the ringing event. If the call is not answered within the duration of this timer, the call is disconnected. |
| *pRingBackTimer* | • Duration of ringback timer, in seconds. The ringback timer starts on the ringback event. If the call is not answered within the duration of this timer, the call is disconnected. |
| *pRTPRTCP-InactTimer* | • Duration of RTP/RTCP inactivity timer, in seconds. If no RTP/RTCP packet is received prior to the expiry of this timer, the call is disconnected. |

### 8.338.2 Field Documentation

#### 8.338.2.1 uint8_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pAmrMode

#### 8.338.2.2 uint8_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pAmrOctetAligned

#### 8.338.2.3 uint8_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pAmrWbEnable

#### 8.338.2.4 uint16_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pAmrWBMode

#### 8.338.2.5 uint8_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pAmrWBOctetAligned

#### 8.338.2.6 uint16_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pMinSessionExpiryTimer

#### 8.338.2.7 uint16_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pRingBackTimer

#### 8.338.2.8 uint16_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pRingingTimer

#### 8.338.2.9 uint16_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pRTPRTCPInactTimer

#### 8.338.2.10 uint8_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pScrAmrEnable

#### 8.338.2.11 uint8_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pScrAmrWbEnable

#### 8.338.2.12 uint16_t∗ pack_ims_SLQSSetIMSVoIPConfig_t::pSessionExpiryTimer

## 8.339 pack_ims_SLQSSetRegMgrConfig_t Struct Reference

### Data Fields

- uint16_t ∗ pPriCSCFPort
- uint8_t ∗ pCSCFPortNameLen
- uint8_t ∗ pCSCFPortName
- uint8_t ∗ pIMSTestMode

### 8.339.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig pack parameters.

**Parameters**

| | |
|---|---|
| *pPriCSCFPort* | • Primary call session control function port |
| *pCSCFPort-NameLen* | • Length of the CSCF Port name parameter to follow |
| *pCSCFPort-Name* | • Call Session control port, fully qualified domain name<br>• Length of this string must be specified in pCSCFPortNameLen parameter |
| *pIMSTestMode* | • IMS Test mode Enabled.<br>– TRUE - Enable, no IMS registration<br>– FALSE - Disable, IMS registration is initiated |

### 8.339.2 Field Documentation

#### 8.339.2.1 uint8_t∗ pack_ims_SLQSSetRegMgrConfig_t::pCSCFPortName

#### 8.339.2.2 uint8_t∗ pack_ims_SLQSSetRegMgrConfig_t::pCSCFPortNameLen

#### 8.339.2.3 uint8_t∗ pack_ims_SLQSSetRegMgrConfig_t::pIMSTestMode

#### 8.339.2.4 uint16_t∗ pack_ims_SLQSSetRegMgrConfig_t::pPriCSCFPort

## 8.340 pack_ims_SLQSSetSIPConfig_t Struct Reference

### Data Fields

- uint16_t ∗ pSIPLocalPort
- uint32_t ∗ pTimerSIPReg
- uint32_t ∗ pSubscribeTimer
- uint32_t ∗ pTimerT1
- uint32_t ∗ pTimerT2
- uint32_t ∗ pTimerTf
- uint8_t ∗ pSigCompEnabled

### 8.340.1 Detailed Description

This structure contains the SLQSSetSIPConfig pack parameters.

**Parameters**

| | |
|---|---|
| *pSIPLocalPort* | • Primary call session control function SIP port number |
| *pTimerSIPReg* | • Initial SIP registration duration from the User equipment, in seconds |
| *pSubscribeTimer* | • Duration of the subscription by the UE for IMS registration notifications, in seconds |

| | |
|---|---|
| *pTimerT1* | • RTT estimate, in milliseconds |
| *pTimerT2* | • The maximum retransmit interval for non-invite requests and invite responses, in milliseconds |
| *pTimerTf* | • Non-invite transaction timeout timer, in milliseconds |
| *pSigComp-Enabled* | • Sig Comp Status<br>– TRUE - Enable<br>– FALSE - Disable |

### 8.340.2 Field Documentation

#### 8.340.2.1 uint8_t* pack_ims_SLQSSetSIPConfig_t::pSigCompEnabled

#### 8.340.2.2 uint16_t* pack_ims_SLQSSetSIPConfig_t::pSIPLocalPort

#### 8.340.2.3 uint32_t* pack_ims_SLQSSetSIPConfig_t::pSubscribeTimer

#### 8.340.2.4 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerSIPReg

#### 8.340.2.5 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerT1

#### 8.340.2.6 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerT2

#### 8.340.2.7 uint32_t* pack_ims_SLQSSetSIPConfig_t::pTimerTf

## 8.341 pack_imsa_SLQSRegisterIMSAIndication_t Struct Reference

### Data Fields

- int has_RegStatusConfig
- uint8_t RegStatusConfig
- int has_ServiceStatusConfig
- uint8_t ServiceStatusConfig
- int has_RatHandoverStatusConfig
- uint8_t RatHandoverStatusConfig
- int has_PdpStatusConfig
- uint8_t PdpStatusConfig

### 8.341.1 Detailed Description

This structure contains parameters of IMSA Config Indication Register

**Parameters**

| | |
|---|---|
| *RegStatus-Config(optional)* | • Register Indication For Registration status.<br>• When this registration is enabled, the device learns of Registration status via the QMI_IMSA_-REGISTRATION_STATUS_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *ServiceStatus-Config(optional)* | • Register Indication For Service status Events.<br>• When this registration is enabled, the device learns of Service status via the QMI_IMSA_SER-VICE_STATUS_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *RatHandover-Status-Config(optional)* | • Registration Indication For RAT handover status.<br>• When this registration is enabled, the device learns of RAT handover status via the QMI_IMSA-_RAT_HANDOVER_STATUS_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *PdpStatus-Config(optional)* | • PDP Status Configuration.<br>– 0x00 - Disable<br>– 0x01 - Enable |

**Note**

One of the optional parameter is mandatory to be present in the request.

### 8.341.2 Field Documentation

#### 8.341.2.1 int pack_imsa_SLQSRegisterIMSAIndication_t::has_PdpStatusConfig

#### 8.341.2.2 int pack_imsa_SLQSRegisterIMSAIndication_t::has_RatHandoverStatusConfig

#### 8.341.2.3 int pack_imsa_SLQSRegisterIMSAIndication_t::has_RegStatusConfig

#### 8.341.2.4 int pack_imsa_SLQSRegisterIMSAIndication_t::has_ServiceStatusConfig

#### 8.341.2.5 uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::PdpStatusConfig

#### 8.341.2.6 uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::RatHandoverStatusConfig

#### 8.341.2.7 uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::RegStatusConfig

#### 8.341.2.8 uint8_t pack_imsa_SLQSRegisterIMSAIndication_t::ServiceStatusConfig

## 8.342 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

### 8.342.1 Detailed Description

This structure contains LOC delete assist data pack

**Parameters**

| | |
|---|---|
| *pSVInfo* | • Pointer to struct loc_SVInfo. See loc_SVInfo for more information |
| *pGnssData* | • Pointer to struct loc_GnssData. See loc_GnssData for more information |
| *pCellDb* | • Pointer to struct loc_CellDb. See loc_CellDb for more information |
| *pClkInfo* | • Pointer to struct loc_ClkInfo. See loc_ClkInfo for more information |
| *pBdsSVInfo* | • Pointer to struct loc_BdsSVInfo. See loc_BdsSVInfo for more information |
| *Tlvresult* | • Pack delete assist data request result. |

### 8.342.2 Field Documentation

#### 8.342.2.1 loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo

#### 8.342.2.2 loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb

#### 8.342.2.3 loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo

#### 8.342.2.4 loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData

#### 8.342.2.5 loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo

#### 8.342.2.6 uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult

## 8.343 pack_loc_EventRegister_t Struct Reference

### Data Fields

- uint64_t eventRegister
- uint16_t Tlvresult

### 8.343.1 Detailed Description

This structure contains the Parameter for RegisterEvents

**Parameters**

| | |
|---|---|
| *eventRegister* | • Specifies the events that the control point is interested in receiving. -Values<br>– 0x00000001 - to receive position report event indications<br>– 0x00000002 - to receive satellite report event indications.These reports are sent at a 1 Hz rate.<br>– 0x00000004 - to receive NMEA reports for position and satellites in view.The report is at a 1 Hz rate.<br>– 0x00000008 - to receive NI Notify/Verify request event indications<br>– 0x00000010 - to receive time injection request event indications.<br>– 0x00000020 - to receive predicted orbits request event indications.<br>– 0x00000040 - to receive position injection request event indications.<br>– 0x00000080 - to receive engine state report event indications.<br>– 0x00000100 - to receive fix session status report event indications.<br>– 0x00000200 - to receive Wi-Fi position request event indications.<br>– 0x00000400 - to receive notifications from the location engine indicating its readiness to accept data from the sensors(accelerometer, gyroscope, etc.).<br>– 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.<br>– 0x00001000 - to receive Stationary Position Indicator (SPI) streaming report indications.<br>– 0x00002000 - to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.<br>– 0x00004000 - to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.<br>– 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.<br>– 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.<br>– 0x00020000 - to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.<br>– 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.<br>– 0x00080000 - to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.<br>– 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.<br>– 0x00200000 - to receive Wi-Fi Access Point (AP) data inject request event indications.<br>– 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.<br>– 0x00800000 - to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).<br>– 0x01000000 - to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.).<br>– 0x02000000 - to receive satellite position reports as polynomials.Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_-CONSTELL_REPORT_CONFIG. |

**Note**

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

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Pack result. |

### 8.343.2 Field Documentation

#### 8.343.2.1 uint64_t pack_loc_EventRegister_t::eventRegister

#### 8.343.2.2 uint16_t pack_loc_EventRegister_t::Tlvresult

## 8.344 pack_loc_SetExtPowerState_t Struct Reference

### Data Fields

- uint32_t extPowerState
- uint16_t Tlvresult

### 8.344.1 Detailed Description

This structure contains the Parameter External Power Source State pack.

**Parameters**

| | |
|---|---|
| *extPowerState* | • Specifies the Power state; injected by the control point.<br>• Values<br>– 0 - Device is not connected to an external power source<br>– 1 - Device is connected to an external power source<br>– 2 - Unknown external power state |
| *Tlvresult* | • Pack result. |

### 8.344.2 Field Documentation

#### 8.344.2.1 uint32_t pack_loc_SetExtPowerState_t::extPowerState

#### 8.344.2.2 uint16_t pack_loc_SetExtPowerState_t::Tlvresult

## 8.345 pack_loc_SetOperationMode_t Struct Reference

### Data Fields

- uint32_t mode
- uint16_t Tlvresult

### 8.345.1 Detailed Description

This structure contains Set Operation Mode pack

**Parameters**

| | |
|---|---|
| *mode* | • Valid values:<br>– eQMI_LOC_OPER_MODE_DEFAULT (1) - Use the default engine mode<br>– eQMI_LOC_OPER_MODE_MSB (2) - Use the MS-based mode<br>– eQMI_LOC_OPER_MODE_MSA (3) - Use the MS-assisted mode<br>– eQMI_LOC_OPER_MODE_STANDALONE (4) - Use Standalone mode<br>– eQMI_LOC_OPER_MODE_CELL_ID (5) - Use cell ID; this mode is only valid for GSM/UMTS networks<br>– eQMI_LOC_OPER_MODE_WWAN (6) - Use WWAN measurements to calculate the position; if this mode is set, AFLT will be used for 1X networks and OTDOA will be used for LTE networks |
| *Tlvresult* | • Pack result. |

### 8.345.2 Field Documentation

#### 8.345.2.1 uint32_t pack_loc_SetOperationMode_t::mode

#### 8.345.2.2 uint16_t pack_loc_SetOperationMode_t::Tlvresult

## 8.346 pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

### Data Fields

- uint32_t xid
- uint16_t Tlvresult

### 8.346.1 Detailed Description

This structure contains Set Operation Mode pack

**Parameters**

| | |
|---|---|
| *xid* | • Identifies the transaction.<br>• The transaction ID is returned in the Get Best Available Position indication. |
| *Tlvresult* | • Pack result. |

### 8.346.2 Field Documentation

#### 8.346.2.1 uint16_t pack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

#### 8.346.2.2 uint32_t pack_loc_SLQSLOCGetBestAvailPos_t::xid

## 8.347 pack_loc_SLQSLOCGetServer_t Struct Reference

### Data Fields

- uint32_t serverType
- int has_serverAddrTypeMask
- uint8_t serverAddrTypeMask

### 8.347.1 Detailed Description

This structure contains parameters to Gets the location server. Please check has_<Param_Name> field for presence of optional parameters

**Parameters**

| | |
|---|---|
| *serverType* | • Type of server.<br>• Valid values<br>– eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE<br>– eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC<br>– eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP<br>– eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4)- Server type is custom PDE |
| *serverAddrTypeMask* | • Optional parameter<br>• Type of address the client wants. If unspecified, the indication will contain all the types of addresses it has for the specified server type.<br>• Valid bitmasks<br>– 0x01 - IPv4<br>– 0x02 - IPv6<br>– 0x04 - URL |

### 8.347.2 Field Documentation

#### 8.347.2.1 int pack_loc_SLQSLOCGetServer_t::has_serverAddrTypeMask

#### 8.347.2.2 uint8_t pack_loc_SLQSLOCGetServer_t::serverAddrTypeMask

#### 8.347.2.3 uint32_t pack_loc_SLQSLOCGetServer_t::serverType

## 8.348 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_vertRelicability
- 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

### 8.348.1 Detailed Description

This structure contains LOC Inject Position parameters Please check has_<Param_Name> field for presence of optional parameters

**Parameters**

| | |
|---|---|
| *latitude* | • Optional parameter<br>• Type - Floating point<br>• Units - Degrees<br>• Range - -90.0 to 90.0<br>• Positive values indicate northern latitude<br>• Negative values indicate southern latitude<br>• Note - This field must be specified together with pLongitude and pHorUncCircular. |
| *longitude* | • Optional parameter<br>• Type - Floating point<br>• Units - Degrees<br>• Range - -180.0 to 180.0<br>• Positive values indicate eastern latitude<br>• Negative values indicate western latitude<br>• Note - This field must be specified together with pLatitude and pHorUncCircular. |

| | |
|---|---|
| *horUncCircular* | • Optional parameter<br>• Horizontal position uncertainty.<br>• Units - Meters<br>• Note - This field must be specified together with pLatitude and pLongitude. |
| *horConfidence* | • Optional parameter<br>• Horizontal confidence.<br>• Units - Percent<br>• Values<br>  – Valid Values - 1 to 99<br>  – Invalid Values - 0, 101 to 255<br>  – If 100 is received, reinterpret to 99<br>• Note - This field must be specified together with horizontal uncertainty. If not specified when pHorUncCircular is set, the default value is 50. |
| *horReliability* | • Optional parameter<br>  – Values<br>  * 0 - Location reliability is not set.<br>  * 1 - Location reliability is very low; use it at your own risk<br>  * 2 - Location reliability is low; little or no cross-checking is possible.<br>  * 3 - Location reliability is medium; limited cross-check passed<br>  * 4 - Location reliability is high; strong cross-check passed |
| *altitudeWrt-Ellipsoid* | • Optional parameter<br>• Altitude With Respect to Ellipsoid.<br>• Units - Meters<br>• Values<br>  – Positive - height<br>  – Negative = depth |
| *altitudeWrt-MeanSeaLevel* | • Optional parameter<br>• Altitude With Respect to Sea Level.<br>• Units - Meters |
| *vertUnc* | • Optional parameter<br>• Vertical uncertainty.<br>• Units - Meters<br>• Note - This is mandatory if either pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified. |

| | |
|---|---|
| *vertConfidence* | • Optional parameter<br>• Vertical confidence.<br>• Units - Percentage<br>• Values<br>  – Valid Values - 0 to 99<br>  – Invalid Values - 0, 100-256<br>  – If 100 is received, reinterpret to 99<br>• Note - This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50. |
| *vertReliability* | • Optional parameter<br>  – Values<br>  * 0 - Location reliability is not set.<br>  * 1 - Location reliability is very low; use it at your own risk.<br>  * 2 - Location reliability is low; little or no cross-checking is possible<br>  * 3 - Location reliability is medium; limited cross-check passed<br>  * 4 - Location reliability is high; strong cross-check passed |
| *altitudeSrcInfo* | • Optional parameter<br>  – Pointer to struct altitudeSrcInfo. See altitudeSrcInfo for more information |
| *timestampUtc* | • Optional parameter<br>• UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970 |
| *timestampAge* | • Optional parameter<br>• Position age, which is an estimate of how long ago this fix was made.<br>• Units - Milliseconds |
| *positionSrc* | • Optional parameter<br>• Source from which this position was obtained<br>• Valid values<br>  – 0 - Position source is GNSS<br>  – 1 - Position source is Cell ID<br>  – 2 - Position source is Enhanced Cell ID<br>  – 3 - Position source is Wi-Fi<br>  – 4 - Position source is Terrestrial<br>  – 5 - Position source is GNSS Terrestrial Hybrid<br>  – 6 - Other sources<br>• 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.<br>  – If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source. |

| | |
|---|---|
| *rawHorUncCircular* | • Optional parameter<br>• Horizontal position uncertainty (circular) without any optimization.<br>• Units - Meters |
| *rawHorConfidence* | • Optional parameter<br>• Horizontal confidence associated with raw horizontal uncertainty<br>• Units: Percent<br>• Values<br>– Valid values - 1 to 99<br>– Invalid values - 0, 101 to 255<br>– If 100 is received, reinterpret to 99<br>• Note - This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50. |

### 8.348.2 Field Documentation

#### 8.348.2.1 altSrcInfo_t pack_loc_SLQSLOCInjectPosition_t::altitudeSrcInfo

#### 8.348.2.2 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtEllipsoid

#### 8.348.2.3 float pack_loc_SLQSLOCInjectPosition_t::altitudeWrtMeanSeaLevel

#### 8.348.2.4 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeSrcInfo

#### 8.348.2.5 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtEllipsoid

#### 8.348.2.6 int pack_loc_SLQSLOCInjectPosition_t::has_altitudeWrtMeanSeaLevel

#### 8.348.2.7 int pack_loc_SLQSLOCInjectPosition_t::has_horConfidence

#### 8.348.2.8 int pack_loc_SLQSLOCInjectPosition_t::has_horReliability

#### 8.348.2.9 int pack_loc_SLQSLOCInjectPosition_t::has_horUncCircular

#### 8.348.2.10 int pack_loc_SLQSLOCInjectPosition_t::has_latitude

#### 8.348.2.11 int pack_loc_SLQSLOCInjectPosition_t::has_longitude

#### 8.348.2.12 int pack_loc_SLQSLOCInjectPosition_t::has_positionSrc

#### 8.348.2.13 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorConfidence

#### 8.348.2.14 int pack_loc_SLQSLOCInjectPosition_t::has_rawHorUncCircular

#### 8.348.2.15 int pack_loc_SLQSLOCInjectPosition_t::has_timestampAge

#### 8.348.2.16 int pack_loc_SLQSLOCInjectPosition_t::has_timestampUtc

#### 8.348.2.17 int pack_loc_SLQSLOCInjectPosition_t::has_vertConfidence

#### 8.348.2.18 int pack_loc_SLQSLOCInjectPosition_t::has_vertRelicability

#### 8.348.2.19 int pack_loc_SLQSLOCInjectPosition_t::has_vertUnc

#### 8.348.2.20 uint8_t pack_loc_SLQSLOCInjectPosition_t::horConfidence

#### 8.348.2.21 uint32_t pack_loc_SLQSLOCInjectPosition_t::horReliability

#### 8.348.2.22 float pack_loc_SLQSLOCInjectPosition_t::horUncCircular

#### 8.348.2.23 double pack_loc_SLQSLOCInjectPosition_t::latitude

#### 8.348.2.24 double pack_loc_SLQSLOCInjectPosition_t::longitude

#### 8.348.2.25 uint32_t pack_loc_SLQSLOCInjectPosition_t::positionSrc

#### 8.348.2.26 uint8_t pack_loc_SLQSLOCInjectPosition_t::rawHorConfidence

#### 8.348.2.27 float pack_loc_SLQSLOCInjectPosition_t::rawHorUncCircular

#### 8.348.2.28 uint32_t pack_loc_SLQSLOCInjectPosition_t::timestampAge

#### 8.348.2.29 uint64_t pack_loc_SLQSLOCInjectPosition_t::timestampUtc

#### 8.348.2.30 uint8_t pack_loc_SLQSLOCInjectPosition_t::vertConfidence

#### 8.348.2.31 uint32_t pack_loc_SLQSLOCInjectPosition_t::vertReliability

#### 8.348.2.32 float pack_loc_SLQSLOCInjectPosition_t::vertUnc

## 8.349 pack_loc_SLQSLOCInjectSensorData_t Struct Reference

### Data Fields

- int has_opaqueId
- uint32_t opaqueId
- int has_accleroData
- 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

### 8.349.1 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**

| | |
|---|---|
| *opaqueId* | • Opaque Identifier (Optional parameter)<br>• 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. |
| *acceleroData* | • 3-Axis Accelerometer Data (Optional parameter)<br>• Pointer to struct sensorData. See sensorData_t for more information |
| *gyroData* | • 3-Axis Cyroscope Data (Optional parameter)<br>• Pointer to struct sensorData. See sensorData_t for more information |
| *acceleroTimeSrc* | • 3-Axis Accelerometer Data Time Source (Optional parameter)<br>• The location service uses this field to identify the time reference used in the accelerometer data time stamps.<br>• If not specified, the location service assumes that the time source for the accelerometer data is unknown.<br>• Valid values<br>– 0 - Sensor time source is unspecified<br>– 1 - Time source is common between the sensors and the location engine |
| *gyroTimeSrc* | • 3-Axis Gyroscope Data Time Source (Optional)<br>• The location service uses this field to identify the time reference used in the gyroscope data time stamps.<br>• If not specified, the location service assumes that the time source for the gyroscope data is unknown.<br>• Valid values<br>– 0 - Sensor time source is unspecified<br>– 1 - Time source is common between the sensors and the location engine |
| *accelTemp* | • Accelerometer Temperature Data (Optional parameter)<br>• Pointer to struct tempratureData. See tempData_t for more information |
| *gyroTemp* | • Gyroscope Temperature Data (Optional parameter)<br>• Pointer to struct tempratureData. See tempData_t for more information |

## 8.349.2 Field Documentation

### 8.349.2.1 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData

### 8.349.2.2 uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc

### 8.349.2.3 tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp

### 8.349.2.4 sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData

#### 8.349.2.5 tempData_t pack_loc_SLQSLOCInjectSensorData_t::gyroTemp

#### 8.349.2.6 uint32_t pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc

#### 8.349.2.7 int pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc

#### 8.349.2.8 int pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp

#### 8.349.2.9 int pack_loc_SLQSLOCInjectSensorData_t::has_accleroData

#### 8.349.2.10 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroData

#### 8.349.2.11 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp

#### 8.349.2.12 int pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc

#### 8.349.2.13 int pack_loc_SLQSLOCInjectSensorData_t::has_opaqueId

#### 8.349.2.14 uint32_t pack_loc_SLQSLOCInjectSensorData_t::opaqueId

## 8.350 pack_loc_SLQSLOCInjectUTCTime_t Struct Reference

### Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec

### 8.350.1 Detailed Description

This structure contains inject UTC time parameter.

**Parameters**

| | |
|---|---|
| *timeMsec* | • The UTC time since Jan. 1, 1970 |
| *timeUncMsec* | • The time Uncertainty |

### 8.350.2 Field Documentation

#### 8.350.2.1 uint64_t pack_loc_SLQSLOCInjectUTCTime_t::timeMsec

#### 8.350.2.2 uint32_t pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec

## 8.351 pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference

### Data Fields

- uint32_t state
- int has_confidence
- uint8_t confidence

### 8.351.1 Detailed Description

This structure contains parameters to set current cradle mount configuration Please check has_<Param_Name> field for presence of optional parameters

**Parameters**

| | |
|---|---|
| *state* | • Cradle Mount State<br>• Valid values:<br>– 0 - Device is mounted on the cradle<br>– 1 - Device is not mounted on the cradle<br>– 2 - Unknown cradle mount state |
| *confidence* | • Cradle Mount Confidence (Optional)<br>• Confidence in the Cradle Mount state expressed as a percentage.<br>• Range - 0 to 100 |

### 8.351.2 Field Documentation

#### 8.351.2.1 uint8_t pack_loc_SLQSLOCSetCradleMountConfig_t::confidence

#### 8.351.2.2 int pack_loc_SLQSLOCSetCradleMountConfig_t::has_confidence

#### 8.351.2.3 uint32_t pack_loc_SLQSLOCSetCradleMountConfig_t::state

## 8.352 pack_loc_SLQSLOCSetServer_t Struct Reference

### Data Fields

- uint32_t serverType
- loc_IPv4Config * pIPv4Config
- loc_IPv6Config * pIPv6Config
- loc_URLAddrInfo * pURLAddr

### 8.352.1 Detailed Description

This structure contains parameters to set A-GPS Server

**Parameters**

| | |
|---|---|
| *serverType* | • Type of server<br>• Valid values:<br>– 1 - Server type is CDMA PDE<br>– 2 - Server type is CDMA MPC<br>– 3 - Server type is UMTS SLP<br>– 4 - Server type is custom PDE |

| | |
|---|---|
| *pIPv4Config* | • IPv4 address and port (Optional)<br>• See loc_IPv4Config for more information |
| *pIPv6Config* | • IPv6 address and port (Optional)<br>• See loc_IPv6Config for more information |
| *pURLAddr* | • Uniform Resource Locator (Optional)<br>• See loc_URLAddrInfo for more information |

### 8.352.2 Field Documentation

#### 8.352.2.1 loc_IPv4Config* pack_loc_SLQSLOCSetServer_t::pIPv4Config

#### 8.352.2.2 loc_IPv6Config* pack_loc_SLQSLOCSetServer_t::pIPv6Config

#### 8.352.2.3 loc_URLAddrInfo* pack_loc_SLQSLOCSetServer_t::pURLAddr

#### 8.352.2.4 uint32_t pack_loc_SLQSLOCSetServer_t::serverType

## 8.353 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

### 8.353.1 Detailed Description

This structure contains the LOC Start pack

**Parameters**

| | |
|---|---|
| *SessionId* | • ID of the session as identified by the control point.<br>• Range: 0 to 255 |
| *pRecurrence-Type* | • Optional Parameter<br>• Specifies the type of session in which the control point is interested.<br>• Defaults to SINGLE. -Values<br>– 1 - Request periodic position fixes<br>– 2 - Request a single position fix |

| | |
|---|---|
| *pHorizontal-AccuracyLvl* | • Optional Parameter<br>• Specifies the horizontal accuracy level required by the control point.<br>• Defaults to LOW<br>• Values<br>– 1 - Low accuracy<br>– 2 - Medium accuracy<br>– 3 - High accuracy |
| *pIntermediate-ReportState* | • Optional Parameter<br>• Specifies if the control point is interested in receiving intermediate reports.<br>• ON by default.<br>• Values<br>– 1 - Intermediate reports are turned on<br>– 2 - Intermediate reports are turned off |
| *pMinInterval-Time* | • Optional Parameter<br>• Minimum time interval, specified by the control point, that must elapse between position reports.<br>• Units - Milliseconds<br>• Default - 1000 ms |
| *LocApplication-Info* | • Optional Parameter<br>• LOC Application Parameters<br>• See loc_LocApplicationInfo for more information |
| *pConfigAltitude-Assumed* | • Optional Parameter<br>• Configuration for Altitude Assumed Info in GNSS SV Info Event<br>• Defaults to ENABLED.<br>• Values<br>– 1 - Enable Altitude Assumed information in GNSS SV Info Event<br>– 2 - Disable Altitude Assumed information in GNSS SV Info Event |
| *Tlvresult* | • Unpack result. |

## 8.353.2 Field Documentation

### 8.353.2.1 loc_LocApplicationInfo* pack_loc_Start_t::pApplicationInfo

### 8.353.2.2 uint32_t* pack_loc_Start_t::pConfigAltitudeAssumed

### 8.353.2.3 uint32_t* pack_loc_Start_t::pHorizontalAccuracyLvl

### 8.353.2.4 uint32_t* pack_loc_Start_t::pIntermediateReportState

### 8.353.2.5 uint32_t* pack_loc_Start_t::pMinIntervalTime

#### 8.353.2.6 uint32_t* pack_loc_Start_t::pRecurrenceType

#### 8.353.2.7 uint8_t pack_loc_Start_t::SessionId

#### 8.353.2.8 uint16_t pack_loc_Start_t::Tlvresult

## 8.354 pack_loc_Stop_t Struct Reference

### Data Fields

- uint8_t SessionId
- uint16_t Tlvresult

### 8.354.1 Detailed Description

This structure contains Stop LOC pack

**Parameters**

| | |
|---|---|
| *sessionId* | • ID of the session as identified by the control point.<br>• Range: 0 to 255 |
| *Tlvresult* | • Unpack result. |

### 8.354.2 Field Documentation

#### 8.354.2.1 uint8_t pack_loc_Stop_t::SessionId

#### 8.354.2.2 uint16_t pack_loc_Stop_t::Tlvresult

## 8.355 pack_nas_InitiateDomainAttach_t Struct Reference

### Data Fields

- uint32_t action

### 8.355.1 Detailed Description

Pack structure for Initiates a domain attach/detach of the device.

**Parameters**

| | |
|---|---|
| *action* | • Domain action to attempt<br>1 - Attach<br>2 - Detach |

### 8.355.2 Field Documentation

#### 8.355.2.1 uint32_t pack_nas_InitiateDomainAttach_t::action

## 8.356 pack_nas_SetACCOLC_t Struct Reference

### Data Fields

- int8_t spc [6]
- uint8_t accolc

### 8.356.1 Detailed Description

Sets the access overload class (ACCOLC)

**Parameters**

| | |
|---|---|
| *spc* | • service programming code NULL-terminated string of six digit |
| *accolc* | • ACCOLC : Valid range is 0 to 15 |

### 8.356.2 Field Documentation

#### 8.356.2.1 uint8_t pack_nas_SetACCOLC_t::accolc

#### 8.356.2.2 int8_t pack_nas_SetACCOLC_t::spc[6]

## 8.357 pack_nas_SetCDMANetworkParameters_t Struct Reference

### Data Fields

- char * pSPC
- uint8_t * pForceRev0
- uint8_t * pCustomSCP
- uint32_t * pProtocol
- uint32_t * pBroadcast
- uint32_t * pApplication
- uint32_t * pRoaming

### 8.357.1 Detailed Description

Pack structure to set CDMA Network parameters.

**Parameters**

| | |
|---|---|
| *pSPC* | • Six digit service programming code (not necessary when only the roaming field is being set) |
| *pForceRev0* | • (Optional)Force CDMA 1x-EV-DO Rev. 0 mode<br>– 0 - Disabled<br>– 1 - Enabled Note: Enabled can only be specified if pCustomSCP state is set to Disabled |

| | |
|---|---|
| *pCustomSCP* | • (Optional)Use a custom config for CDMA 1x-EV-DO SCP<br>– 0 - Disabled<br>– 1 - Enabled Note: Enabled can only be specified if pForceRev0 is set to Disabled |
| *pProtocol* | • Protocol mask for custom SCP config<br>– 0x00000001 - Subtype 2 Physical Layer<br>– 0x00000002 - Enhanced CCMAC<br>– 0x00000004 - Enhanced ACMAC<br>– 0x00000008 - Enhanced FTCMAC<br>– 0x00000010 - Subtype 3 RTCMAC<br>– 0x00000020 - Subsystem 1 RTCMAC<br>– 0x00000040 - Enhanced Idle<br>– 0x00000080 - Generic Multimode Capable Disc Port<br>– 0xFFFFFFFF - Unknown |
| *pBroadcast* | • Broadcast mask for custom SCP config<br>– 0x00000001 - Generic broadcast enabled<br>– 0xFFFFFFFF - Unknown |
| *pApplication* | • Application mask for custom SCP config<br>– 0x00000001 - SN Multiflow Packet Application<br>– 0x00000002 - SN Enhanced Multiflow Packet Application<br>– 0xFFFFFFFF - Unknown |
| *pRoaming* | • Roaming preference<br>– 0 - Automatic<br>– 1 - Home Only<br>– 2 - Affiliated Roaming Only<br>– 3 - Home and Affiliated Roaming<br>– 0xFFFFFFFF - Unknown |

### 8.357.2 Field Documentation

#### 8.357.2.1 uint32_t* pack_nas_SetCDMANetworkParameters_t::pApplication

#### 8.357.2.2 uint32_t* pack_nas_SetCDMANetworkParameters_t::pBroadcast

#### 8.357.2.3 uint8_t* pack_nas_SetCDMANetworkParameters_t::pCustomSCP

#### 8.357.2.4 uint8_t* pack_nas_SetCDMANetworkParameters_t::pForceRev0

#### 8.357.2.5 uint32_t* pack_nas_SetCDMANetworkParameters_t::pProtocol

#### 8.357.2.6 uint32_t* pack_nas_SetCDMANetworkParameters_t::pRoaming

#### 8.357.2.7 char* pack_nas_SetCDMANetworkParameters_t::pSPC

## 8.358 pack_nas_SetNetworkPreference_t Struct Reference

### Data Fields

- uint32_t TechnologyPref
- uint32_t Duration
- uint16_t Tlvresult

### 8.358.1 Detailed Description

This structure contains pack set network preference parameters.

**Parameters**

| | |
|---|---|
| *TechnologyPref[-IN]* | • Bitmask representing the radio technology preference set.<br>• No bits set indicates to the device to automatically determine the technology to use<br>• Values:<br>– Bit 0 - Technology is 3GPP2<br>– Bit 1 - Technology is 3GPP<br>• Any combination of the following may be returned:<br>– Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP<br>– Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP<br>– Bit 4 - HDR<br>– Bit 5 - LTE<br>– Bits 6 to 15 - Reserved |
| *Duration[IN]* | • Duration of active preference<br>– 0 - Permanent<br>– 1 - Power cycle<br>– 2 - Until the end of the next call or a power cycle<br>– 3 - Until the end of the next call, a specified time, or a power cycle<br>– 4 to 6 - Until the end of the next call |
| *Tlvresult* | • pack result |

### 8.358.2 Field Documentation

#### 8.358.2.1 uint32_t pack_nas_SetNetworkPreference_t::Duration

#### 8.358.2.2 uint32_t pack_nas_SetNetworkPreference_t::TechnologyPref

#### 8.358.2.3 uint16_t pack_nas_SetNetworkPreference_t::Tlvresult

## 8.359 pack_nas_SLQSConfigSigInfo_t Struct Reference

### Data Fields

- nas_RSSIThresh * pRSSIThresh
- nas_ECIOThresh * pECIOThresh
- nas_HDRSINRThresh * pHDRSINRThresh
- nas_LTESNRThresh * pLTESNRThresh
- nas_IOThresh * pIOThresh
- nas_RSRQThresh * pRSRQThresh
- nas_RSRPThresh * pRSRPThresh
- nas_LTESigRptCfg * pLTESigRptCfg
- nas_TDSCDMASINRCONFThresh * pTDSCDMASINRCONFThresh

### 8.359.1 Detailed Description

Structure for ConfigSigInfo pack.

**Parameters**

| | |
|---|---|
| *pRSSIThresh* | • RSSI threshold List<br>• See nas_RSSIThresh for more details |
| *pECIOThresh* | • ECIO Threshold List<br>• See nas_ECIOThresh for more details |
| *pHDRSINR-Thresh* | • HDR SINR Threshold List<br>• See nas_HDRSINRThresh for more details |
| *pLTESNR-Thresh* | • LTE SNR Threshold List<br>• See nas_LTESNRThresh for more details |
| *pIOThresh* | • IO Threshold List<br>• See nas_IOThresh for more details |
| *pRSRQThresh* | • RSRQ Threshold List<br>• See nas_RSRQThresh for more details |
| *pRSRPThresh* | • RSRP Threshold List<br>• See nas_RSRPThresh for more details |
| *pLTESigRptCfg* | • LTE signal report config<br>• See nas_LTESigRptCfg for more details |
| *pTDSCDMASIN-RCONFThresh* | • TD-SCDMA SINR Threshold List<br>• See nas_TDSCDMASINRCONFThresh for more details |

### 8.359.2 Field Documentation

#### 8.359.2.1 nas_ECIOThresh∗ pack_nas_SLQSConfigSigInfo_t::pECIOThresh

#### 8.359.2.2 nas_HDRSINRThresh∗ pack_nas_SLQSConfigSigInfo_t::pHDRSINRThresh

#### 8.359.2.3 nas_IOThresh∗ pack_nas_SLQSConfigSigInfo_t::pIOThresh

#### 8.359.2.4 nas_LTESigRptCfg∗ pack_nas_SLQSConfigSigInfo_t::pLTESigRptCfg

#### 8.359.2.5 nas_LTESNRThresh∗ pack_nas_SLQSConfigSigInfo_t::pLTESNRThresh

#### 8.359.2.6 nas_RSRPThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRPThresh

#### 8.359.2.7 nas_RSRQThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSRQThresh

#### 8.359.2.8 nas_RSSIThresh∗ pack_nas_SLQSConfigSigInfo_t::pRSSIThresh

#### 8.359.2.9 nas_TDSCDMASINRCONFThresh∗ pack_nas_SLQSConfigSigInfo_t::pTDSCDMASINRCONFThresh

## 8.360 pack_nas_SLQSGetPLMNName_t Struct Reference

### Data Fields

- uint16_t mcc
- uint16_t mnc
- uint8_t ∗ pMncPcsStatus

### 8.360.1 Detailed Description

This structure contains pack get operator name for specificed network parameters.

**Parameters**

| | |
|---|---|
| *mcc* | • A 16-bit integer representation of MCC. Range: 0 to 999 |
| *mnc* | • A 16-bit integer representation of MNC. Range: 0 to 999 |
| *pMncPcsStatus* | • MNC PCS Digit Include Status<br>• Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01).<br>• Values<br>– TRUE - MNC is a three-digit value. e.g. a reported value of 90 corresponds to an MNC value of 090<br>– FALSE - MNC is a two-digit value. e.g. a reported value of 90 corresponds to an MNC value of 90 |

**Note**

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

### 8.360.2 Field Documentation

#### 8.360.2.1 uint16_t pack_nas_SLQSGetPLMNName_t::mcc

#### 8.360.2.2 uint16_t pack_nas_SLQSGetPLMNName_t::mnc

#### 8.360.2.3 uint8_t∗ pack_nas_SLQSGetPLMNName_t::pMncPcsStatus

## 8.361 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference

### Data Fields

- uint32_t regAction
- nas_MNRInfo ∗ pMNRInfo
- uint32_t ∗ pChangeDuration
- uint8_t ∗ pMncPcsDigitStatus

### 8.361.1 Detailed Description

This structure contains Initiate Network Registration request parameters

**Parameters**

| | |
|---|---|
| *regAction* | • Specifies one of the following register actions :<br>– AUTO_REGISTER - Device registers according to its provisioning and optional parameters supplied with the command are ignored.<br>– MANUAL_REGISTER - Device registers to a specified network and the optional Manual Network Register Information parameter pMNRInfo must also be included for the command to process successfully and supported only for 3GPP. |
| *pMNRInfo* | [Optional]<br>• Pointer to structure MNRInfo<br>– See nas_MNRInfo for more information |
| *pChange-Duration* | [Optional]<br>• Duration of the change.<br>– 0x00 - Power cycle - Remains active until the next device power cycle<br>– 0x01 - Permanent - Remains active through power cycles until changed by the client |
| *pMncPcsDigit-Status* | [Optional]<br>• MNC PCS Digit Include Status<br>– True - MNC is a 3-digit value.<br>– False - MNC is a 2-digit value. |

### 8.361.2 Field Documentation

#### 8.361.2.1 uint32_t∗ pack_nas_SLQSInitiateNetworkRegistration_t::pChangeDuration

#### 8.361.2.2 uint8_t∗ pack_nas_SLQSInitiateNetworkRegistration_t::pMncPcsDigitStatus

#### 8.361.2.3 nas_MNRInfo* pack_nas_SLQSInitiateNetworkRegistration_t::pMNRInfo

#### 8.361.2.4 uint32_t pack_nas_SLQSInitiateNetworkRegistration_t::regAction

## 8.362 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_TDSCDMARSCPThresh * pTDSCDMARSCPThresh
- uint16_t * pTDSCDMARSCPDelta
- nas_TDSCDMARSSIThresh * pTDSCDMARSSIThresh
- float * pTDSCDMARSSIDelta
- nas_TDSCDMAECIOThresh * pTDSCDMAECIOThresh
- float * pTDSCDMAECIODelta
- nas_TDSCDMASINRThresh * pTDSCDMASINRThresh
- float * pTDSCDMASINRDelta

### 8.362.1 Detailed Description

This structure contains pack set the signal strength reporting thresholds parameters.

**Parameters**

| | |
|---|---|
| *pCDMARSSI-Thresh* | • CDMA RSSI threshold List |
| *pCDMARSSI-Delta* | • RSSI delta (in units of 0.1 dBm).<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pCDMAECIO-Thresh* | • CDMA ECIO Threshold List |
| *pCDMAECIO-Delta* | • ECIO delta (in units of 0.1 dB).<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pHDRRSSI-Thresh* | • HDR RSSI Threshold List |
| *pHDRRSSIDelta* | • RSSI delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pHDRECIO-Thresh* | • HDR ECIO Threshold List |
| *pHDRECIODelta* | • ECIO delta (in units of 0.1 dB)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pHDRSINR-Thresh* | • HDR SINR Threshold List |
| *pHDRSINRDelta* | • SINR delta (in units of 1 SINR level)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pHDRIOThresh* | • HDR IO Threshold List |
| *pHDRIODelta* | • IO delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pGSMRSSI-Thresh* | • GSM RSSI Threshold List<br>• See nas_GSMRSSIThresh for more details |
| *pGSMRSSIDelta* | • RSSI delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |

| | |
|---|---|
| *pWCDMARSSI-Thresh* | • WCDMA RSSI Threshold List<br>• See nas_WCDMARSSIThresh for more details |
| *pWCDMARSSI-Delta* | • RSSI delta (in units of 0.1 dBm).<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pWCDMAECIO-Thresh* | • WCDMA ECIO Threshold List |
| *pWCDMAECIO-Delta* | • ECIO delta (in units of 0.1 dB)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pLTERSSI-Thresh* | • LTE RSSI Threshold List |
| *pLTERSSIDelta* | • RSSI delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pLTESNR-Thresh* | • LTE SNR Threshold List |
| *pLTESNRDelta* | • SNR delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pLTERSRQ-Thresh* | • LTE RSRQ Threshold List |
| *pLTERSRQ-Delta* | • RSRQ delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pLTERSRP-Thresh* | • LTE RSRP Threshold List |
| *pLTERSRPDelta* | • RSRP delta (in units of 0.1 dBm).<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |
| *pLTESigRpt-Config* | • LTE Signal Report Config |
| *pTDSCDMARS-CPThresh* | • TDSCDMA RSCP Threshold List |
| *pTDSCDMARS-CPDelta* | • RSCP delta (in units of 0.1 dBm)<br>• A value of 0 is rejected with a QMI_ERR_INVALID_ARG error. |

| | |
|---|---|
| *pTDSCDMARSSIThresh* | • TDSCDMA RSSI Threshold List |
| *pTDSCDMARSSIDelta* | • RSSI delta (in dBm) used by TD-SCDMA. |
| *pTDSCDMAECIOThresh* | • TDSCDMA ECIO Threshold List |
| *pTDSCDMAECIODelta* | • ECIO delta (in dB) used by TD-SCDMA |
| *pTDSCDMASINRThresh* | • TDSCDMA SINR Threshold List |
| *pTDSCDMASINRDelta* | • SINR delta (in dB) used by TD-SCDMA. |

**Note**

Mixture of threshold and delta values can be provided in the request. But for each type and RAT,only one of threshold list or delta value is to be provided.

### 8.362.2 Field Documentation

#### 8.362.2.1 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIODelta

#### 8.362.2.2 nas_CDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMAECIOThresh

#### 8.362.2.3 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIDelta

#### 8.362.2.4 nas_CDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pCDMARSSIThresh

#### 8.362.2.5 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIDelta

#### 8.362.2.6 nas_GSMRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pGSMRSSIThresh

#### 8.362.2.7 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIODelta

#### 8.362.2.8 nas_HDRECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOThresh

#### 8.362.2.9 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta

#### 8.362.2.10 nas_HDRIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOThresh

#### 8.362.2.11 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIDelta

#### 8.362.2.12 nas_HDRRSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pHDRRSSIThresh

#### 8.362.2.13 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRDelta

#### 8.362.2.14 nas_HDRSINRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pHDRSINRThresh

#### 8.362.2.15 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta

#### 8.362.2.16 nas_LTERSRPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPThresh

#### 8.362.2.17 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta

#### 8.362.2.18 nas_LTERSRQThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQThresh

#### 8.362.2.19 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta

#### 8.362.2.20 nas_LTERSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIThresh

#### 8.362.2.21 nas_LTESigRptConfig* pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig

#### 8.362.2.22 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta

#### 8.362.2.23 nas_LTESNRThreshold* pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh

#### 8.362.2.24 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta

#### 8.362.2.25 nas_TDSCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOThresh

#### 8.362.2.26 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta

#### 8.362.2.27 nas_TDSCDMARSCPThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh

#### 8.362.2.28 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta

#### 8.362.2.29 nas_TDSCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh

#### 8.362.2.30 float* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta

#### 8.362.2.31 nas_TDSCDMASINRThresh* pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh

#### 8.362.2.32 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIODelta

#### 8.362.2.33 nas_WCDMAECIOThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh

#### 8.362.2.34 uint16_t* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta

#### 8.362.2.35 nas_WCDMARSSIThresh* pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh

## 8.363 pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference

### Data Fields

- uint8_t namID

### 8.363.1 Detailed Description

Structure for 3GPP2 Subscription pack.

**Parameters**

| | |
|---|---|
| *namID* | [Mandatory]<br>• NAM ID of the information to be retrieved. The index starts from 0. A nam_id of 0xFF is used to retrieve information of current NAM. |

### 8.363.2 Field Documentation

#### 8.363.2.1 uint8_t pack_nas_SLQSNasGet3GPP2Subscription_t::namID

## 8.364 pack_nas_SLQSNasGetTxRxInfo_t Struct Reference

### Data Fields

- uint8_t radio_if

### 8.364.1 Detailed Description

Structure for TxRx Info pack

**Parameters**

| | |
|---|---|
| *radio_if* | [Mandatory]<br>• Radio interface technology of the signal being measured<br>• Valid Values<br>– 0x01 - NAS_RADIO_IF_CDMA_1X - CDMA<br>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - HDR<br>– 0x04 - NAS_RADIO_IF_GSM - GSM<br>– 0x05 - NAS_RADIO_IF_UMTS - UMTS<br>– 0x08 - NAS_RADIO_IF_LTE - LTE |

### 8.364.2 Field Documentation

#### 8.364.2.1 uint8_t pack_nas_SLQSNasGetTxRxInfo_t::radio_if

## 8.365 pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference

### Data Fields

- uint8_t * pSystemSelectionInd
- uint8_t * pDDTMInd
- uint8_t * pServingSystemInd
- uint8_t * pDualStandByPrefInd
- uint8_t * pSubscriptionInfoInd
- uint8_t * pNetworkTimeInd
- uint8_t * pSysInfoInd
- uint8_t * pSignalStrengthInd
- uint8_t * pErrorRateInd
- uint8_t * pHDRNewUATIAssInd
- uint8_t * pHDRSessionCloseInd
- uint8_t * pManagedRoamingInd

- uint8_t * pNetworkRejectInd
- uint8_t * pSuppressSysInfoInd
- uint8_t * pLTECphyCa

### 8.365.1 Detailed Description

This structure contains pack indication register parameters.

**Parameters**

| | |
|---|---|
| *pSystemSelectionInd* | [Optional]<br>• System Selection Preference indication registration. The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pDDTMInd* | [Optional]<br>• DDTM (Data Dedicated Transmission Mode) indication registration. The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pServingSystemInd* | [Optional]<br>• Serving System indication registration. The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pDualStandByPrefInd* | [Optional]<br>• Dual Standby Preference indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSubscriptionInfoInd* | [Optional]<br>• Subscription Information indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pNetworkTimeInd* | [Optional]<br>• Network Time indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSysInfoInd* | [Optional]<br>• System Information indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |

| | |
|---|---|
| *pSignalStrength-Ind* | [Optional]<br>• Signal Strength indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pErrorRateInd* | [Optional]<br>• Error Rate indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pHDRNewUATI-AssInd* | [Optional]<br>• HDR New UATI Assigned indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pHDRSession-CloseInd* | [Optional]<br>• HDR Session Closed indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pManaged-RoamingInd* | [Optional]<br>• Managed Roaming indication registration.The callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable |
| *pNetworkReject-Ind* | [Optional]<br>• Network reject information indication registration.The following callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSuppressSys-InfoInd* | [Optional]<br>• Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values:<br>– 0x00 - Disable<br>– 0x01 - Enable- 0x01 - Enable |
| *pLTECphyCa* | [Optional]<br>• LTE Physical Carrier Aggregation Information.The following callbacks would not be invoked if the indication is disabled.<br>– 0x00 - Disable (default value)<br>– 0x01 - Enable |

### 8.365.2 Field Documentation

#### 8.365.2.1 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd

#### 8.365.2.2 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd

#### 8.365.2.3 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd

#### 8.365.2.4 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd

#### 8.365.2.5 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd

#### 8.365.2.6 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa

#### 8.365.2.7 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd

#### 8.365.2.8 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd

#### 8.365.2.9 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd

#### 8.365.2.10 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd

#### 8.365.2.11 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd

#### 8.365.2.12 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd

#### 8.365.2.13 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd

#### 8.365.2.14 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd

#### 8.365.2.15 uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd

## 8.366 pack_nas_SLQSNASSeteDRXParams_t Struct Reference

### Data Fields

- uint8_t * pEdrxEnable
- uint8_t * pCycleLen
- uint8_t * pPagingTimeWindow

### 8.366.1 Detailed Description

This structure contains the SLQSNASSeteDRXParams request parameters.

**Parameters**

| | |
|---|---|
| *pEdrxEnable* | [Optional]<br>• eDRX Enable<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pCycleLen* | [Optional]<br>• eDRX cycle length<br>• Value to be set per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 |

| | |
|---|---|
| *pPagingTime-Window* | [Optional]<br>• eDRX paging time window<br>• Value to be set per 3GPP TS 24.008 Rel-13 Section 10.5.5.32 |

### 8.366.2 Field Documentation

#### 8.366.2.1 uint8_t∗ pack_nas_SLQSNASSeteDRXParams_t::pCycleLen

#### 8.366.2.2 uint8_t∗ pack_nas_SLQSNASSeteDRXParams_t::pEdrxEnable

#### 8.366.2.3 uint8_t∗ pack_nas_SLQSNASSeteDRXParams_t::pPagingTimeWindow

## 8.367 pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference

### Data Fields

- uint8_t lteEsmUl
- uint8_t lteEsmDl
- uint8_t lteEmmUl
- uint8_t lteEmmDl
- uint8_t gsmUmtsUl
- uint8_t gsmUmtsDl
- uint8_t ∗ pRankIndicatorInd
- uint8_t ∗ pTimer

### 8.367.1 Detailed Description

This structure contains the OTA message indication.

**Parameters**

| | |
|---|---|
| *lteEsmUl* | • 0 - do not report<br>• 1 - report LTE ESM uplink messages |
| *lteEsmDl* | • 0 - do not report<br>• 1 - report LTE ESM downlink messages |
| *lteEmmUl* | • 0 - do not report<br>• 1 - report LTE EMM uplink messages |
| *lteEmmDl* | • 0 - do not report<br>• 1 - report GSM/UMTS uplink messages |
| *gsmUmtsUl* | • 0 - do not report<br>• 1 - report GSM/UMTS uplink messages |

| | |
|---|---|
| *gsmUmtsDl* | • 0 - do not report<br>• 1 - report GSM/UMTS downlink messages |
| *pRankIndicator-Ind* | • 0 - do not report<br>• 1 - report Rank Indicator messages |
| *pTimer* | • 0 - do not report<br>• 1 - report Timer Indicator messages |

### 8.367.2 Field Documentation

#### 8.367.2.1 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsDl

#### 8.367.2.2 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::gsmUmtsUl

#### 8.367.2.3 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmDl

#### 8.367.2.4 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEmmUl

#### 8.367.2.5 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmDl

#### 8.367.2.6 uint8_t pack_nas_SLQSNasSwiIndicationRegister_t::lteEsmUl

#### 8.367.2.7 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pRankIndicatorInd

#### 8.367.2.8 uint8_t* pack_nas_SLQSNasSwiIndicationRegister_t::pTimer

## 8.368 pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference

### Data Fields

- nas_wcdmaUARFCN * pWcdmaUARFCN
- nas_lteEARFCN * pLteEARFCN
- nas_ltePCI * pLtePCI

### 8.368.1 Detailed Description

Structure for ChannelLock pack.

**Parameters**

| | |
|---|---|
| *pWcdmaUARF-CN* | [Optional]<br>• See nas_wcdmaUARFCN for more information |
| *pLteEARFCN* | [Optional]<br>• See nas_lteEARFCN for more information |
| *pLtePCI* | [Optional]<br>• See nas_ltePCI for more information |

### 8.368.2 Field Documentation

#### 8.368.2.1 nas_lteEARFCN* pack_nas_SLQSNASSwiSetChannelLock_t::pLteEARFCN

#### 8.368.2.2 nas_ltePCI* pack_nas_SLQSNASSwiSetChannelLock_t::pLtePCI

#### 8.368.2.3 nas_wcdmaUARFCN* pack_nas_SLQSNASSwiSetChannelLock_t::pWcdmaUARFCN

## 8.369 pack_nas_SLQSSetBandPreference_t Struct Reference

### Data Fields

- uint64_t bandPref

### 8.369.1 Detailed Description

Provides information about the band preference.

**Parameters**

| | |
|---|---|
| *bandpreference* | • Bit mask representing the band preference to be set.<br>• Bit position meanings:<br>– 0 - BC0_A - Band Class 0, A-System<br>– 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band<br>– 2 - BC1 - Band Class 1, all blocks<br>– 3 - BC2 - Band Class 2 place holder<br>– 4 - BC3 - Band Class 3, A-System<br>– 5 - BC4 - Band Class 4, all blocks<br>– 6 - BC5 - Band Class 5, all blocks<br>– 7 - GSM_DCS_1800 - GSM DCS band<br>– 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band<br>– 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band<br>– 10 - BC6 - Band Class 6<br>– 11 - BC7 - Band Class 7<br>– 12 - BC8 - Band Class 8<br>– 13 - BC9 - Band Class 9<br>– 14 - BC10 - Band Class 10<br>– 15 - BC11 - Band Class 11<br>– 16 - GSM_450 - GSM 450 band<br>– 17 - GSM_480 - GSM 480 band<br>– 18 - GSM_750 - GSM 750 band<br>– 19 - GSM_850 - GSM 850 band<br>– 20 - GSM_RGSM_900 - GSM Railways GSM Band<br>– 21 - GSM_PCS_1900 - GSM PCS band<br>– 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band<br>– 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band<br>– 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band<br>– 25 - WCDMA_IV_1700 - WCDMA US 1700 band<br>– 26 - WCDMA_V_850 - WCDMA US 850 band<br>– 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band<br>– 28 - BC12 - Band Class 12<br>– 29 - BC14 - Band Class 14<br>– 30 - RESERVED_2 - Reserved 2<br>– 31 - BC15 - Band Class 15<br>– 32 - 47 - Reserved<br>– 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band<br>– 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band<br>– 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band<br>– 51 to 55 - Reserved<br>– 56 - BBC16 - Band Class 16<br>– 57 - BC17 - Band Class 17<br>– 58 - BC18 - Band Class 18<br>– 59 - BC19 - Band Class 19<br>– 60 to 64 - Reserved |

### 8.369.2 Field Documentation

#### 8.369.2.1 uint64_t pack_nas_SLQSSetBandPreference_t::bandPref

## 8.370 pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference

### Data Fields

- uint8_t bEnable
- nas_SLQSSignalStrengthsIndReq * pSigIndReq

### 8.370.1 Detailed Description

This structure contains paramaters pack set strength thresholds callback.

**Parameters**

| | |
|---|---|
| *bEnable* | 0/1 to disable/enable RSSI signal stregth indication |
| *pSigIndReq* | parameters to control signal strength indication |

### 8.370.2 Field Documentation

#### 8.370.2.1 uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable

#### 8.370.2.2 nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq

## 8.371 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 * pTdscdmaBandPref
- struct nas_acqOrderPref * pAcqOrderPref
- uint32_t * pSrvRegRestriction
- struct nas_CSGID * pCSGID
- unsigned char * pRAT

### 8.371.1 Detailed Description

Contain the system selection preferences.

**Parameters**

| | |
|---|---|
| *pEmerMode* | • Optional parameter specifying the emergency Mode<br>• Values:<br>– 0 - OFF (normal)<br>– 1 - ON (Emergency) |
| *pModePref* | • Optional parameter<br>• Bit Mask indicating the radio technology mode preference<br>• Bit values:<br>– Bit 0 - cdma2000 1x<br>– Bit 1 - cdma2000 HRPD(1xEV-DO)<br>– Bit 2 - GSM<br>– Bit 3 - UMTS<br>– Bit 4 - LTE |

| | |
|---|---|
| *pBandPref* | • Optional parameter<br>• Bit mask representing the band preference<br>• Bit values:<br>– Bit 0 - Band Class 0, A-System<br>– Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band<br>– Bit 2 - Band Class 1, all blocks<br>– Bit 3 - Band Class 2 place holder<br>– Bit 4 - Band Class 3, A-System<br>– Bit 5 - Band Class 4, all blocks<br>– Bit 6 - Band Class 5, all blocks<br>– Bit 7 - GSM_DCS_1800 band<br>– Bit 8 - GSM Extended GSM (E-GSM) 900 band<br>– Bit 9 - GSM Primary GSM (P-GSM) 900 band<br>– Bit 10 - Band Class 6<br>– Bit 11 - Band Class 7<br>– Bit 12 - Band Class 8<br>– Bit 13 - Band Class 9<br>– Bit 14 - Band Class 10<br>– Bit 15 - Band Class 11<br>– Bit 16 - GSM 450 band<br>– Bit 17 - GSM 480 band<br>– Bit 18 - GSM 750 band<br>– Bit 19 - GSM 850 band<br>– Bit 20 - GSM Railways GSM 900 Band<br>– Bit 21 - GSM PCS 1900 band<br>– Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band<br>– Bit 23 - WCDMA U.S. PCS 1900 band<br>– Bit 24 - WCDMA Europe and China DCS 1800 band<br>– Bit 25 - WCDMA U.S. 1700 band<br>– Bit 26 - WCDMA U.S. 850 band<br>– Bit 27 - WCDMA Japan 800 band<br>– Bit 28 - Band Class 12<br>– Bit 29 - Band Class 14<br>– Bit 30 - Reserved<br>– Bit 31 - Band Class 15<br>– Bit 32 to 47 - Reserved<br>– Bit 48 - WCDMA Europe 2600 band<br>– Bit 49 - WCDMA Europe and Japan 900 band<br>– Bit 50 - WCDMA Japan 1700 band<br>– Bit 51 to 55 - Reserved<br>– Bit 56 - Band Class 16<br>– Bit 57 - Band Class 17<br>– Bit 58 - Band Class 18<br>– Bit 59 - Band Class 19<br>– Bit 60 to 64 - Reserved |

| | |
|---|---|
| *pPRLPref* | • Optional parameter indicating the CDMA PRL Preference<br>• Values:<br>– 0x0001 - Acquire available system only on the A side<br>– 0x0002 - Acquire available system only on the B side<br>– 0x3FFF - Acquire any available systems |
| *pRoamPref* | • Optional parameter indicating the roaming Preference<br>• Values:<br>– 0x01 - Acquire only systems for which the roaming indicator is off<br>– 0x02 - Acquire a system as long as its roaming indicator is not off<br>– 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only<br>– 0xFF - Acquire systems, regardless of their roaming indicator |
| *pLTEBandPref* | • Optional parameter<br>• Bit mask representing the LTE band preference<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– All other bits are reserved |

| | |
|---|---|
| *pNetSelPref* | - netSelectionPref<br>• Optional parameter for specifying Network Selection Preference<br>• Modem selects networks based on this parameter(if present).<br>• see nas_netSelectionPref for more information |
| *pChgDuration* | • Optional parameter specifying the duration of the change<br>• Values:<br>– 0x00 - Power cycle - Remains active until the next device power cycle<br>– 0x01 - Permanent - Remains active through power cycles until changed by client<br>– Device will use "0x01 - permanent" as default if this parameter is omitted |
| *pMNCIncPCS-DigStat* | • Optional parameter indicating if MNC includes PCS digit<br>• Values:<br>– TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090<br>– FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90 |
| *pSrvDomainPref* | • Optional parameter indicating Service domain preference<br>• Values:<br>– 0x00 - Circuit switched only<br>– 0x01 - Packet switched only<br>– 0x02 - Circuit switched and packet switched<br>– 0x03 - Packet switched attach<br>– 0x04 - Packet switched detach |
| *pGWAcqOrder-Pref* | • Optional parameter indicating GSM/WCDMA Acquisition order Preference<br>• Values:<br>– 0x00 - Automatic<br>– 0x01 - GSM then WCDMA<br>– 0x02 - WCDMA then GSM |
| *pTdscdmaBand-Pref* | • Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set.<br>• Values:<br>– 0x01 - TD-SCDMA Band A<br>– 0x02 - TD-SCDMA Band B<br>– 0x04 - TD-SCDMA Band C<br>– 0x08 - TD-SCDMA Band D<br>– 0x10 - TD-SCDMA Band E<br>– 0x20 - TD-SCDMA Band F<br>– All other bits are reserved |
| *pAcqOrderPref* | - acqOrderPref<br>• Optional parameter for specifying Acquisition Order Preference<br>• see nas_acqOrderPref for more information |

| | |
|---|---|
| *pSrvReg-Restriction* | • Optional parameter indicating Network Selection Registration Restriction Preference<br>• Values:<br>– 0x00 - Device follows the normal registration process<br>– 0x01 - Device camps on the network according to its provisioning, but does not register<br>– 0x02 - Device selects the network for limited service<br>– All other values are reserved. |
| *pCSGID* | - CSGID<br>• Optional parameter for specifying CSG ID<br>• Either of pNetSelPref or pCSGID can be set.<br>• see nas_CSGID for more information |
| *pRAT* | • Optional parameter Radio Access Technology order Preference<br>• Values:<br>– 0x04 - GSM<br>– 0x05 - UMTS<br>– 0x08 - LTE<br>– 0x09 - TDSCDMA |

### 8.371.2 Field Documentation

#### 8.371.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPref_t::pAcqOrderPref

#### 8.371.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pBandPref

#### 8.371.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pChgDuration

#### 8.371.2.4 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPref_t::pCSGID

#### 8.371.2.5 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pEmerMode

#### 8.371.2.6 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pGWAcqOrderPref

#### 8.371.2.7 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pLTEBandPref

#### 8.371.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPref_t::pMNCIncPCSDigStat

#### 8.371.2.9 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pModePref

#### 8.371.2.10 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPref_t::pNetSelPref

#### 8.371.2.11 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pPRLPref

#### 8.371.2.12 unsigned char* pack_nas_SLQSSetSysSelectionPref_t::pRAT

#### 8.371.2.13 uint16_t* pack_nas_SLQSSetSysSelectionPref_t::pRoamPref

#### 8.371.2.14 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvDomainPref

#### 8.371.2.15 uint32_t* pack_nas_SLQSSetSysSelectionPref_t::pSrvRegRestriction

#### 8.371.2.16 uint64_t* pack_nas_SLQSSetSysSelectionPref_t::pTdscdmaBandPref

## 8.372 pack_nas_SLQSSetSysSelectionPrefExt_t Struct Reference

### Data Fields

- uint8_t * pEmerMode
- uint16_t * pModePref
- uint64_t * pBandPref
- uint16_t * pPRLPref
- uint16_t * pRoamPref
- uint64_t * pLTEBandPref
- struct nas_netSelectionPref * pNetSelPref
- uint8_t * pChgDuration
- uint8_t * pMNCIncPCSDigStat
- uint32_t * pSrvDomainPref
- uint32_t * pGWAcqOrderPref
- uint64_t * pTdscdmaBandPref
- struct nas_acqOrderPref * pAcqOrderPref
- uint32_t * pSrvRegRestriction
- struct nas_CSGID * pCSGID
- unsigned char * pRAT
- struct nas_lteBandPrefExt * pLTEBandPrefExt
- uint64_t * pCiotLteOpModePref
- uint64_t * pLteM1BandPref
- uint64_t * pLteNb1BandPref
- uint32_t * pCiotLteOpMode
- struct nas_ciotAcqOrderPref * pCiotAcqOrderPref
- struct nas_nr5gBandPref * pNr5gBandPref

### 8.372.1 Detailed Description

Contain the system selection preferences.

**Parameters**

| | |
|---|---|
| *pEmerMode* | • Optional parameter specifying the emergency Mode<br>• Values:<br>– 0 - OFF (normal)<br>– 1 - ON (Emergency) |

| | |
|---|---|
| *pModePref* | • Optional parameter<br>• Bit Mask indicating the radio technology mode preference<br>• Bit values:<br>– Bit 0 - cdma2000 1x<br>– Bit 1 - cdma2000 HRPD(1xEV-DO)<br>– Bit 2 - GSM<br>– Bit 3 - UMTS<br>– Bit 4 - LTE<br>– Bit 5 - TD-SCDMA<br>– Bit 6 - NR5G |

| | |
|---|---|
| *pBandPref* | • Optional parameter<br>• Bit mask representing the band preference<br>• Bit values:<br>– Bit 0 - Band Class 0, A-System<br>– Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band<br>– Bit 2 - Band Class 1, all blocks<br>– Bit 3 - Band Class 2 place holder<br>– Bit 4 - Band Class 3, A-System<br>– Bit 5 - Band Class 4, all blocks<br>– Bit 6 - Band Class 5, all blocks<br>– Bit 7 - GSM_DCS_1800 band<br>– Bit 8 - GSM Extended GSM (E-GSM) 900 band<br>– Bit 9 - GSM Primary GSM (P-GSM) 900 band<br>– Bit 10 - Band Class 6<br>– Bit 11 - Band Class 7<br>– Bit 12 - Band Class 8<br>– Bit 13 - Band Class 9<br>– Bit 14 - Band Class 10<br>– Bit 15 - Band Class 11<br>– Bit 16 - GSM 450 band<br>– Bit 17 - GSM 480 band<br>– Bit 18 - GSM 750 band<br>– Bit 19 - GSM 850 band<br>– Bit 20 - GSM Railways GSM 900 Band<br>– Bit 21 - GSM PCS 1900 band<br>– Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band<br>– Bit 23 - WCDMA U.S. PCS 1900 band<br>– Bit 24 - WCDMA Europe and China DCS 1800 band<br>– Bit 25 - WCDMA U.S. 1700 band<br>– Bit 26 - WCDMA U.S. 850 band<br>– Bit 27 - WCDMA Japan 800 band<br>– Bit 28 - Band Class 12<br>– Bit 29 - Band Class 14<br>– Bit 30 - Reserved<br>– Bit 31 - Band Class 15<br>– Bit 32 to 47 - Reserved<br>– Bit 48 - WCDMA Europe 2600 band<br>– Bit 49 - WCDMA Europe and Japan 900 band<br>– Bit 50 - WCDMA Japan 1700 band<br>– Bit 51 to 55 - Reserved<br>– Bit 56 - Band Class 16<br>– Bit 57 - Band Class 17<br>– Bit 58 - Band Class 18<br>– Bit 59 - Band Class 19<br>– Bit 60 to 64 - Reserved |

| | |
|---|---|
| *pPRLPref* | • Optional parameter indicating the CDMA PRL Preference<br>• Values:<br>– 0x0001 - Acquire available system only on the A side<br>– 0x0002 - Acquire available system only on the B side<br>– 0x3FFF - Acquire any available systems |
| *pRoamPref* | • Optional parameter indicating the roaming Preference<br>• Values:<br>– 0x01 - Acquire only systems for which the roaming indicator is off<br>– 0x02 - Acquire a system as long as its roaming indicator is not off<br>– 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only<br>– 0xFF - Acquire systems, regardless of their roaming indicator |

| | |
|---|---|
| *pLTEBandPref* | • Optional parameter<br>• Bit mask representing the LTE band preference<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 22 - E-UTRA Operating Band 23<br>– Bit 23 - E-UTRA Operating Band 24<br>– Bit 24 - E-UTRA Operating Band 25<br>– Bit 25 - E-UTRA Operating Band 26<br>– Bit 27 - E-UTRA Operating Band 28<br>– Bit 28 - E-UTRA Operating Band 29<br>– Bit 29 - E-UTRA Operating Band 32<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– Bit 40 - E-UTRA Operating Band 41<br>– Bit 41 - E-UTRA Operating Band 42<br>– Bit 42 - E-UTRA Operating Band 43<br>– Bit 60 - E-UTRA Operating Band 125<br>– All other bits are reserved |

| | |
|---|---|
| *pNetSelPref* | - netSelectionPref<br>• Optional parameter for specifying Network Selection Preference<br>• Modem selects networks based on this parameter(if present).<br>• Either of pNetSelPref or pCSGID can be set.<br>• see nas_netSelectionPref for more information |
| *pChgDuration* | • Optional parameter specifying the duration of the change<br>• At least one system selection setting to be set if pChgDuration is populated.<br>• Values:<br>– 0x00 - Power cycle - Remains active until the next device power cycle<br>– 0x01 - Permanent - Remains active through power cycles until changed by client<br>– Device will use "0x01 - permanent" as default if this parameter is omitted |
| *pMNCIncPCS-DigStat* | • Optional parameter indicating if MNC includes PCS digit<br>• pNetSelPref is expected if MNC includes PCS digit is set to 1.<br>• Values:<br>– TRUE - MNC is a 3 digit value; e.g., a reported value of 90 corresponds to an MNC value of 090<br>– FALSE - MNC is a 2-digit value; e.g., a reported value of 90 corresponds to an MNC value of 90 |
| *pSrvDomainPref* | • Optional parameter indicating Service domain preference<br>• Values:<br>– 0x00 - Circuit switched only<br>– 0x01 - Packet switched only<br>– 0x02 - Circuit switched and packet switched<br>– 0x03 - Packet switched attach<br>– 0x04 - Packet switched detach |
| *pGWAcqOrder-Pref* | • Optional parameter indicating GSM/WCDMA Acquisition order Preference<br>• Values:<br>– 0x00 - Automatic<br>– 0x01 - GSM then WCDMA<br>– 0x02 - WCDMA then GSM |
| *pTdscdmaBand-Pref* | • Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set.<br>• Values:<br>– 0x01 - TD-SCDMA Band A<br>– 0x02 - TD-SCDMA Band B<br>– 0x04 - TD-SCDMA Band C<br>– 0x08 - TD-SCDMA Band D<br>– 0x10 - TD-SCDMA Band E<br>– 0x20 - TD-SCDMA Band F<br>– All other bits are reserved |

| | |
|---|---|
| *pAcqOrderPref* | - acqOrderPref<br>• Optional parameter for specifying Acquisition Order Preference<br>• see nas_acqOrderPref for more information |
| *pSrvReg-Restriction* | • Optional parameter indicating Network Selection Registration Restriction Preference<br>• Values:<br>– 0x00 - Device follows the normal registration process<br>– 0x01 - Device camps on the network according to its provisioning, but does not register<br>– 0x02 - Device selects the network for limited service<br>– All other values are reserved. |
| *pCSGID* | - CSGID<br>• Optional parameter for specifying CSG ID<br>• Either of pNetSelPref or pCSGID can be set.<br>• see nas_CSGID for more information |
| *pRAT* | • Optional parameter Radio Access Technology order Preference<br>• Values:<br>– 0x04 - GSM<br>– 0x05 - UMTS<br>– 0x08 - LTE<br>– 0x09 - TDSCDMA<br>– 0x0A - LTE-M1<br>– 0x0B - LTE-NB1<br>– 0x0C - NR5G |
| *pLTEBandPref-Ext[IN]* | • LTE Band Preference Extended<br>• see nas_lteBandPrefExt for more information |
| *pCiotLteOp-ModePref[IN]* | • CIOT LTE Operational Mode Preference<br>• Bitmask representing the CIoT LTE operational mode preference to be set.<br>• Values :<br>– Bit 0 - LTE wideband<br>– Bit 1 - LTE M1<br>– Bit 2 - LTE NB1 All unlisted bits are reserved for future use and the service point ignores them if used. If this TLV is not present, the modem will use the value read from the NV during bootup. |

| | |
|---|---|
| *pLteM1Band-Pref[IN]* | • Bitmask representing the LTE M1 band preference to be set.<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 22 - E-UTRA Operating Band 23<br>– Bit 23 - E-UTRA Operating Band 24<br>– Bit 24 - E-UTRA Operating Band 25<br>– Bit 25 - E-UTRA Operating Band 26<br>– Bit 27 - E-UTRA Operating Band 28<br>– Bit 28 - E-UTRA Operating Band 29<br>– Bit 29 - E-UTRA Operating Band 32<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– Bit 40 - E-UTRA Operating Band 41<br>– Bit 41 - E-UTRA Operating Band 42<br>– Bit 42 - E-UTRA Operating Band 43<br>– Bit 60 - E-UTRA Operating Band 125<br>– Bit 61 - E-UTRA Operating Band 126<br>– Bit 62 - E-UTRA Operating Band 127 |

- If this field is not present, the modem will use the M1 band preference value read from the NV.

**Parameters**

| | |
|---|---|
| *pLteNb1Band-Pref[IN]* | • Bitmask representing the LTE NB1 band preference to be set.<br>• Bit Values<br>– Bit 0 - E-UTRA Operating Band 1<br>– Bit 1 - E-UTRA Operating Band 2<br>– Bit 2 - E-UTRA Operating Band 3<br>– Bit 3 - E-UTRA Operating Band 4<br>– Bit 4 - E-UTRA Operating Band 5<br>– Bit 5 - E-UTRA Operating Band 6<br>– Bit 6 - E-UTRA Operating Band 7<br>– Bit 7 - E-UTRA Operating Band 8<br>– Bit 8 - E-UTRA Operating Band 9<br>– Bit 9 - E-UTRA Operating Band 10<br>– Bit 10 - E-UTRA Operating Band 11<br>– Bit 11 - E-UTRA Operating Band 12<br>– Bit 12 - E-UTRA Operating Band 13<br>– Bit 13 - E-UTRA Operating Band 14<br>– Bit 16 - E-UTRA Operating Band 17<br>– Bit 17 - E-UTRA Operating Band 18<br>– Bit 18 - E-UTRA Operating Band 19<br>– Bit 19 - E-UTRA Operating Band 20<br>– Bit 20 - E-UTRA Operating Band 21<br>– Bit 22 - E-UTRA Operating Band 23<br>– Bit 23 - E-UTRA Operating Band 24<br>– Bit 24 - E-UTRA Operating Band 25<br>– Bit 25 - E-UTRA Operating Band 26<br>– Bit 27 - E-UTRA Operating Band 28<br>– Bit 28 - E-UTRA Operating Band 29<br>– Bit 29 - E-UTRA Operating Band 32<br>– Bit 32 - E-UTRA Operating Band 33<br>– Bit 33 - E-UTRA Operating Band 34<br>– Bit 34 - E-UTRA Operating Band 35<br>– Bit 35 - E-UTRA Operating Band 36<br>– Bit 36 - E-UTRA Operating Band 37<br>– Bit 37 - E-UTRA Operating Band 38<br>– Bit 38 - E-UTRA Operating Band 39<br>– Bit 39 - E-UTRA Operating Band 40<br>– Bit 40 - E-UTRA Operating Band 41<br>– Bit 41 - E-UTRA Operating Band 42<br>– Bit 42 - E-UTRA Operating Band 43<br>– Bit 60 - E-UTRA Operating Band 125<br>– Bit 61 - E-UTRA Operating Band 126<br>– Bit 62 - E-UTRA Operating Band 127<br>• If this field is not present, the modem will use the NB1 band preference value read from the NV. |

| | |
|---|---|
| *pCiotLteOpMode* | • This TLV is used when the net_sel_pref is MANUAL and RAT TLV is LTE.<br>• This TLV indicates the the operational mode to be used during the manual LTE PLMN selection.<br>• Values :<br>– 0x00 - No service<br>– 0x01 - Camped on LTE wideband<br>– 0x02 - Camped on LTE M1<br>– 0x03 - Camped on LTE NB1 |
| *pCiotAcqOrder-Pref[IN]* | • CIOT Acquisition Order Preference<br>• see nas_ciotAcqOrderPref for more information |
| *pNr5gBandPref[-IN]* | • NR5G Band Preference<br>• see nas_nr5gBandPref for more information |

## 8.372.2 Field Documentation

### 8.372.2.1 struct nas_acqOrderPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pAcqOrderPref

### 8.372.2.2 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pBandPref

### 8.372.2.3 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pChgDuration

### 8.372.2.4 struct nas_ciotAcqOrderPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotAcqOrderPref

### 8.372.2.5 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotLteOpMode

### 8.372.2.6 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pCiotLteOpModePref

### 8.372.2.7 struct nas_CSGID* pack_nas_SLQSSetSysSelectionPrefExt_t::pCSGID

### 8.372.2.8 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pEmerMode

### 8.372.2.9 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pGWAcqOrderPref

### 8.372.2.10 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLTEBandPref

### 8.372.2.11 struct nas_lteBandPrefExt* pack_nas_SLQSSetSysSelectionPrefExt_t::pLTEBandPrefExt

### 8.372.2.12 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLteM1BandPref

### 8.372.2.13 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pLteNb1BandPref

### 8.372.2.14 uint8_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pMNCIncPCSDigStat

### 8.372.2.15 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pModePref

### 8.372.2.16 struct nas_netSelectionPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pNetSelPref

### 8.372.2.17 struct nas_nr5gBandPref* pack_nas_SLQSSetSysSelectionPrefExt_t::pNr5gBandPref

#### 8.372.2.18 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pPRLPref

#### 8.372.2.19 unsigned char* pack_nas_SLQSSetSysSelectionPrefExt_t::pRAT

#### 8.372.2.20 uint16_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pRoamPref

#### 8.372.2.21 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pSrvDomainPref

#### 8.372.2.22 uint32_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pSrvRegRestriction

#### 8.372.2.23 uint64_t* pack_nas_SLQSSetSysSelectionPrefExt_t::pTdscdmaBandPref

## 8.373 pack_nas_SLQSSwiPSDetach_t Struct Reference

### Data Fields

- uint8_t * pDetachAction

### 8.373.1 Detailed Description

Structure for PS connection detach pack.

**Parameters**

| | |
|---|---|
| *pDetachAction[IN]* | • Values<br>– 2- Initiates an immediate packet domain detach. |

### 8.373.2 Field Documentation

#### 8.373.2.1 uint8_t* pack_nas_SLQSSwiPSDetach_t::pDetachAction

## 8.374 pack_pds_PDSInjectTimeReference_t Struct Reference

### Data Fields

- uint64_t systemTime
- uint16_t systemDiscontinuities

### 8.374.1 Detailed Description

Structure contain the parameter for PDS Inject time.

**Parameters**

| | |
|---|---|
| *systemTime* | • System time( milliseconds ) |
| *systemDiscontinuities* | • Number of system time discontinuities |

### 8.374.2 Field Documentation

#### 8.374.2.1 uint16_t pack_pds_PDSInjectTimeReference_t::systemDiscontinuities

#### 8.374.2.2 uint64_t pack_pds_PDSInjectTimeReference_t::systemTime

## 8.375 pack_pds_ResetPDSData_t Struct Reference

### Data Fields

- uint32_t ∗ pGPSDataMask
- uint32_t ∗ pCellDataMask

### 8.375.1 Detailed Description

Structure contain the parameter to Resets the specified PDS data.

**Parameters**

| | |
|---|---|
| *pGPSDataMask* | • Bitmask of GPS data to clear (optional)<br>– 0x00000001 - EPH<br>– 0x00000002 - ALM<br>– 0x00000004 - POS<br>– 0x00000008 - TIME<br>– 0x00000010 - IONO<br>– 0x00000020 - UTC<br>– 0x00000040 - HEALTH<br>– 0x00000080 - SVDIR<br>– 0x00000100 - SVSTEER<br>– 0x00000200 - SADATA<br>– 0x00000400 - RTI<br>– 0x00000800 - ALM_CORR<br>– 0x00001000 - FREQ_BIAS_EST |
| *pCellDataMask* | • Bitmask of cell data to clear (optional)<br>– 0x00000001 - POS<br>– 0x00000002 - LATEST_GPS_POS<br>– 0x00000004 - OTA_POS<br>– 0x00000008 - EXT_REF_POS<br>– 0x00000010 - TIMETAG<br>– 0x00000020 - CELLID<br>– 0x00000040 - CACHED_CELLID<br>– 0x00000080 - LAST_SRV_CELL<br>– 0x00000100 - CUR_SRV_CELL<br>– 0x00000200 - NEIGHBOR_INFO |

### 8.375.2 Field Documentation

#### 8.375.2.1 uint32_t* pack_pds_ResetPDSData_t::pCellDataMask

#### 8.375.2.2 uint32_t* pack_pds_ResetPDSData_t::pGPSDataMask

## 8.376 pack_pds_SetEventReportCallback_t Struct Reference

### Data Fields

- uint8_t posDataNmea
- uint8_t rptPosData

### 8.376.1 Detailed Description

Structure contain the parameter for the automatic tracking configuration for the NMEA COM port.

**Parameters**

| | |
|---|---|
| *posDataNmea* | • Report new position data in NMEA format.<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *rptPosData* | • Report new position data in raw format.<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable |

### 8.376.2 Field Documentation

#### 8.376.2.1 uint8_t pack_pds_SetEventReportCallback_t::posDataNmea

#### 8.376.2.2 uint8_t pack_pds_SetEventReportCallback_t::rptPosData

## 8.377 pack_pds_SetPDSDefaults_t Struct Reference

### Data Fields

- uint32_t operation
- uint8_t timeout
- uint32_t interval
- uint32_t accuracy

### 8.377.1 Detailed Description

Structure containing session configuration parameters.

**Parameters**

| | |
|---|---|
| *operation* | • Current session operating mode<br>– 0 - Standalone<br>– 1 - MS based<br>– 2 - MS assisted |
| *timeout* | • Maximum amount of time (seconds) to work on each fix, maximum is 255 |
| *interval* | • Interval (seconds) between fix requests |
| *accuracy* | • Preferred accuracy threshold (meters) |

### 8.377.2 Field Documentation

#### 8.377.2.1 uint32_t pack_pds_SetPDSDefaults_t::accuracy

#### 8.377.2.2 uint32_t pack_pds_SetPDSDefaults_t::interval

#### 8.377.2.3 uint32_t pack_pds_SetPDSDefaults_t::operation

#### 8.377.2.4 uint8_t pack_pds_SetPDSDefaults_t::timeout

## 8.378 pack_pds_SetPDSState_t Struct Reference

### Data Fields

- uint32_t enable

### 8.378.1 Detailed Description

Structure contain PDS state parameters.

**Parameters**

| | |
|---|---|
| *enable* | • Desired PDS state<br>– Zero - disable<br>– Non-Zero - enable |

### 8.378.2 Field Documentation

#### 8.378.2.1 uint32_t pack_pds_SetPDSState_t::enable

## 8.379 pack_pds_SetPortAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t bAuto

### 8.379.1 Detailed Description

Structure contain the parameter for the automatic tracking configuration for the NMEA COM port.

**Parameters**

| | |
|---|---|
| *bAuto* | • Enable automatic tracking for NMEA COM port<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

### 8.379.2 Field Documentation

#### 8.379.2.1 uint32_t pack_pds_SetPortAutomaticTracking_t::bAuto

## 8.380 pack_pds_SetServiceAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t bAuto

### 8.380.1 Detailed Description

Structure contain the parameter to Sets the automatic tracking state.

**Parameters**

| | |
|---|---|
| *bAuto* | • Automatic tracking session started for service<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

### 8.380.2 Field Documentation

#### 8.380.2.1 uint32_t pack_pds_SetServiceAutomaticTracking_t::bAuto

## 8.381 pack_pds_SetXTRAAutomaticDownload_t Struct Reference

### Data Fields

- uint32_t bEnabled
- uint16_t interval

### 8.381.1 Detailed Description

Structure contain the parameter to set the XTRA automatic database download configuration.

**Parameters**

| | |
|---|---|
| *bEnabled* | • Automatic XTRA download status<br>– 0 - Disabled<br>– 1 - Enabled |
| *interval* | • Interval (hours) between XTRA downloads |

### 8.381.2 Field Documentation

#### 8.381.2.1 uint32_t pack_pds_SetXTRAAutomaticDownload_t::bEnabled

#### 8.381.2.2 uint16_t pack_pds_SetXTRAAutomaticDownload_t::interval

## 8.382 pack_pds_SetXTRANetwork_t Struct Reference

### Data Fields

- uint32_t preference

### 8.382.1 Detailed Description

Structure contain the parameter to Sets the XTRA WWAN network preference

**Parameters**

| | |
|---|---|
| *preference* | • XTRA WWAN network preference<br>– 0x00 - None (any available network)<br>– 0x01 - Home-only, only when on home systems<br>– 0x02 - Roam-only, only when on non-home systems |

### 8.382.2 Field Documentation

#### 8.382.2.1 uint32_t pack_pds_SetXTRANetwork_t::preference

## 8.383 pack_pds_SLQSGetAGPSConfig_t Struct Reference

### Data Fields

- uint8_t * pNetworkMode

### 8.383.1 Detailed Description

Structure contain the parameter to PDS AGPS (MS-based) configuration pack.

**Parameters**

| | |
|---|---|
| *pNetworkMode* | • Network Mode of AGPS Server [optional - should be present in Multimode Systems]<br>– 0x00 - UMTS<br>– 0x01 - CDMA |

### 8.383.2 Field Documentation

#### 8.383.2.1 uint8_t* pack_pds_SLQSGetAGPSConfig_t::pNetworkMode

## 8.384 pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference

### Data Fields

- uint64_t timeMsec
- uint32_t timeUncMsec
- uint8_t timeBase
- uint8_t forceFlag

### 8.384.1 Detailed Description

Structure contain the parameter to Injects a absolute time reference.

**Parameters**

| | |
|---|---|
| *timeMsec* | • Represents the number of milliseconds elapsed since either a GPS or UTC time base. If the time base is UTC, this value should NOT include leap seconds |
| *timeUncMsec* | • Time uncertainty in milliseconds |
| *timeBase* | • Time base<br>– 0x00 - GPS (midnight, Jan 6, 1980)<br>– 0x01 - UTC (midnight, Jan 1, 1970) |
| *forceFlag* | • Force acceptance of data |

### 8.384.2 Field Documentation

#### 8.384.2.1 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::forceFlag

#### 8.384.2.2 uint8_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeBase

#### 8.384.2.3 uint64_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeMsec

#### 8.384.2.4 uint32_t pack_pds_SLQSPDSInjectAbsoluteTimeReference_t::timeUncMsec

## 8.385 pack_pds_SLQSPDSInjectPositionData_t Struct Reference

### Data Fields

- uint64_t * pTimeStamp
- uint64_t * pLatitude
- uint64_t * pLongitude
- uint32_t * pAltitudeWrtEllipsoid
- uint32_t * pAltitudeWrtSealevel
- uint32_t * pHorizontalUncCircular
- uint32_t * pVerticalUnc
- uint8_t * pHorizontalConfidence
- uint8_t * pVerticalConfidence
- uint8_t * pPositionSource
- uint8_t * pTimeType

### 8.385.1 Detailed Description

Position Data Parameters from the external source to be injected to PDS engine.

**Parameters**

| | |
|---|---|
| *pTimeStamp* | • Timestamp of the injected position in msec. The time can be of type UTC, GPS, or Age and is defined in the pTimeType parameter. If the pTimeType is not present, the timestamp shall be assumed to be UTC time |
| *pLatitude* | • Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator. Value (in decimal degrees) in the range from -90 degrees to +90 degrees.Value in double float format (refer toIEEE Std 754-1985) |
| *pLongitude* | • Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of Greenwich meridian. Value (in decimal degrees) in the range from -180 degrees to +180 degrees. |
| *pAltitudeWrt-Ellipsoid* | • Height above the WGS-84 reference ellipsoid. Value conveys height (in meters). When injecting altitude information, the control point should include either this parameter or the pAltitudeWrt-Sealevel parameter. Value in single float format (refer to IEEE Std 754-1985) |
| *pAltitudeWrt-Sealevel* | • Height of MS above the mean sea level in units (in meters). When injecting altitude information, the control point should include either this parameter or the pAltitudeWrtEllipsoid parameter. Value in single float format (refer to IEEE Std 754-1985) |
| *pHorizontalUnc-Circular* | • Circular horizontal uncertainty (in meters). This parameter must be included if the latitude and longitude parameters are specified. Value in single float format (refer to IEEE Std 754-1985) |

| | |
|---|---|
| *pVerticalUnc* | • Vertical uncertainty (in meters). This parameter must be included if one of the altitude parameter are specified.Value in single float format (refer to IEEE Std 754-1985) |
| *pHorizontal-Confidence* | • Confidence value of the location horizontal uncertainty, specified as percentage, 1 to 100. This parameter must be included if the latitude and longitude parameters are specified. |
| *pVertical-Confidence* | • Confidence value of the location vertical uncertainty, specified as percentage, 1 to 100. This parameter must be included if one of the altitude paramters are specified. |
| *pPositionSource* | • Source of injected position:<br>– 0x00 - Unknown<br>– 0x01 - GPS<br>– 0x02 - Cell ID<br>– 0x03 - Enhanced cell ID<br>– 0x04 - WiFi<br>– 0x05 - Terrestrial<br>– 0x06 - Terrestrial hybrid<br>– 0x07 - Other |
| *pTimeType* | • Defines the time value set in the pTimeStamp parameter.<br>– 0x00 - UTC Time: starting Jan 1, 1970<br>– 0x01 - GPS Time: starting Jan 6, 1980<br>– 0x02 - Age: Age of position information |

### 8.385.2 Field Documentation

#### 8.385.2.1 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtEllipsoid

#### 8.385.2.2 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pAltitudeWrtSealevel

#### 8.385.2.3 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalConfidence

#### 8.385.2.4 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pHorizontalUncCircular

#### 8.385.2.5 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLatitude

#### 8.385.2.6 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pLongitude

#### 8.385.2.7 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pPositionSource

#### 8.385.2.8 uint64_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeStamp

#### 8.385.2.9 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pTimeType

#### 8.385.2.10 uint8_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalConfidence

#### 8.385.2.11 uint32_t* pack_pds_SLQSPDSInjectPositionData_t::pVerticalUnc

## 8.386 pack_pds_SLQSSetAGPSConfig_t Struct Reference

### Data Fields

- uint32_t * pServerAddress
- uint32_t * pServerPort
- uint8_t * pServerURL
- uint8_t * pServerURLLength
- uint8_t * pNetworkMode

### 8.386.1 Detailed Description

Structure contain the parameter to Sets the PDS AGPS (MS-based) configuration.

**Parameters**

| | |
|---|---|
| *pServerAddress* | • IPv4 address of AGPS server [optional] |
| *pServerPort* | • Port number of AGPS server [optional - should be present when pServerAddress is present] |
| *pServerURL* | • URL of the AGPS server [optional] |
| *pServerURL-Length* | • URL length of AGPS server [optional - should be present when pServerURL is present] |
| *pNetworkMode* | • Network Mode of AGPS Server [optional - should be present in Multimode Systems]<br>– 0x00 - UMTS<br>– 0x01 - CDMA |

### 8.386.2 Field Documentation

#### 8.386.2.1 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pNetworkMode

#### 8.386.2.2 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerAddress

#### 8.386.2.3 uint32_t* pack_pds_SLQSSetAGPSConfig_t::pServerPort

#### 8.386.2.4 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURL

#### 8.386.2.5 uint8_t* pack_pds_SLQSSetAGPSConfig_t::pServerURLLength

## 8.387 pack_pds_SLQSSetPositionMethodState_t Struct Reference

### Data Fields

- uint8_t * pXtraTimeState

- uint8_t * pXtraDataState
- uint8_t * pWifiState

### 8.387.1 Detailed Description

Parameters to Set state of positioning method for a device.

**Parameters**

| | |
|---|---|
| *pXtraTimeState* | • XTRA Time Position Method State.<br>• Values:<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pXtraDataState* | • XTRA Data Position Method State.<br>• Values:<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *Latitude* | • WiFi Position Method State<br>• Values:<br>– 0x00 - Disable<br>– 0x01 - Enable |

### 8.387.2 Field Documentation

#### 8.387.2.1 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pWifiState

#### 8.387.2.2 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraDataState

#### 8.387.2.3 uint8_t* pack_pds_SLQSSetPositionMethodState_t::pXtraTimeState

## 8.388 pack_pds_StartPDSTrackingSessionExt_t Struct Reference

### Data Fields

- uint8_t sessionControl
- uint8_t sessionType
- uint8_t sessionOperation
- uint8_t sessionServerOption
- uint8_t fixTimeout
- uint32_t fixInterval
- uint32_t fixCount
- uint32_t fixAccuracy

### 8.388.1 Detailed Description

Structure contain the parameter for PDS tracking session.

**Parameters**

| | |
|---|---|
| *sessionControl* | • Control method:<br>– 0x0 - Manual |
| *sessionType* | • Type:<br>– 0x0 - New |
| *session-Operation* | • Operating mode:<br>– 0x00 - Standalone<br>– 0x01 - MS-based |
| *sessionServer-Option* | • Location server option:<br>– 0x0 - Default |
| *fixTimeout* | • Maximum time to work on each fix (in seconds, max 255) |
| *fixCount* | • Count of position fix requests for this session (must be at least 1) |
| *fixInterval* | • interval between position fix requests (in seconds) |
| *fixAccuracy* | • Preferred accuracy threshold(in meters) |

### 8.388.2 Field Documentation

#### 8.388.2.1 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixAccuracy

#### 8.388.2.2 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixCount

#### 8.388.2.3 uint32_t pack_pds_StartPDSTrackingSessionExt_t::fixInterval

#### 8.388.2.4 uint8_t pack_pds_StartPDSTrackingSessionExt_t::fixTimeout

#### 8.388.2.5 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionControl

#### 8.388.2.6 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionOperation

#### 8.388.2.7 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionServerOption

#### 8.388.2.8 uint8_t pack_pds_StartPDSTrackingSessionExt_t::sessionType

## 8.389 pack_qmi_t Struct Reference

### Data Fields

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

### 8.389.1 Detailed Description

qmi request context

**Parameters**

| | | |
|---|---|---|
| in | *xid* | transaction id |
| out | *timeout* | timeout recommended in seconds |
| out | *msgid* | message id |
| out | *svc* | qmi service |

### 8.389.2 Field Documentation

#### 8.389.2.1 uint16_t pack_qmi_t::msgid

#### 8.389.2.2 uint8_t pack_qmi_t::svc

#### 8.389.2.3 int pack_qmi_t::timeout

#### 8.389.2.4 uint16_t pack_qmi_t::xid

## 8.390 pack_qos_BindDataPort_t Struct Reference

### Data Fields

- qos_BindDataPortPeripheralEndPointID_t * pPeripheralEndPointID
- qos_BindDataPortMuxID_t * pMuxID
- qos_BindDataPortSIODataPort_t * pSIODataPort

### 8.390.1 Detailed Description

Structure that contains the request Binds a control point to a data port.

**Parameters**

| | |
|---|---|
| *pPeripheralEndPointID* | • See qos_BindDataPortPeripheralEndPointID_t for more information. |
| *pMuxID* | • See qos_BindDataPortMuxID_t for more information. |
| *pSIODataPort* | • See qos_BindDataPortSIODataPort_t for more information. |

### 8.390.2 Field Documentation

#### 8.390.2.1 qos_BindDataPortMuxID_t* pack_qos_BindDataPort_t::pMuxID

#### 8.390.2.2 qos_BindDataPortPeripheralEndPointID_t* pack_qos_BindDataPort_t::pPeripheralEndPointID

#### 8.390.2.3 qos_BindDataPortSIODataPort_t* pack_qos_BindDataPort_t::pSIODataPort

## 8.391 pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

### Data Fields

- uint32_t apnId

### 8.391.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

**Parameters**

| | |
|---|---|
| *apnId[IN]* | • APN id |

### 8.391.2 Field Documentation

#### 8.391.2.1 uint32_t pack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

## 8.392 pack_qos_SLQSQosSwiReadDataStats_t Struct Reference

### Data Fields

- uint32_t apnId

### 8.392.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

**Parameters**

| | |
|---|---|
| *apnId[IN]* | • APN id |

### 8.392.2 Field Documentation

#### 8.392.2.1 uint32_t pack_qos_SLQSQosSwiReadDataStats_t::apnId

## 8.393 pack_qos_SLQSSetQosEventCallback_t Struct Reference

### Data Fields

- uint8_t enable

### 8.393.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

**Parameters**

| | | |
|---|---|---|
| in | *enable* | • 1 - Enable QoS event reporting<br>• 0 - Disable QoS event reporting |

### 8.393.2 Field Documentation

#### 8.393.2.1 uint8_t pack_qos_SLQSSetQosEventCallback_t::enable

## 8.394 pack_rms_SetSMSWake_t Struct Reference

### Data Fields

- uint32_t enabled
- uint32_t wake_mask

### 8.394.1 Detailed Description

Pack set SMS wake parameters

**Parameters**

| | |
|---|---|
| *enabled* | • SMS wake functionality enabled<br>– 0 - Disabled<br>– 1 - Enabled |
| *wake_mask* | • SMS wake mask to search for incoming messages (only relevant when enabled) |

### 8.394.2 Field Documentation

#### 8.394.2.1 uint32_t pack_rms_SetSMSWake_t::enabled

#### 8.394.2.2 uint32_t pack_rms_SetSMSWake_t::wake_mask

## 8.395 pack_sar_SLQSSetRfSarState_t Struct Reference

### Data Fields

- uint32_t RfSarState

### 8.395.1 Detailed Description

This sturcture contain sets the specified RF SAR state parameter.

**Parameters**

| | |
|---|---|
| *RfSarState* | • SAR RF State<br>– QMI_SAR_RF_STATE_DEFAULT<br>– QMI_SAR_RF_STATE_1<br>– QMI_SAR_RF_STATE_2<br>– QMI_SAR_RF_STATE_3<br>– QMI_SAR_RF_STATE_4<br>– QMI_SAR_RF_STATE_5<br>– QMI_SAR_RF_STATE_6<br>– QMI_SAR_RF_STATE_7<br>– QMI_SAR_RF_STATE_8 |

### 8.395.2 Field Documentation

#### 8.395.2.1 uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState

## 8.396 pack_sms_SaveSMS_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage

### 8.396.1 Detailed Description

Sutructure containing parameters to Save an SMS message to device memory

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *messageFormat* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize* | • The length of the message contents in bytes |

| | |
|---|---|
| *pMessage* | • The message contents |

### 8.396.2 Field Documentation

#### 8.396.2.1 uint32_t pack_sms_SaveSMS_t::messageFormat

#### 8.396.2.2 uint32_t pack_sms_SaveSMS_t::messageSize

#### 8.396.2.3 uint8_t* pack_sms_SaveSMS_t::pMessage

#### 8.396.2.4 uint32_t pack_sms_SaveSMS_t::storageType

## 8.397 pack_sms_SendSMS_t Struct Reference

### Data Fields

- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage
- uint8_t * pLinktimer

### 8.397.1 Detailed Description

This structure contains pack send SMS parameters.

**Parameters**

| | |
|---|---|
| *messageFormat* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize* | • The length of the message contents in bytes |
| *pLinktimer* | • GW SMS link open for the specified number of second |
| *pMessage* | • The message contents in PDU format contains SMS header and payload message |

### 8.397.2 Field Documentation

#### 8.397.2.1 uint32_t pack_sms_SendSMS_t::messageFormat

#### 8.397.2.2 uint32_t pack_sms_SendSMS_t::messageSize

#### 8.397.2.3 uint8_t* pack_sms_SendSMS_t::pLinktimer

#### 8.397.2.4 uint8_t* pack_sms_SendSMS_t::pMessage

## 8.398 pack_sms_SetNewSMSCallback_t Struct Reference

### Data Fields

- enum eqmiCbkSetStatus status

### 8.398.1 Detailed Description

This strucure contains pack set new SMS callback parameters.

**Parameters**

| | |
|---|---|
| *status* | callback parameter |

### 8.398.2 Field Documentation

#### 8.398.2.1 enum eqmiCbkSetStatus pack_sms_SetNewSMSCallback_t::status

## 8.399 pack_sms_SetSMSCAddress_t Struct Reference

### Data Fields

- uint8_t * pSMSCAddress
- uint8_t * pSMSCType

### 8.399.1 Detailed Description

Structure to Set the SMS center address.

**Parameters**

| | |
|---|---|
| *SMSCAddress* | • The SMS center address represented as a NULL terminated string |
| *SMSCType* | • The SMS center address type represented as a NULL terminated string (optional). |

### 8.399.2 Field Documentation

#### 8.399.2.1 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCAddress

#### 8.399.2.2 uint8_t* pack_sms_SetSMSCAddress_t::pSMSCType

## 8.400 pack_sms_SLQSDeleteSMS_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t * pMessageIndex
- uint32_t * pMessageTag
- uint8_t * pMessageMode

### 8.400.1 Detailed Description

This structure contains pack delete SMS parameters.

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *pMessageIndex* | • (Optional) message index |
| *pMessageTag* | • (Optional) message tag<br>– 0 - Read<br>– 1 - Not read<br>– 2 - Mobile originated and sent<br>– 3 - Mobile originated but not yet sent |
| *pMessageMode* | • (Optional) message mode<br>• this must be included if the device is capable of supporting more than one protocol<br>• e.g. CDMA and GW<br>– 0x00 - CDMA, LTE (if network type is CDMA)<br>– 0x01 - GW, LTE (if network type is UMTS) |

### 8.400.2 Field Documentation

#### 8.400.2.1 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageIndex

#### 8.400.2.2 uint8_t* pack_sms_SLQSDeleteSMS_t::pMessageMode

#### 8.400.2.3 uint32_t* pack_sms_SLQSDeleteSMS_t::pMessageTag

#### 8.400.2.4 uint32_t pack_sms_SLQSDeleteSMS_t::storageType

## 8.401 pack_sms_SLQSGetSMS_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint8_t * pMessageMode

### 8.401.1 Detailed Description

This sturcture contains pack get SMS parameters.

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *messageIndex* | • Message index |
| *pMessageMode* | • 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

### 8.401.2 Field Documentation

#### 8.401.2.1 uint32_t pack_sms_SLQSGetSMS_t::messageIndex

#### 8.401.2.2 uint8_t∗ pack_sms_SLQSGetSMS_t::pMessageMode

#### 8.401.2.3 uint32_t pack_sms_SLQSGetSMS_t::storageType

## 8.402 pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

### Data Fields

- uint8_t mode

### 8.402.1 Detailed Description

This structure contains get SMS boardcast configure parameter.

**Parameters**

| | |
|---|---|
| *mode* | • Mode<br>– 0x00 - CDMA, LTE (if network type is CDMA)<br>– 0x01 - GW, LTE (if network type is UMTS) |

### 8.402.2 Field Documentation

#### 8.402.2.1 uint8_t pack_sms_SLQSGetSmsBroadcastConfig_t::mode

## 8.403 pack_sms_SLQSGetSMSList_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t * pRequestedTag
- uint8_t * pMessageMode

### 8.403.1 Detailed Description

This structure contains pack get SMS list parameters.

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *requestedTag* | • (Optional) Message tag<br>– 0 - Read<br>– 1 - Not read<br>– 2 - Mobile originated and sent<br>– 3 - Mobile originated but not yet sent |
| *messageMode* | • 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

### 8.403.2 Field Documentation

#### 8.403.2.1 uint8_t* pack_sms_SLQSGetSMSList_t::pMessageMode

#### 8.403.2.2 uint32_t* pack_sms_SLQSGetSMSList_t::pRequestedTag

#### 8.403.2.3 uint32_t pack_sms_SLQSGetSMSList_t::storageType

## 8.404 pack_sms_SLQSModifySMSStatus_t Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageIndex
- uint32_t messageTag
- uint8_t * pMessageMode

### 8.404.1 Detailed Description

This structure contains pack modify SMS status parameters.

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *messageIndex* | • Message index |
| *messageTag* | • Message tag<br>– 0 - Read<br>– 1 - Not read |
| *pMessageMode* | • 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

### 8.404.2 Field Documentation

#### 8.404.2.1 uint32_t pack_sms_SLQSModifySMSStatus_t::messageIndex

#### 8.404.2.2 uint32_t pack_sms_SLQSModifySMSStatus_t::messageTag

#### 8.404.2.3 uint8_t* pack_sms_SLQSModifySMSStatus_t::pMessageMode

#### 8.404.2.4 uint32_t pack_sms_SLQSModifySMSStatus_t::storageType

## 8.405 pack_sms_SLQSSendAsyncSMS_t Struct Reference

### Data Fields

- sms_sendAsyncsmsParams * pSendSmsParams

### 8.405.1 Detailed Description

Structure contain Parameter to Send an SMS message for immediate over-the-air transmission

**Parameters**

| | |
|---|---|
| *pSendSms-Params* | • structure containing the SMS parameters. Refer sms_sendasyncsmsparams |

### 8.405.2 Field Documentation

#### 8.405.2.1 sms_sendAsyncsmsParams* pack_sms_SLQSSendAsyncSMS_t::pSendSmsParams

## 8.406 pack_sms_SLQSSetIndicationRegister_t Struct Reference

### Data Fields

- sms_setIndicationReg * pSetIndicationRegReq

### 8.406.1 Detailed Description

Structure contain Parameter that sets the registration state of different WMS indications.

**Parameters**

| | |
|---|---|
| *pSetIndication-Reg* | • Pointer to structure of indicationRegReqParams<br>– See sms_setIndicationReg for more information |

### 8.406.2 Field Documentation

#### 8.406.2.1 sms_setIndicationReg* pack_sms_SLQSSetIndicationRegister_t::pSetIndicationRegReq

## 8.407 pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

### Data Fields

- uint8_t mode
- uint8_t broadcastActivate

### 8.407.1 Detailed Description

Structure contain parameters that Enables or disables the reception of broadcast SMS messages.

**Parameters**

| | |
|---|---|
| *Mode* | • Mode<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |
| *broadcast-Activate* | • 0x00 - Disable broadcast<br>• 0x01 - Activate broadcast |

### 8.407.2 Field Documentation

#### 8.407.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::broadcastActivate

#### 8.407.2.2 uint8_t pack_sms_SLQSSetSmsBroadcastActivation_t::mode

## 8.408 pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

### Data Fields

- uint8_t mode
- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig

### 8.408.1 Detailed Description

Structure containing parameters that provides the information about the SMS BroadcastConfiguration

**Parameters**

| | |
|---|---|
| *mode* | • Mode<br>– 0x00 - CDMA, LTE (if network type is CDMA)<br>– 0x01 - GW, LTE (if network type is UMTS) |
| *pBroadcast-Config* | • The data for 3GPP Broadcast Information(Optional). |
| *pCDMA-BroadcastConfig* | • The data for 3GPP2 Broadcast Information(Optional). |

### 8.408.2 Field Documentation

#### 8.408.2.1 uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode

#### 8.408.2.2 sms_qaQmi3GPPBroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig

#### 8.408.2.3 sms_qaQmi3GPP2BroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig

## 8.409 pack_sms_SLQSSetSmsStorage_t Struct Reference

### Data Fields

- uint8_t smsStorage

### 8.409.1 Detailed Description

Structure contain Parameter to set the SMS Storage on the device

**Parameters**

| | |
|---|---|
| *smsStorage* | • SMS Storage<br>– 0x01 - device's permanent memory<br>– 0x02 - UICC |

### 8.409.2 Field Documentation

#### 8.409.2.1 uint8_t pack_sms_SLQSSetSmsStorage_t::smsStorage

## 8.410 pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

### Data Fields

- sms_maxStorageSizeReq * pMaxStorageSizeReq

### 8.410.1 Detailed Description

This structure contains get maximum storage size.

**Parameters**

| | |
|---|---|
| *pMaxStorage-SizeReq* | • Request parameters for SmsSLQSGetMaxStorageSize<br>– See sms_maxStorageSizeReq for more information |

### 8.410.2 Field Documentation

#### 8.410.2.1 sms_maxStorageSizeReq* pack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeReq

## 8.411 pack_sms_SLQSSmsSetRoutes_t Struct Reference

### Data Fields

- sms_setRoutesReq * pSetRoutesReq

### 8.411.1 Detailed Description

Structure contain parameter that Sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports.

**Parameters**

| | |
|---|---|
| *pSetRoutesReq* | • Pointer to structure of smsSetRoutesReq<br>– See sms_setRoutesReq for more information |

### 8.411.2 Field Documentation

#### 8.411.2.1 sms_setRoutesReq* pack_sms_SLQSSmsSetRoutes_t::pSetRoutesReq

## 8.412 pack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference

### Data Fields

- uint8_t * pGenerator

### 8.412.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile pack parameters.

**Parameters**

| | |
|---|---|
| *pGenerator[optional]* | • Generator<br>– 0 - Voice |

### 8.412.2 Field Documentation

#### 8.412.2.1 uint8_t* pack_swiaudio_SLQSGetM2MAudioProfile_t::pGenerator

## 8.413 pack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t Generator

### 8.413.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-5 |
| *Generator* | • Generator<br>– 0 - Voice |

### 8.413.2 Field Documentation

#### 8.413.2.1 uint8_t pack_swiaudio_SLQSGetM2MAudioVolume_t::Generator

#### 8.413.2.2 uint8_t pack_swiaudio_SLQSGetM2MAudioVolume_t::Profile

## 8.414 pack_swiaudio_SLQSGetM2MAVMute_t Struct Reference

### Data Fields

- uint8_t Profile

### 8.414.1 Detailed Description

This structure contains the SLQSGetM2MAVMute pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |

### 8.414.2 Field Documentation

#### 8.414.2.1 uint8_t pack_swiaudio_SLQSGetM2MAVMute_t::Profile

## 8.415 pack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference

### Data Fields

- uint8_t Profile

### 8.415.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |

### 8.415.2 Field Documentation

#### 8.415.2.1 uint8_t pack_swiaudio_SLQSGetM2MSpkrGain_t::Profile

## 8.416 pack_swiaudio_SLQSSetM2MAudioAVCFG_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t Device
- uint8_t PIFACEId
- swiaudio_PCMparams * pPCMParams

### 8.416.1 Detailed Description

This structure contains the SLQSSetM2MAudioAVCFG pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0-5 |

| | |
|---|---|
| *Device* | • ACDB Device<br>• See qaGobiApiTableSwiAudio.h for more information on ACDB Device |
| *PIFACEId* | • Physical Interface<br>• See qaGobiApiTableSwiAudio.h for more information on physical interface |
| *pPCMParams* | • PCM parameters<br>• See swiaudio_PCMparams for more information |

### 8.416.2 Field Documentation

#### 8.416.2.1 uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::Device

#### 8.416.2.2 uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::PIFACEId

#### 8.416.2.3 swiaudio_PCMparams* pack_swiaudio_SLQSSetM2MAudioAVCFG_t::pPCMParams

#### 8.416.2.4 uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::Profile

## 8.417 pack_swiaudio_SLQSSetM2MAudioLPBK_t Struct Reference

### Data Fields

- uint8_t Enable

### 8.417.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK pack parameters.

**Parameters**

| | |
|---|---|
| *Enable* | • Operation to be performed<br>– 0 - stop<br>– 1 - VOCODER loop<br>– 2 - internal codec loop |

### 8.417.2 Field Documentation

#### 8.417.2.1 uint8_t pack_swiaudio_SLQSSetM2MAudioLPBK_t::Enable

## 8.418 pack_swiaudio_SLQSSetM2MAudioProfile_t Struct Reference

### Data Fields

- uint8_t Profile

- uint8_t * pEarMute
- uint8_t * pMicMute
- uint8_t * pGenerator
- uint8_t * pVolume
- uint8_t * pCwtMute

### 8.418.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |
| *pEarMute* | • Ear Mute<br>– 0 - mute<br>– 1 - unmute |
| *pMicMute* | • Mic Mute<br>– 0 - mute<br>– 1 - unmute |
| *pGenerator* | • Generator<br>– 0 - voice |
| *pVolume* | • Set RX Volume level<br>– 0-5 |
| *pCwtMute* | • Call Waiting Tone Mute<br>– 0 - Mute<br>– 1 - UnMute |

### 8.418.2 Field Documentation

#### 8.418.2.1 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pCwtMute

#### 8.418.2.2 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pEarMute

#### 8.418.2.3 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pGenerator

#### 8.418.2.4 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pMicMute

#### 8.418.2.5 uint8_t pack_swiaudio_SLQSSetM2MAudioProfile_t::Profile

#### 8.418.2.6 uint8_t* pack_swiaudio_SLQSSetM2MAudioProfile_t::pVolume

## 8.419 pack_swiaudio_SLQSSetM2MAudioVolume_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t Generator
- uint8_t Level

### 8.419.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |
| *Generator* | • Generator<br>– 0 - voice |
| *Level* | • Audio volume level<br>– 0-5 |

### 8.419.2 Field Documentation

#### 8.419.2.1 uint8_t pack_swiaudio_SLQSSetM2MAudioVolume_t::Generator

#### 8.419.2.2 uint8_t pack_swiaudio_SLQSSetM2MAudioVolume_t::Level

#### 8.419.2.3 uint8_t pack_swiaudio_SLQSSetM2MAudioVolume_t::Profile

## 8.420 pack_swiaudio_SLQSSetM2MAVMute_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t EarMute
- uint8_t MicMute
- uint8_t * pCwtMute

### 8.420.1 Detailed Description

This structure contains the SLQSSetM2MAVMute pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |

| | |
|---|---|
| *EarMute* | • Ear Mute<br>– 0-1 |
| *MicMute* | • Mic Mute<br>– 0-1 |
| *pCwtMute* | [ Optional ]<br>• Call Waiting Tone Mute<br>– 0-1 |

### 8.420.2 Field Documentation

#### 8.420.2.1 uint8_t pack_swiaudio_SLQSSetM2MAVMute_t::EarMute

#### 8.420.2.2 uint8_t pack_swiaudio_SLQSSetM2MAVMute_t::MicMute

#### 8.420.2.3 uint8_t* pack_swiaudio_SLQSSetM2MAVMute_t::pCwtMute

#### 8.420.2.4 uint8_t pack_swiaudio_SLQSSetM2MAVMute_t::Profile

## 8.421 pack_swiaudio_SLQSSetM2MSpkrGain_t Struct Reference

### Data Fields

- uint8_t Profile
- uint16_t Value

### 8.421.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain pack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile Number<br>– 0-5 |
| *Value* | • RX speakerphone gain<br>– 0x0 - 0x7fff |

### 8.421.2 Field Documentation

#### 8.421.2.1 uint8_t pack_swiaudio_SLQSSetM2MSpkrGain_t::Profile

#### 8.421.2.2 uint16_t pack_swiaudio_SLQSSetM2MSpkrGain_t::Value

## 8.422 pack_swiavms_SLQSAVMSSendSelection_t Struct Reference

### Data Fields

- uint8_t selection
- uint32_t * pDeferTime
- uint8_t * pClientPerformOperationFlag
- uint8_t * pPackageID
- uint16_t * pRejectReason

### 8.422.1 Detailed Description

Structure containing the AVMS selection

**Parameters**

| | |
|---|---|
| *selection[IN]* | • User Selection<br>– 0x01 - Accept<br>– 0x02 - Reject<br>– 0x03 - Defer |
| *pDeferTime[IN/-OPTIONAL]* | • Defer time in minutes. A value of 0 will cause the prompt to be resent immediately. |
| *pClientPerform-OperationFlag[I-N/OPTIONAL]* | • Client operation flag after accept.<br>– 0: if modem performs the operation (download or update)<br>– 1: if client performs the operation (download or update) |
| *pPackageID[IN/-OPTIONAL]* | • Package ID. |
| *pRejectReason[-IN/OPTIONAL]* | • Reject Reason. |

### 8.422.2 Field Documentation

#### 8.422.2.1 uint8_t* pack_swiavms_SLQSAVMSSendSelection_t::pClientPerformOperationFlag

#### 8.422.2.2 uint32_t* pack_swiavms_SLQSAVMSSendSelection_t::pDeferTime

#### 8.422.2.3 uint8_t* pack_swiavms_SLQSAVMSSendSelection_t::pPackageID

#### 8.422.2.4 uint16_t* pack_swiavms_SLQSAVMSSendSelection_t::pRejectReason

#### 8.422.2.5 uint8_t pack_swiavms_SLQSAVMSSendSelection_t::selection

## 8.423 pack_swiavms_SLQSAVMSSetSettings_t Struct Reference

### Data Fields

- uint8_t AutoConnect
- uint8_t AutoReboot
- uint8_t PromptFwDownload
- uint8_t PromptFwUpdate
- uint8_t * pFwAutoSDM
- uint32_t * pPollingTimer
- PackSwiAvmsSetSettingsConnectionRetryTimers * pConnectionRetryTimers
- PackSwiAvmsSetSettingsAPNInfo * pAPNInfo
- uint8_t * pNotifStore
- PackSwiAvmsSetSettingsPeriodInfo * pPeriodInfo

### 8.423.1 Detailed Description

Structure containing the AVMS settings to be set on the device This maps to structure SLQSAVMSSetSettings

**Parameters**

| | |
|---|---|
| *AutoConnect[IN]* | • Auto Connect to AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *AutoReboot[IN]* | • Automatic device reboot when the request is received from AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *PromptFw-Download[IN]* | • Firmware Auto Download<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *PromptFw-Update[IN]* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware autoupdate FALSE<br>– 0x01 - Firmware autoupdate TRUE |
| *pFwAutoSDM[IN/OPTIONAL]* | • OMA Automatic UI Alert Response<br>– 0x00 - DISABLED<br>– 0x01 - ENABLED ACCEPT<br>– 0x02 - ENABLED REJECT |
| *pPollingTimer[IN/OPTIONAL]* | • Polling timer to connect to AVMS server<br>– 0-525600 (min)<br>– 0:disabled |
| *pConnection-RetryTimers[IN/-OPTIONAL]* | • See PackSwiAvmsSetSettingsConnectionRetryTimers for more information |

| | |
|---|---|
| *pAPNInfo[IN/OPTIONAL]* | • See PackSwiAvmsSetSettingsAPNInfo for more information |
| *pNotifStore[IN/OPTIONAL]* | • See PackSwiAvmsSetSettingsAPNInfo for more information |
| *pPeriodInfo[IN/OPTIONAL]* | • See PackSwiAvmsSetSettingsAPNInfo for more information |

### 8.423.2 Field Documentation

#### 8.423.2.1 uint8_t pack_swiavms_SLQSAVMSSetSettings_t::AutoConnect

#### 8.423.2.2 uint8_t pack_swiavms_SLQSAVMSSetSettings_t::AutoReboot

#### 8.423.2.3 PackSwiAvmsSetSettingsAPNInfo* pack_swiavms_SLQSAVMSSetSettings_t::pAPNInfo

#### 8.423.2.4 PackSwiAvmsSetSettingsConnectionRetryTimers* pack_swiavms_SLQSAVMSSetSettings_t::pConnectionRetryTimers

#### 8.423.2.5 uint8_t* pack_swiavms_SLQSAVMSSetSettings_t::pFwAutoSDM

#### 8.423.2.6 uint8_t* pack_swiavms_SLQSAVMSSetSettings_t::pNotifStore

#### 8.423.2.7 PackSwiAvmsSetSettingsPeriodInfo* pack_swiavms_SLQSAVMSSetSettings_t::pPeriodInfo

#### 8.423.2.8 uint32_t* pack_swiavms_SLQSAVMSSetSettings_t::pPollingTimer

#### 8.423.2.9 uint8_t pack_swiavms_SLQSAVMSSetSettings_t::PromptFwDownload

#### 8.423.2.10 uint8_t pack_swiavms_SLQSAVMSSetSettings_t::PromptFwUpdate

## 8.424 pack_swiavms_SLQSAVMSSetSettings_v2_t Struct Reference

### Data Fields

- uint8_t AutoConnect
- uint8_t PromptFwDownload
- uint8_t PromptFwUpdate
- uint8_t * pFwAutoSDM
- uint32_t * pPollingTimer
- PackSwiAvmsSetSettingsConnectionRetryTimers * pConnectionRetryTimers
- PackSwiAvmsSetSettingsAPNInfo * pAPNInfo
- uint8_t * pNotifStore
- PackSwiAvmsSetSettingsPeriodInfo * pPeriodInfo
- uint8_t * pAutoReboot

### 8.424.1 Detailed Description

Structure containing the AVMS settings to be set on the device This maps to structure SLQSAVMSSetSettings_v2 (For AVC2 service)

**Parameters**

| | |
|---|---|
| *AutoConnect[IN]* | • Auto Connect to AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *PromptFw-Download[IN]* | • Firmware Auto Download<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *PromptFw-Update[IN]* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware autoupdate FALSE<br>– 0x01 - Firmware autoupdate TRUE |
| *pFwAutoSDM[IN/OPTIONAL]* | • OMA Automatic UI Alert Response<br>– 0x00 - DISABLED<br>– 0x01 - ENABLED ACCEPT<br>– 0x02 - ENABLED REJECT |
| *pPollingTimer[IN/OPTIONAL]* | • Polling timer to connect to AVMS server<br>– 0-525600 (min)<br>– 0:disabled |
| *pConnection-RetryTimers[IN/-OPTIONAL]* | • See PackSwiAvmsSetSettingsConnectionRetryTimers for more information |
| *pAPNInfo[IN/OPTIONAL]* | • See PackSwiAvmsSetSettingsAPNInfo for more information |
| *pNotifStore[IN/-OPTIONAL]* | • See PackSwiAvmsSetSettingsAPNInfo for more information |
| *pPeriodInfo[IN/-OPTIONAL]* | • See PackSwiAvmsSetSettingsAPNInfo for more information |
| *pAutoReboot[IN/OPTIONAL]* | • Automatic device reboot when the request is received from AirVantage server<br>– 0x00 - FALSE<br>– 0x01 - TRUE |

**Note**

Setting PromptFwDownload/PromptFwUpdate as TRUE implies AutoConnect to be also TRUE even if AutoConnect was set FALSE initially. Automatic firmware download/update selection trumps the connection setting flag.

### 8.424.2 Field Documentation

#### 8.424.2.1 uint8_t pack_swiavms_SLQSAVMSSetSettings_v2_t::AutoConnect

#### 8.424.2.2 PackSwiAvmsSetSettingsAPNInfo* pack_swiavms_SLQSAVMSSetSettings_v2_t::pAPNInfo

#### 8.424.2.3 uint8_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pAutoReboot

#### 8.424.2.4 PackSwiAvmsSetSettingsConnectionRetryTimers* pack_swiavms_SLQSAVMSSetSettings_v2_t::pConnectionRetryTimers

#### 8.424.2.5 uint8_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pFwAutoSDM

#### 8.424.2.6 uint8_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pNotifStore

#### 8.424.2.7 PackSwiAvmsSetSettingsPeriodInfo* pack_swiavms_SLQSAVMSSetSettings_v2_t::pPeriodInfo

#### 8.424.2.8 uint32_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pPollingTimer

#### 8.424.2.9 uint8_t pack_swiavms_SLQSAVMSSetSettings_v2_t::PromptFwDownload

#### 8.424.2.10 uint8_t pack_swiavms_SLQSAVMSSetSettings_v2_t::PromptFwUpdate

## 8.425 pack_swiavms_SLQSAVMSStartSession_t Struct Reference

### Data Fields

- uint8_t sessionType

### 8.425.1 Detailed Description

Structure that contains the session type request for AVMS start session command

**Parameters**

| | |
|---|---|
| *sessionType[IN]* | • Session type<br>– 0x01 - FOTA, to check availability of FW Update. This field is mandatory in OMA-DM case but not necessary in LWM2M one. To keep compatibility, this shall be used in LWM2M without any specific treatment. |

### 8.425.2 Field Documentation

#### 8.425.2.1 uint8_t pack_swiavms_SLQSAVMSStartSession_t::sessionType

## 8.426 pack_swiavms_SLQSAVMSStopSession_t Struct Reference

### Data Fields

- uint8_t sessionType

### 8.426.1 Detailed Description

Structure that contains the session type for AVMS Stop session command

**Parameters**

| | |
|---|---|
| *sessionType* | • Session type<br>– 0x01 - FOTA, to suspend FOTA session.<br>– 0xFF - Suspend ongoing FOTA session or stop any other active AVMS session. |
| *resultcode* | • Tlv Result Code |

### 8.426.2 Field Documentation

#### 8.426.2.1 uint8_t pack_swiavms_SLQSAVMSStopSession_t::sessionType

## 8.427 pack_swidms_SLQSSwiDmsSetMTU_t Struct Reference

### Data Fields

- uint16_t MTUSize

### 8.427.1 Detailed Description

This structure contains the Set MTU request parameter.

**Parameters**

| | |
|---|---|
| *MTUSize* | • Set MTU size<br>• MTU Values<br>– 0 - use default value<br>– 576 to 2000 - other values required by carrier.<br>– Set the same MTU for all RAT/interfaces.<br>– New MTU size effective after modem reboot. |

### 8.427.2 Field Documentation

#### 8.427.2.1 uint16_t pack_swidms_SLQSSwiDmsSetMTU_t::MTUSize

## 8.428 pack_swidms_SLQSSwiDmsSetUsbNetNum_t Struct Reference

### Data Fields

- uint8_t nUsbNetNum

### 8.428.1 Detailed Description

This structure contains usb net number to set on remote endpoint for QMAP configuration

**Parameters**

| | |
|---|---|
| *nUsbNetNum* | • value of usb net numbers to set for the device<br>• range is 0-8 |

### 8.428.2 Field Documentation

#### 8.428.2.1 uint8_t pack_swidms_SLQSSwiDmsSetUsbNetNum_t::nUsbNetNum

## 8.429 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

### 8.429.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

**Parameters**

| | |
|---|---|
| *function* | • Setting to indicate when modem should start an automatic GNSS fix<br>– 0 - disabled<br>– 1 - At bootup<br>– 2 - When NMEA port is opened |
| *set_function* | • 0 - do not set to modem<br>• I - set to modem |
| *fix_type* | • Type of GNSS fix:<br>– 1 - Default Engine mode<br>– 2 - MS-Based<br>– 3 - MS-Assisted<br>– 4 - Standalone |
| *set_fix_type* | • 0 - do not set to modem<br>• I - set to modem |

| | |
|---|---|
| *max_time* | • Maximum time allowed for the receiver to get a fix in seconds<br>• Valid range: 1-255 |
| *set_max_time* | • 0 - do not set to modem<br>• I - set to modem |
| *max_dist* | • Maximum uncertainty of a fix measured by distance in meters<br>• Valid range: 1 - 4294967280 |
| *set_max_dist* | • 0 - do not set to modem<br>• I - set to modem |
| *fix_rate* | • Time between fixes in seconds<br>• Valid range: 1–65535 |
| *set_fix_rate* | • 0 - do not set to modem<br>• I - set to modem |

### 8.429.2 Field Documentation

#### 8.429.2.1 uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate

#### 8.429.2.2 uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type

#### 8.429.2.3 uint8_t pack_swiloc_SwiLocSetAutoStart_t::function

#### 8.429.2.4 uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist

#### 8.429.2.5 uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time

#### 8.429.2.6 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate

#### 8.429.2.7 int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type

#### 8.429.2.8 int pack_swiloc_SwiLocSetAutoStart_t::set_function

#### 8.429.2.9 int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist

#### 8.429.2.10 int pack_swiloc_SwiLocSetAutoStart_t::set_max_time

## 8.430 pack_swioma_SLQSOMADMCancelSession_t Struct Reference

### Data Fields

- uint32_t sessionType

### 8.430.1 Detailed Description

Structure that contains the session type for OMA cancel session command

**Parameters**

| | |
|---|---|
| *sessionType* | [in]<br>• Session type<br>– 0x01 - FOTA, to check availability of FW Update<br>– 0xFF - Cancel any active OMADM session |

### 8.430.2 Field Documentation

#### 8.430.2.1 uint32_t pack_swioma_SLQSOMADMCancelSession_t::sessionType

## 8.431 pack_swioma_SLQSOMADMCancelSessionExt_t Struct Reference

### Data Fields

- uint32_t sessionType

### 8.431.1 Detailed Description

Structure that contains the session type for OMA cancel session command

**Parameters**

| | |
|---|---|
| *sessionType* | [IN]<br>• Session type<br>– 0x01 - FOTA, to check availability of FW Update<br>– 0xFF - Cancel any active OMADM session |

### 8.431.2 Field Documentation

#### 8.431.2.1 uint32_t pack_swioma_SLQSOMADMCancelSessionExt_t::sessionType

## 8.432 pack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference

### Data Fields

- uint32_t SessionType

### 8.432.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *SessionType* | [in]<br>• Session type<br>– 0x01 - FOTA<br>– 0xFF - Any active OMADM session. If no active sessions are available, then previous OMADM session info is returned |

### 8.432.2 Field Documentation

#### 8.432.2.1 uint32_t pack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

## 8.433 pack_swioma_SLQSOMADMSendSelection_t Struct Reference

### Data Fields

- uint32_t selection
- uint32_t ∗ pDeferTime
- uint32_t ∗ pRejectReason

### 8.433.1 Detailed Description

Structure containing the OMA DM selection

**Parameters**

| | |
|---|---|
| *selection* | [in]<br>• OMA-DM NIA Selection<br>– 0x01 - Accept<br>– 0x02 - Reject<br>– 0x03 - Defer |
| *pDeferTime* | [in]<br>• Defer time in minutes. A value of 0 will cause the prompt to be resent immediately.<br>• This TLV is mandatory if selection is set to 0x03. |
| *pRejectReason* | [in]<br>• Reject Reason<br>• This TLV is processed if selection is set to 0x02. If it is not present, the reject reason 0 is used as default. |

### 8.433.2 Field Documentation

#### 8.433.2.1 uint32_t∗ pack_swioma_SLQSOMADMSendSelection_t::pDeferTime

#### 8.433.2.2 uint32_t∗ pack_swioma_SLQSOMADMSendSelection_t::pRejectReason

#### 8.433.2.3 uint32_t pack_swioma_SLQSOMADMSendSelection_t::selection

## 8.434 pack_swioma_SLQSOMADMSendSelectionExt_t Struct Reference

### Data Fields

- uint32_t selection

### 8.434.1 Detailed Description

Structure containing the OMA DM selection

**Parameters**

| | |
|---|---|
| *selection* | [IN]<br>• OMA-DM NIA Selection<br>– 0x01 - Accept<br>– 0x02 - Reject<br>– 0x03 - Defer |

### 8.434.2 Field Documentation

#### 8.434.2.1 uint32_t pack_swioma_SLQSOMADMSendSelectionExt_t::selection

## 8.435 pack_swioma_SLQSOMADMSetSettings_t Struct Reference

### Data Fields

- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t * pAutosdm
- uint8_t * pFwAutoCheck

### 8.435.1 Detailed Description

Structure containing the OMA DM settings to be set on the device This maps to structure SLQSOMADMSettings-ReqParams3

**Parameters**

| | |
|---|---|
| *FOTAdownload* | • 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Firmware autodownload FALSE<br>– 0x01 - Firmware autodownload TRUE |
| *FOTAUpdate* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware autoupdate FALSE<br>– 0x01 - Firmware autoupdate TRUE |
| *pAutosdm* | • Optional 1 byte parameter indicating OMA Automatic UI Alert Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled Accept<br>– 0x02 - Enabled Reject |

| | |
|---|---|
| *pFwAutoCheck* | • Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

### 8.435.2 Field Documentation

#### 8.435.2.1 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAdownload

#### 8.435.2.2 uint8_t pack_swioma_SLQSOMADMSetSettings_t::FOTAUpdate

#### 8.435.2.3 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pAutosdm

#### 8.435.2.4 uint8_t* pack_swioma_SLQSOMADMSetSettings_t::pFwAutoCheck

## 8.436 pack_swioma_SLQSOMADMSetSettingsExt_t Struct Reference

### Data Fields

- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t OMADMEnable
- uint8_t OMADMLogEnable
- uint8_t FUMOEnable
- uint8_t PRLEnable

### 8.436.1 Detailed Description

Structure containing the OMA DM settings to be set on the device This maps to structure SLQSOMADMSetSettings-Ext

**Parameters**

| | |
|---|---|
| *FOTAdownload* | • 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Firmware autodownload FALSE<br>– 0x01 - Firmware autodownload TRUE |
| *FOTAUpdate* | • 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - Firmware autoupdate FALSE<br>– 0x01 - Firmware autoupdate TRUE |
| *OMADMEnable* | [IN]<br>• 1 byte parameter indicating OMA DM<br>– 0x00 - OMA DM Disabled<br>– 0x01 - OMA DM Enabled |

| | |
|---|---|
| *OMADMLog-Enable* | [IN]<br>• 1 byte parameter indicating OMA DM Logs<br>– 0x00 - OMA DM Logs Disabled<br>– 0x01 - OMA DM Logs Enabled |
| *FUMOEnable* | [IN]<br>• 1 byte parameter indicating FUMO enabled<br>– 0x00 - FUMO Disabled<br>– 0x01 - FUMO enabled |
| *PRLEnable* | [IN]<br>• 1 byte parameter indicating PRL Enabled<br>– 0x00 - PRL Disabled<br>– 0x01 - PRL Enabled<br>– 0x02 - Launch a CI PRL<br>– 0x03 - PRL Enabled and PRL automatic update every 45 days<br>– 0x04 - PRL Enabled and PRL update ever 90 days |

### 8.436.2 Field Documentation

#### 8.436.2.1 uint8_t pack_swioma_SLQSOMADMSetSettingsExt_t::FOTAdownload

#### 8.436.2.2 uint8_t pack_swioma_SLQSOMADMSetSettingsExt_t::FOTAUpdate

#### 8.436.2.3 uint8_t pack_swioma_SLQSOMADMSetSettingsExt_t::FUMOEnable

#### 8.436.2.4 uint8_t pack_swioma_SLQSOMADMSetSettingsExt_t::OMADMEnable

#### 8.436.2.5 uint8_t pack_swioma_SLQSOMADMSetSettingsExt_t::OMADMLogEnable

#### 8.436.2.6 uint8_t pack_swioma_SLQSOMADMSetSettingsExt_t::PRLEnable

## 8.437 pack_swioma_SLQSOMADMStartSession_t Struct Reference

### Data Fields

- uint32_t sessionType

### 8.437.1 Detailed Description

Structure that contains the session type for OMA start session command

**Parameters**

| | |
|---|---|
| *sessionType[IN]* | • Session type<br>– 0x01 - FOTA, to check availability of FW Update<br>– 0x02 - DM, to check availability of DM Update<br>– 0x03 - PRL, to check availability of PRL Update |

### 8.437.2 Field Documentation

#### 8.437.2.1 uint32_t pack_swioma_SLQSOMADMStartSession_t::sessionType

## 8.438 pack_swioma_SLQSOMADMStartSessionExt_t Struct Reference

### Data Fields

- uint32_t sessionType

### 8.438.1 Detailed Description

Structure that contains the session type for OMA start session command

**Parameters**

| | |
|---|---|
| *sessionType* | [IN]<br>• Session type<br>– 0x01 - FOTA, to check availability of FW Update<br>– 0x02 - DM, to check availability of DM Update<br>– 0x03 - PRL, to check availability of PRL Update |

### 8.438.2 Field Documentation

#### 8.438.2.1 uint32_t pack_swioma_SLQSOMADMStartSessionExt_t::sessionType

## 8.439 pack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference

### Data Fields

- uint8_t mitigationDevIDLen
- char mitigationDevID [255]

### 8.439.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevID-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

### 8.439.2 Field Documentation

#### 8.439.2.1 char pack_tmd_SLQSTmdDeRegNotMitigationLvl_t::mitigationDevID[255]

#### 8.439.2.2 uint8_t pack_tmd_SLQSTmdDeRegNotMitigationLvl_t::mitigationDevIDLen

## 8.440 pack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference

### Data Fields

- uint8_t mitigationDevIDLen
- char mitigationDevID [255]

### 8.440.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevID-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

### 8.440.2 Field Documentation

#### 8.440.2.1 char pack_tmd_SLQSTmdGetMitigationLvl_t::mitigationDevID[255]

#### 8.440.2.2 uint8_t pack_tmd_SLQSTmdGetMitigationLvl_t::mitigationDevIDLen

## 8.441 pack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference

### Data Fields

- uint8_t mitigationDevIDLen
- char mitigationDevID [255]

### 8.441.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *mitigationDevID-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevID* | • Mitigation device ID |

### 8.441.2 Field Documentation

#### 8.441.2.1 char pack_tmd_SLQSTmdRegNotMitigationLvl_t::mitigationDevID[255]

#### 8.441.2.2 uint8_t pack_tmd_SLQSTmdRegNotMitigationLvl_t::mitigationDevIDLen

## 8.442 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

### 8.442.1 Detailed Description

This structure contains information of the request parameters associated with a Change PIN API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *changePIN* | • See uim_changeUIMPIN for more information. |
| *pKeyReferenceID(optional)* | • Indicates the PIN key reference ID.<br>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.<br>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases. |
| *pIndicationToken(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• Valid Values<br>– 0 - Result of operation in response. Indication will not be generated by the modem<br>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function |

**Note**

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

### 8.442.2 Field Documentation

#### 8.442.2.1 uim_changeUIMPIN pack_uim_ChangePin_t::changePIN

#### 8.442.2.2 uim_encryptedPIN1 pack_uim_ChangePin_t::EncryptedPIN1

#### 8.442.2.3 uint32_t* pack_uim_ChangePin_t::pIndicationToken

#### 8.442.2.4 uint8_t* pack_uim_ChangePin_t::pKeyReferenceID

#### 8.442.2.5 uim_sessionInformation pack_uim_ChangePin_t::sessionInfo

#### 8.442.2.6 uint16_t pack_uim_ChangePin_t::Tlvresult

## 8.443 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

### 8.443.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *fileIndex* | • See uim_fileInfo for more information. |
| *readTransparent* | • See uim_readTransparentInfo for more information. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• Valid Values<br>– 0 - Result of operation in response. Indication will not be generated by the modem<br>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function |
| *pEncrypt-Data(optional)* | • Encrypt Data.<br>• Indicates whether the data read from the card is to be encrypted. |

**Note**

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

### 8.443.2 Field Documentation

#### 8.443.2.1 uim_fileInfo pack_uim_ReadTransparent_t::fileIndex

#### 8.443.2.2 uint8_t* pack_uim_ReadTransparent_t::pEncryptData

#### 8.443.2.3 uint32_t* pack_uim_ReadTransparent_t::pIndicationToken

#### 8.443.2.4 uim_readTransparentInfo pack_uim_ReadTransparent_t::readTransparent

#### 8.443.2.5 uim_sessionInformation pack_uim_ReadTransparent_t::sessionInfo

#### 8.443.2.6 uint16_t pack_uim_ReadTransparent_t::Tlvresult

## 8.444 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

### 8.444.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *pinProtection* | • See uim_setPINProtection for more information. |
| *pKeyReferenceID(optional)* | • Indicates the PIN key reference ID.<br>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.<br>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases. |
| *pIndicationToken(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• Valid Values<br>– 0 - Result of operation in response. Indication will not be generated by the modem<br>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function |

**Note**

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

### 8.444.2 Field Documentation

#### 8.444.2.1 uim_encryptedPIN1 pack_uim_SetPinProtection_t::EncryptedPIN1

#### 8.444.2.2 uint32_t* pack_uim_SetPinProtection_t::pIndicationToken

#### 8.444.2.3 uim_setPINProtection pack_uim_SetPinProtection_t::pinProtection

#### 8.444.2.4 uint8_t* pack_uim_SetPinProtection_t::pKeyReferenceID

#### 8.444.2.5 uim_sessionInformation pack_uim_SetPinProtection_t::sessionInfo

#### 8.444.2.6 uint16_t pack_uim_SetPinProtection_t::Tlvresult

## 8.445 pack_uim_SLQSUIMAuthenticate_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uim_authenticationData authData
- uint32_t * pIndicationToken

### 8.445.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *authData* | • See uim_authenticationData for more information. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication. |

**Note**

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

### 8.445.2 Field Documentation

#### 8.445.2.1 uim_authenticationData pack_uim_SLQSUIMAuthenticate_t::authData

#### 8.445.2.2 uint32_t* pack_uim_SLQSUIMAuthenticate_t::pIndicationToken

#### 8.445.2.3 uim_sessionInformation pack_uim_SLQSUIMAuthenticate_t::sessionInfo

## 8.446 pack_uim_SLQSUIMDepersonalization_t Struct Reference

### Data Fields

- uim_depersonalizationInformation depersonilisationInfo

### 8.446.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *depersonilisation-Info* | • See uim_depersonalizationInformation for more information. |

### 8.446.2 Field Documentation

#### 8.446.2.1 uim_depersonalizationInformation pack_uim_SLQSUIMDepersonalization_t::depersonilisationInfo

## 8.447 pack_uim_SLQSUIMEventRegister_t Struct Reference

### Data Fields

- uint32_t eventMask

### 8.447.1 Detailed Description

This structure contains pack event register parameter.

**Parameters**

| | |
|---|---|
| *eventMask* | - bit 1 - card status<br>• bit 4 - physical slot status |

### 8.447.2 Field Documentation

#### 8.447.2.1 uint32_t pack_uim_SLQSUIMEventRegister_t::eventMask

## 8.448 pack_uim_SLQSUIMGetConfiguration_t Struct Reference

### Data Fields

- uint32_t * pConfigurationMask

### 8.448.1 Detailed Description

This structure contains information of the request parameters associated with Get Configuration to get the modem configuration for the UIM module

**Parameters**

| | |
|---|---|
| *pConfiguration-Mask(optional)* | • Requested configurations<br>– Bit 0 - Automatic selection<br>– Bit 1 - Personalization status<br>– Bit 2 - Halt subscription<br>– All other bits are reserved for future use |

**Note**

- if the TLV is missing, the service returns all configuration items in the response.

### 8.448.2 Field Documentation

#### 8.448.2.1 uint32_t* pack_uim_SLQSUIMGetConfiguration_t::pConfigurationMask

## 8.449 pack_uim_SLQSUIMGetFileAttributes_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uim_fileInfo fileIndex
- uint32_t * pIndicationToken

### 8.449.1 Detailed Description

This structure contains information of the request parameters associated with a Get File Attributes

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *fileIndex* | • See uim_fileInfo for more information. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication. |

**Note**

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

### 8.449.2 Field Documentation

#### 8.449.2.1 uim_fileInfo pack_uim_SLQSUIMGetFileAttributes_t::fileIndex

#### 8.449.2.2 uint32_t* pack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

#### 8.449.2.3 uim_sessionInformation pack_uim_SLQSUIMGetFileAttributes_t::sessionInfo

## 8.450 pack_uim_SLQSUIMPowerDown_t Struct Reference

### Data Fields

- uint8_t slot

### 8.450.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *slot* | • Indicates the slot to be used.<br>– 1 - Slot 1<br>– 2 - Slot 2 |

### 8.450.2 Field Documentation

#### 8.450.2.1 uint8_t pack_uim_SLQSUIMPowerDown_t::slot

## 8.451 pack_uim_SLQSUIMPowerUp_t Struct Reference

### Data Fields

- uint8_t slot
- uint8_t * pIgnoreHotSwapSwitch

### 8.451.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *slot* | • Indicates the slot to be used.<br>– 1 - Slot 1<br>– 2 - Slot 2 |
| *pIgnoreHot-Swap-Switch(optional)* | • Hot-swap switch status.<br>– 0 - Checks the hot-swap switch status<br>– 1 - Ignores the hot-swap switch status |

### 8.451.2 Field Documentation

#### 8.451.2.1 uint8_t* pack_uim_SLQSUIMPowerUp_t::pIgnoreHotSwapSwitch

#### 8.451.2.2 uint8_t pack_uim_SLQSUIMPowerUp_t::slot

## 8.452 pack_uim_SLQSUIMRefreshComplete_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uint8_t refreshComplete

### 8.452.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Complete event.

**Parameters**

| | |
|---|---|
| *sessionInfo(-Mandatory)* | • See uim_sessionInformation for more information. |
| *refresh-Complete(-Mandatory)* | • Indicates whether the refresh was successful. Valid values:<br>– 0 - Refresh was not completed successfully<br>– 1 - Refresh was completed successfully |

### 8.452.2 Field Documentation

#### 8.452.2.1 uint8_t pack_uim_SLQSUIMRefreshComplete_t::refreshComplete

#### 8.452.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshComplete_t::sessionInfo

## 8.453 pack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo

### 8.453.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Get Last Event.

**Parameters**

| | |
|---|---|
| *sessionInfo(-Mandatory)* | • See uim_sessionInformation for more information. |

### 8.453.2 Field Documentation

#### 8.453.2.1 uim_sessionInformation pack_uim_SLQSUIMRefreshGetLastEvent_t::sessionInfo

## 8.454 pack_uim_SLQSUIMRefreshOK_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uint8_t OKtoRefresh

### 8.454.1 Detailed Description

This structure contains Parameters of the Session Information

**Parameters**

| | |
|---|---|
| *sessionInfo* | • Session Information<br>• See uim_sessionInformation for more information |
| *OKtoRefresh* | • Indicates whether a refresh is OK. Valid values:<br>– 0 - Not OK to refresh<br>– 1 - OK to refresh |

### 8.454.2 Field Documentation

#### 8.454.2.1 uint8_t pack_uim_SLQSUIMRefreshOK_t::OKtoRefresh

#### 8.454.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshOK_t::sessionInfo

## 8.455 pack_uim_SLQSUIMRefreshRegister_t Struct Reference

### Data Fields

- uim_sessionInformation sessionInfo
- uim_registerRefresh regRefresh

### 8.455.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Register.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • Session Information params<br>• See uim_sessionInformation for more information |
| *regRefresh* | • Register Refresh parameters<br>• See uim_registerRefresh for more information |

### 8.455.2 Field Documentation

#### 8.455.2.1 uim_registerRefresh pack_uim_SLQSUIMRefreshRegister_t::regRefresh

#### 8.455.2.2 uim_sessionInformation pack_uim_SLQSUIMRefreshRegister_t::sessionInfo

## 8.456 pack_uim_SLQSUIMSwitchSlot_t Struct Reference

### Data Fields

- uint8_t bLogicalSlot
- uint32_t ulPhysicalSlot

### 8.456.1 Detailed Description

This structure contains information of the request parameters associated with a Switch Slot.

**Parameters**

| | |
|---|---|
| *bLogicalSlot* | • Indicates the slot to be used.<br>– 1 - Slot 1<br>– 2 - Slot 2<br>– 3 - Slot 3<br>– 4 - Slot 4<br>– 5 - Slot 5 |
| *ulPhysicalSlot* | • 1 - Slot 1<br>• 2 - Slot 2<br>• 3 - Slot 3<br>• 4 - Slot 4<br>• 5 - Slot 5 |

### 8.456.2 Field Documentation

#### 8.456.2.1 uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot

#### 8.456.2.2 uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot

## 8.457 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

### 8.457.1 Detailed Description

This structure contains information of the request parameters associated with a Unblock PIN API.

**Parameters**

| | |
|---|---|
| *EncryptedPIN1* | • See uim_encryptedPIN1 for more information. |
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *pinProtection* | • See uim_unblockUIMPIN for more information. |
| *pKeyReferenceI-D(optional)* | • Indicates the PIN key reference ID.<br>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.<br>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• Valid Values<br>– 0 - Result of operation in response. Indication will not be generated by the modem<br>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function |

### 8.457.2 Field Documentation

#### 8.457.2.1 uim_encryptedPIN1 pack_uim_UnblockPin_t::EncryptedPIN1

#### 8.457.2.2 uint32_t* pack_uim_UnblockPin_t::pIndicationToken

#### 8.457.2.3 uim_unblockUIMPIN pack_uim_UnblockPin_t::pinProtection

#### 8.457.2.4 uint8_t* pack_uim_UnblockPin_t::pKeyReferenceID

#### 8.457.2.5 uim_sessionInformation pack_uim_UnblockPin_t::sessionInfo

#### 8.457.2.6 uint16_t pack_uim_UnblockPin_t::Tlvresult

## 8.458 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

### 8.458.1 Detailed Description

This structure contains information of the request parameters associated with a verify PIN API.

**Parameters**

| | |
|---|---|
| *sessionInfo* | • See uim_sessionInformation for more information. |
| *verifyPIN* | • See uim_verifyUIMPIN for more information. |
| *pEncryptedPIN1(optional)* | • See uim_encryptedPIN1 for more information. |
| *pKeyReferenceID(optional)* | • Indicates the PIN key reference ID.<br>• Indicates the PIN key reference ID. Valid values are from 1 to 8, respectively, for application 1 to application 8.<br>• This TLV is used only for PIN1 and PIN2 and is ignored in all other cases. |
| *pIndicationToken(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• Valid Values<br>– 0 - Result of operation in response. Indication will not be generated by the modem<br>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function |

**Note**

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

### 8.458.2 Field Documentation

#### 8.458.2.1 uim_encryptedPIN1∗ pack_uim_VerifyPin_t::pEncryptedPIN1

#### 8.458.2.2 uint32_t∗ pack_uim_VerifyPin_t::pIndicationToken

#### 8.458.2.3 uint8_t∗ pack_uim_VerifyPin_t::pKeyReferenceID

#### 8.458.2.4 uim_sessionInformation pack_uim_VerifyPin_t::sessionInfo

#### 8.458.2.5 uint16_t pack_uim_VerifyPin_t::Tlvresult

#### 8.458.2.6 uim_verifyUIMPIN pack_uim_VerifyPin_t::verifyPIN

## 8.459 pack_voice_AnswerUSSD_t Struct Reference

### Data Fields

- uint8_t ∗ pInfo

### 8.459.1 Detailed Description

This structure contains pack anser USSD parameter.

**Parameters**

| | |
|---|---|
| *pInfo* | • USS information<br>• See voice_USSInfo for more details |

### 8.459.2 Field Documentation

#### 8.459.2.1 uint8_t* pack_voice_AnswerUSSD_t::pInfo

## 8.460 pack_voice_OriginateUSSD_t Struct Reference

### Data Fields

- uint8_t * pInfo

### 8.460.1 Detailed Description

This structure contains pack orginate USSD parameter.

**Parameters**

| | |
|---|---|
| *pInfo* | • USS information<br>• See voice_USSInfo for more details |

### 8.460.2 Field Documentation

#### 8.460.2.1 uint8_t* pack_voice_OriginateUSSD_t::pInfo

## 8.461 pack_voice_SLQSOriginateUSSD_t Struct Reference

### Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

### 8.461.1 Detailed Description

This structure contains USS Information

**Parameters**

| | |
|---|---|
| *ussDCS* | • 1 - ASCII coding scheme<br>• 2 - 8-BIT coding scheme<br>• 3 - UCS2 |
| *ussLen* | • Range 1 to 182 |
| *ussData* | • Data encoded as per the DCS |

### 8.461.2 Field Documentation

#### 8.461.2.1 uint8_t pack_voice_SLQSOriginateUSSD_t::ussData[182]

#### 8.461.2.2 uint8_t pack_voice_SLQSOriginateUSSD_t::ussDCS

#### 8.461.2.3 uint8_t pack_voice_SLQSOriginateUSSD_t::ussLen

## 8.462 pack_voice_SLQSVoiceALSSelectLine_t Struct Reference

### Data Fields

- uint8_t lineValue

### 8.462.1 Detailed Description

This structure contains ALS Select Line Information Parameters.

**Parameters**

| | |
|---|---|
| *lineValue* | • ALS Line Value.<br>– 0x00 - ALS_LINE1 - Line 1 (default)<br>– 0x01 - ALS_LINE2 - Line 2 |

### 8.462.2 Field Documentation

#### 8.462.2.1 uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue

## 8.463 pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference

### Data Fields

- uint8_t switchOption

### 8.463.1 Detailed Description

This structure contains ALS Set Line Switching Information Parameters.

**Parameters**

| | |
|---|---|
| *switchOption* | • Switch Option.<br>– 0x00 - VOICE_LINE_SWITCHING_NOT_ALLOWED - Line switching is not allowed<br>– 0x01 - VOICE_LINE_SWITCHING_ALLOWED - Line switching is allowed |

### 8.463.2 Field Documentation

#### 8.463.2.1 uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption

## 8.464 pack_voice_SLQSVoiceAnswerCall_t Struct Reference

### Data Fields

- uint8_t ∗ pCallId

### 8.464.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

**Parameters**

| | |
|---|---|
| *pCallId* | • Unique call identifier for the call that must be answered. |

### 8.464.2 Field Documentation

#### 8.464.2.1 uint8_t∗ pack_voice_SLQSVoiceAnswerCall_t::pCallId

## 8.465 pack_voice_SLQSVoiceBindSubscription_t Struct Reference

### Data Fields

- uint8_t subsType

### 8.465.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

**Parameters**

| | |
|---|---|
| *subsType* | • Subscription Type.<br>– 0x00 - VOICE_SUBS_TYPE_PRIMARY - Primary<br>– 0x01 - VOICE_SUBS_TYPE_SECONDARY - Secondary |

### 8.465.2 Field Documentation

#### 8.465.2.1 uint8_t pack_voice_SLQSVoiceBindSubscription_t::subsType

## 8.466 pack_voice_SLQSVoiceBurstDTMF_t Struct Reference

### Data Fields

- voice_burstDTMFInfo BurstDTMFInfo
- voice_DTMFLengths * pBurstDTMFLengths

### 8.466.1 Detailed Description

This structure contains parameters of burst Dual-Tone Multifrequency (DTMF)

**Parameters**

| | |
|---|---|
| *BurstDTMFInfo* | • Burst DTMF Information<br>– See voice_burstDTMFInfo for more information |
| *pBurstDTMF-Lengths* | [optional]<br>• DTMF Lengths<br>– See voice_DTMFLengths for more information<br>• NULL pointer - Invalid data. |

### 8.466.2 Field Documentation

#### 8.466.2.1 voice_burstDTMFInfo pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo

#### 8.466.2.2 voice_DTMFLengths* pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths

## 8.467 pack_voice_SLQSVoiceDialCall_t Struct Reference

### Data Fields

- uint8_t callNumber [81]
- uint8_t * pCallType
- uint8_t * pCLIRType
- voice_UUSInfo * pUUSInFo
- voice_CUGInfo * pCUGInfo
- uint8_t * pEmergencyCategory
- voice_calledPartySubAdd * pCallPartySubAdd
- uint8_t * pSvcType

### 8.467.1 Detailed Description

This structure contains Voice Call Request Parameters

**Parameters**

| | |
|---|---|
| *callNumber[81]* | • Number to be dialed in ASCII string, NULL terminated.<br>• Length Range [1 to 81] |
| *pCall-Type(optional)* | • the type of call to be dialed. CALL_TYPE_VOICE is automatically selected if this parameter is not provided. When CALL_TYPE_NON_STD_OTASP is selected, the call is sent as a nonstandard OTASP call regardless of the digit string Call type values are:<br>– 0x00 - CALL_TYPE_VOICE - Voice (automatic selection)<br>– 0x01 - CALL_TYPE_VOICE_FORCED - Avoid modem call classification<br>– 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP∗<br>– 0x09 - CALL_TYPE_EMERGENCY - Emergency |
| *pCLIR-Type(optional)* | • CLIR type values are:<br>– 0x01 - CLIR_SUPPRESSION - Suppression<br>– 0x02 - CLIR_INVOCATION - Invocation |
| *pUUSIn-Fo(optional)* | • Pointer to structure of UUSInfo<br>– See voice_UUSInfo for more information |
| *pCUG-Info(optional)* | • Pointer to structure of CUGInfo<br>– See voice_CUGInfo for more information |
| *pEmergency-Category(optional)* | • Bit mask of emergency number categories. This is only applicable when the call type is set to Emergency.<br>– Bit 0 - VOICE_EMER_CAT_POLICE_BIT - Police<br>– Bit 1 - VOICE_EMER_CAT_AMBULANCE_BIT - Ambulance<br>– Bit 2 - VOICE_EMER_CAT_FIRE_BRIGADE_BIT- Fire brigade<br>– Bit 3 - VOICE_EMER_CAT_MARINE_GUARD_BIT - Marine guard<br>– Bit 4 - VOICE_EMER_CAT_MOUNTAIN_RESCUE_BIT - Mountain rescue<br>– Bit 5 - VOICE_EMER_CAT_MANUAL_ECALL_BIT - Manual emergency call<br>– Bit 6 - VOICE_EMER_CAT_AUTO_ECALL_BIT - Automatic emergency call<br>– Bit 7 - VOICE_EMER_CAT_SPARE_BIT - Spare bit |
| *pCallPartySub-Add(optional)* | • Pointer to structure of calledPartySubAdd<br>– See voice_calledPartySubAdd for more information |
| *pSvc-Type(optional)* | • Service Type.<br>– 0x01 - VOICE_DIAL_CALL_SRV_TYPE_AUTOMATIC - Automatic<br>– 0x02 - VOICE_DIAL_CALL_SRV_TYPE_GSM - GSM<br>– 0x03 - VOICE_DIAL_CALL_SRV_TYPE_WCDMA - WCDMA<br>– 0x04 - VOICE_DIAL_CALL_SRV_TYPE_CDMA_AUTOMATIC - CDMA automatic<br>– 0x05 - VOICE_DIAL_CALL_SRV_TYPE_GSM_WCDMA - GSM or WCDMA<br>– 0x06 - VOICE_DIAL_CALL_SRV_TYPE_LTE -LTE |

### 8.467.2 Field Documentation

#### 8.467.2.1 uint8_t pack_voice_SLQSVoiceDialCall_t::callNumber[81]

#### 8.467.2.2 voice_calledPartySubAdd* pack_voice_SLQSVoiceDialCall_t::pCallPartySubAdd

#### 8.467.2.3 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCallType

#### 8.467.2.4 uint8_t* pack_voice_SLQSVoiceDialCall_t::pCLIRType

#### 8.467.2.5 voice_CUGInfo* pack_voice_SLQSVoiceDialCall_t::pCUGInfo

#### 8.467.2.6 uint8_t* pack_voice_SLQSVoiceDialCall_t::pEmergencyCategory

#### 8.467.2.7 uint8_t* pack_voice_SLQSVoiceDialCall_t::pSvcType

#### 8.467.2.8 voice_UUSInfo* pack_voice_SLQSVoiceDialCall_t::pUUSInFo

## 8.468 pack_voice_SLQSVoiceEndCall_t Struct Reference

### Data Fields

- uint8_t * pCallId

### 8.468.1 Detailed Description

This structure contain pack voice end call parameter.

**Parameters**

| | |
|---|---|
| *pCallId* | • Unique call identifier for the call that must be ended |

### 8.468.2 Field Documentation

#### 8.468.2.1 uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId

## 8.469 pack_voice_SLQSVoiceGetCallBarring_t Struct Reference

### Data Fields

- uint8_t reason
- uint8_t * pSvcClass

### 8.469.1 Detailed Description

This structure contains Voice Get Call Barring Pack Parameters

**Parameters**

| | |
|---|---|
| *reason* | • Call Barring Reason<br>• Values:<br>– 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing<br>– 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal<br>– 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME - Outgoing external to home<br>– 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming<br>– 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming<br>– 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred<br>– 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred<br>– 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred |
| *pSvcClass* | • Service class is a combination (sum) of information class constants (optional)<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes.<br>• Service Class is set to 0 if call waiting is not active for any of the information classes.<br>• 0xFF,if Not Available |

### 8.469.2 Field Documentation

#### 8.469.2.1 uint8_t∗ pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

#### 8.469.2.2 uint8_t pack_voice_SLQSVoiceGetCallBarring_t::reason

## 8.470 pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

### Data Fields

- uint8_t Reason
- uint8_t ∗ pSvcClass

### 8.470.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Pack Parameters

**Parameters**

| | |
|---|---|
| *Reason* | • Call Forwarding Reason<br>• Values:<br>– 0x01 - QMI_VOICE_REASON_FWDREASON_UNCONDITIONAL - Unconditional call forwarding<br>– 0x02 - QMI_VOICE_REASON_FWDREASON_MOBILEBUSY - Forward when the mobile is busy<br>– 0x03 - QMI_VOICE_REASON_FWDREASON_NOREPLY - Forward when there is no reply<br>– 0x04 - QMI_VOICE_REASON_FWDREASON_UNREACHABLE - Forward when the call is unreachable<br>– 0x05 - QMI_VOICE_REASON_FWDREASON_ALLFORWARDING - All forwarding<br>– 0x06 - QMI_VOICE_REASON_FWDREASON_ALLCONDITIONAL - All conditional forwarding |
| *pSvcClass(optional)* | • Service Class is a combination (sum) of information class constants<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes. |

### 8.470.2 Field Documentation

#### 8.470.2.1 uint8_t* pack_voice_SLQSVoiceGetCallForwardingStatus_t::pSvcClass

#### 8.470.2.2 uint8_t pack_voice_SLQSVoiceGetCallForwardingStatus_t::Reason

## 8.471 pack_voice_SLQSVoiceGetCallInfo_t Struct Reference

### Data Fields

- uint8_t callID

### 8.471.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *callID* | • Call identifier for the call queried for information. |

### 8.471.2 Field Documentation

#### 8.471.2.1 uint8_t pack_voice_SLQSVoiceGetCallInfo_t::callID

## 8.472 pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

### Data Fields

- uint8_t * pSvcClass

### 8.472.1 Detailed Description

This structure contains Voice Get Call Waiting Response Parameters

**Parameters**

| | |
|---|---|
| *pSvcClass* | • Service class is a combination (sum) of information class constants (optional)<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes.<br>• Service Class is set to 0 if call waiting is not active for any of the information classes.<br>• 0xFF,if Not Available |

### 8.472.2 Field Documentation

#### 8.472.2.1 uint8_t* pack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

## 8.473 pack_voice_SLQSVoiceGetConfig_t Struct Reference

### Data Fields

- uint8_t * pAutoAnswer
- uint8_t * pAirTimer
- uint8_t * pRoamTimer
- uint8_t * pTTYMode
- uint8_t * pPrefVoiceSO
- uint8_t * pAMRStatus
- uint8_t * pPrefVoicePrivacy
- uint8_t * pNamID
- uint8_t * pVoiceDomainPref

### 8.473.1 Detailed Description

This structure contains Voice Get Configuration Request Parameters

**Parameters**

| | |
|---|---|
| *pAutoAnswer(optional)* | • Indicator to retrieve the Auto Answer Information.<br>  – 0x01 - Include auto answer information |
| *pAirTimer(optional)* | • Indicator to retrieve the Air Timer Information.<br>  – 0x01 - Include air calls timer count information<br>• Currently Not Supported. |
| *pRoamTimer(optional)* | • Indicator to retrieve the Roam Timer Information.<br>  – 0x01 - Include roam calls timer information<br>• Currently Not Supported. |

| | |
|---|---|
| *pTTY-Mode(optional)* | • Indicator to retrieve the TTY Mode Information.<br>– 0x01 - Include TTY configuration status information |
| *pPrefVoiceS-O(optional)* | • Indicator to retrieve the Preferred Voice SO Information.<br>– 0x01 - Include preferred voice configuration status information<br>• Currently Not Supported. |
| *pAMR-Status(optional)* | • Indicator to retrieve the AMR Status Information.<br>– 0x01 - Include AMR status information |
| *pPrefVoice-Privacy(optional)* | • Indicator to retrieve the Preferred Voice Privacy Information.<br>– 0x01 - Include preferred voice privacy status information |
| *pNamI-D(optional)* | • Index of the Number Assignment Module Index (CDMA subscription) to be configured<br>• Range: 0 to 3.<br>• Some modems support only 1 or 2 NAMs.<br>• The NAM Index is valid only when the request contains at least one of Air Timer, Roam Timer, and Preferred Voice SO.<br>• If no nam_id value is specified in the request,the default value is 0. |
| *pVoiceDomain-Pref(optional)* | • Indicator to retrieve the Preferred Voice Domain Information.<br>– 0x01 - Include voice domain preference information |

**Note**

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

### 8.473.2 Field Documentation

#### 8.473.2.1 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAirTimer

#### 8.473.2.2 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAMRStatus

#### 8.473.2.3 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pAutoAnswer

#### 8.473.2.4 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pNamID

#### 8.473.2.5 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoicePrivacy

#### 8.473.2.6 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pPrefVoiceSO

#### 8.473.2.7 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pRoamTimer

#### 8.473.2.8 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pTTYMode

#### 8.473.2.9 uint8_t* pack_voice_SLQSVoiceGetConfig_t::pVoiceDomainPref

## 8.474 pack_voice_SLQSVoiceIndicationRegister_t Struct Reference

### Data Fields

- uint8_t * pRegDTMFEvents
- uint8_t * pRegVoicePrivacyEvents
- uint8_t * pSuppsNotifEvents

### 8.474.1 Detailed Description

This structure contains parameters of Indication Register Information

**Parameters**

| | |
|---|---|
| *pRegDTMF-Events(optional)* | • Registration Indication For DTMF Events.<br>• When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_DTMF_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pRegVoice-Privacy-Events(optional)* | • Registration Indication For Voice Privacy Events.<br>• When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_PRIVACY_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pSuppsNotif-Events(optional)* | • Registration Indication For Supplementary Service Notification Events.<br>• When this registration is enabled, the device learns of DTMF events via the QMI_VOICE_SUPS_NOTIFICATION_IND indication.<br>– 0x00 - Disable<br>– 0x01 - Enable |

**Note**

One of the optional parameter is mandatory to be present in the request.

### 8.474.2 Field Documentation

#### 8.474.2.1 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegDTMFEvents

#### 8.474.2.2 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pRegVoicePrivacyEvents

#### 8.474.2.3 uint8_t* pack_voice_SLQSVoiceIndicationRegister_t::pSuppsNotifEvents

## 8.475 pack_voice_SLQSVoiceManageCalls_t Struct Reference

### Data Fields

- uint8_t SUPSType

- uint8_t * pCallID

### 8.475.1 Detailed Description

This structure contains Manage Calls Information.

**Parameters**

| | |
|---|---|
| *SUPSType* | • Supplementary service type during the call.<br>– 0x01 - SUPS_TYPE_RELEASE_HELD_OR_WAITING<br>* Release is held or waiting<br>– 0x02 - SUPS_TYPE_RELEASE_ACTIVE_ACCEPT_HELD_OR_WAITING<br>* Release is active and accepting held or waiting<br>– 0x03 - SUPS_TYPE_HOLD_ACTIVE_ACCEPT_WAITING_OR_HELD<br>* Hold is active and accepting waiting or held<br>– 0x04 - SUPS_TYPE_HOLD_ALL_EXCEPT_SPECIFIED_CALL<br>* Hold all calls except a specified one<br>– 0x05 - SUPS_TYPE_MAKE_CONFERENCE_CALL<br>* Make a conference call<br>– 0x06 - SUPS_TYPE_EXPLICIT_CALL_TRANSFER<br>* Explicit call transfer<br>– 0x07 - SUPS_TYPE_CCBS_ACTIVATION<br>* Activate completion of calls to busy subscriber<br>– 0x08 - SUPS_TYPE_END_ALL_CALLS<br>* End all calls<br>– 0x09 - SUPS_TYPE_RELEASE_SPECIFIED_CALL<br>* Release a specified call |
| *pCallID[Optional]* | • Applicable only for SUPSType 0x04, 0x07, and 0x09<br>• NULL pointer - Invalid data. |

### 8.475.2 Field Documentation

#### 8.475.2.1 uint8_t* pack_voice_SLQSVoiceManageCalls_t::pCallID

#### 8.475.2.2 uint8_t pack_voice_SLQSVoiceManageCalls_t::SUPSType

## 8.476 pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference

### Data Fields

- struct voice_USSInfo USSInformation

### 8.476.1 Detailed Description

This structure contains Orig USSD No Wait Information Parameters.

**Parameters**

| | |
|---|---|
| *USSInformation* | • See voice_USSInfo for more information. |

### 8.476.2 Field Documentation

#### 8.476.2.1 struct voice_USSInfo pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation

## 8.477 pack_voice_SLQSVoiceSendFlash_t Struct Reference

### Data Fields

- uint8_t ∗ pCallID
- uint8_t ∗ pFlashPayLd
- uint8_t ∗ pFlashType

### 8.477.1 Detailed Description

This structure contains the flash information associated with a call.

**Parameters**

| | |
|---|---|
| *pCallID* | • Unique call identifier associated with the current call. |
| *pFlashPayLd(optional)* | • Payload in ASCII to be sent in Flash.<br>• Variable Length, NULL terminated. |
| *pFlashType(optional)* | • Flash type.<br>– 0 - Simple Flash (default)<br>– 1 - Activate answer hold<br>– 2 - Deactivate answer hold |

### 8.477.2 Field Documentation

#### 8.477.2.1 uint8_t∗ pack_voice_SLQSVoiceSendFlash_t::pCallID

#### 8.477.2.2 uint8_t∗ pack_voice_SLQSVoiceSendFlash_t::pFlashPayLd

#### 8.477.2.3 uint8_t∗ pack_voice_SLQSVoiceSendFlash_t::pFlashType

## 8.478 pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

### Data Fields

- uint8_t Reason
- uint8_t oldPasswd [4]
- uint8_t newPasswd [4]

- uint8_t newPasswdAgain [4]

### 8.478.1 Detailed Description

This structure contains Voice Set Call Barring Password Pack Parameters

**Parameters**

| | |
|---|---|
| *Reason* | • Call Barring Reason<br>• Values:<br>– 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing<br>– 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal<br>– 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME - Outgoing external to home<br>– 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming<br>– 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming<br>– 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred<br>– 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred<br>– 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred |
| *oldPasswd[BARRING_PASSWORD_LENGTH]* | • Old password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |
| *newPasswd[BARRING_PASSWORD_LENGTH]* | • New password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |
| *newPasswdAgain[BARRING_PASSWORD_LENGTH]* | • New password Again.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |

### 8.478.2 Field Documentation

#### 8.478.2.1 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]

#### 8.478.2.2 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]

#### 8.478.2.3 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]

#### 8.478.2.4 uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason

## 8.479 pack_voice_SLQSVoiceSetConfig_t Struct Reference

### Data Fields

- uint8_t * pAutoAnswer
- voice_airTimer * pAirTimerConfig
- voice_roamTimer * pRoamTimerConfig
- uint8_t * pTTYMode
- voice_prefVoiceSO * pPrefVoiceSO
- uint8_t * pPrefVoiceDomain

### 8.479.1 Detailed Description

This structure contains information about the Set Configuration Request Parameters.

**Parameters**

| | |
|---|---|
| *pAutoAnswer* | • Value specified is written to NV_AUTO_ANSWER_I. (optional)<br>• Values:<br>– 0x00 - Disable<br>– 0x01 - Enable |
| *pAirTimerConfig* | • Value specified is written to NV_AIR_CNT_I. (optional)<br>• See voice_airTimer for more information |
| *pRoamTimer-Config* | • Value specified is written to NV_ROAM_CNT_I. (optional)<br>• See voice_roamTimer for more information |
| *pTTYMode* | • Value specified is written to NV_TTY_I. (optional)<br>• Values:<br>– 0x00 - TTY_MODE_FULL - Full<br>– 0x01 - TTY_MODE_VCO - Voice carry over<br>– 0x02 - TTY_MODE_HCO - Hearing carry over<br>– 0x03 - TTY_MODE_OFF - Off |
| *pPrefVoiceSO* | • Value specified is written to NV_PREF_VOICE_SO_I. (optional)<br>• See voice_prefVoiceSO for more information |
| *pPrefVoice-Domain* | • Preferred Voice-Domain. (optional)<br>• Values:<br>– 0x00 - VOICE_DOMAIN_PREF_CS_ONLY - Circuit-switched (CS) only<br>– 0x01 - VOICE_DOMAIN_PREF_PS_ONLY - Packet-switched (PS) only<br>– 0x02 - VOICE_DOMAIN_PREF_CS_PREF - CS is preferred, PS is secondary<br>– 0x03 - VOICE_DOMAIN_PREF_PS_PREF - PS is preferred, CS is secondary |

**Note**

One of the optional parameters must be present in the request.

### 8.479.2 Field Documentation

#### 8.479.2.1 voice_airTimer* pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig

#### 8.479.2.2 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer

#### 8.479.2.3 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain

#### 8.479.2.4 voice_prefVoiceSO* pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO

#### 8.479.2.5 voice_roamTimer* pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig

#### 8.479.2.6 uint8_t* pack_voice_SLQSVoiceSetConfig_t::pTTYMode

## 8.480 pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference

### Data Fields

- uint8_t privacyPref

### 8.480.1 Detailed Description

This structure contains the preferred voice privacy values.

**Parameters**

| | |
|---|---|
| *privacyPref* | • Voice Privacy Preference<br>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy<br>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy |

### 8.480.2 Field Documentation

#### 8.480.2.1 uint8_t pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref

## 8.481 pack_voice_SLQSVoiceSetSUPSService_t Struct Reference

### Data Fields

- uint8_t voiceSvc
- uint8_t reason
- uint8_t * pServiceClass
- uint8_t * pCallBarringPasswd
- uint8_t * pCallForwardingNumber
- uint8_t * pTimerVal
- voice_callFwdTypeAndPlan * pCallFwdTypeAndPlan

### 8.481.1 Detailed Description

This structure contains Supplementary Service request parameters related to different features and their activation, deactivation, registration and erasure (applicable only for 3GPP)

**Parameters**

| | |
|---|---|
| *voiceSvc* | • Manages all call-independent supplementary services, such as activation, deactivation, registration, and erasure (mandatory)<br>– 0x01 - VOICE_SERVICE_ACTIVATE<br>– 0x02 - VOICE_SERVICE_DEACTIVATE<br>– 0x03 - VOICE_SERVICE_REGISTER<br>– 0x04 - VOICE_SERVICE_ERASE |
| *reason* | • supplementary service reason values (mandatory)<br>– 0x01 - QMI_VOICE_REASON_FWD_UNCONDITIONAL<br>Unconditional call forwarding<br>– 0x02 - QMI_VOICE_REASON_FWD_MOBILEBUSY<br>Forward when the mobile is busy<br>– 0x03 - QMI_VOICE_REASON_FWD_NOREPLY<br>Forward when there is no reply<br>– 0x04 - QMI_VOICE_REASON_FWD_UNREACHABLE<br>Forward when the call is unreachable<br>– 0x05 - QMI_VOICE_REASON_FWD_ALLFORWARDING<br>All forwarding<br>– 0x06 - QMI_VOICE_REASON_FWD_ALLCONDITIONAL<br>All conditional forwarding<br>– 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING<br>All outgoing calls are barred<br>– 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT<br>Outgoing internal calls are barred<br>– 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME<br>Outgoing calls external to home are barred<br>– 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING<br>All incoming calls are barred<br>– 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING<br>Roaming incoming calls are barred<br>– 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING<br>All calls are barred<br>– 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING<br>All outgoing calls are barred<br>– 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING<br>All incoming calls are barred<br>– 0x0F - QMI_VOICE_REASON_CALLWAITING<br>Call waiting |
| *pServiceClass* | • Service class is a combination (sum) of information class constants (optional)<br>– See liteServiceClassInformation for more information |
| *pCallBarring-Passwd* | • Password is required if call barring is provisioned using a password. Password consists of 4 ASCII digits. Range: 0000 to 9999 (optional) |

| | |
|---|---|
| *pCallForwarding-Number* | • Call forwarding number to be registered with the network. This has to be included in the request only when the service is set to VOICE_SERVICE_REGISTER. NULL terminated ASCII string. (optional) |
| *pTimerVal* | • Call forwarding no reply timer value in seconds. This has to be included in the request only when the service is set to VOICE_SERVICE_REGISTER and the reason is QMI_VOICE_REASON_FWD_NOREPLY. (optional)<br>– Range: 5 to 30 in steps of 5 |
| *pCallFwdType-AndPlan* | • Information about call forwarding type and plan. This parameter is ignored when the Call Forwarding Number is not included (optional)<br>– See voice_callFwdTypeAndPlan for more information |

### 8.481.2 Field Documentation

#### 8.481.2.1 uint8_t∗ pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd

#### 8.481.2.2 uint8_t∗ pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber

#### 8.481.2.3 voice_callFwdTypeAndPlan∗ pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan

#### 8.481.2.4 uint8_t∗ pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass

#### 8.481.2.5 uint8_t∗ pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal

#### 8.481.2.6 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason

#### 8.481.2.7 uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc

## 8.482 pack_voice_SLQSVoiceStartContDTMF_t Struct Reference

### Data Fields

- uint8_t ∗ pCallID
- uint8_t DTMFdigit

### 8.482.1 Detailed Description

This structure contains parameters of continuous DTMF

**Parameters**

| | |
|---|---|
| *pCallID* | • Call ID associated with call on which the DTMF information has to be sent.<br>Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.<br>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.<br>• If the call ID value received is 0, no value has been returned by the device |

| | |
|---|---|
| *DTMFdigit* | • DTMF digit in ASCII. |

### 8.482.2 Field Documentation

#### 8.482.2.1 uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit

#### 8.482.2.2 uint8_t∗ pack_voice_SLQSVoiceStartContDTMF_t::pCallID

## 8.483 pack_voice_SLQSVoiceStopContDTMF_t Struct Reference

### Data Fields

- uint8_t callID

### 8.483.1 Detailed Description

This structure contains parameters of stop continuous DTMF

**Parameters**

| | |
|---|---|
| *pCallID* | • Call ID associated with call on which the DTMF information has to be sent.<br>Stop continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.<br>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.<br>• If the call ID value received is 0, no value has been returned by the device |

### 8.483.2 Field Documentation

#### 8.483.2.1 uint8_t pack_voice_SLQSVoiceStopContDTMF_t::callID

## 8.484 pack_wds_DHCPv4ClientLeaseChange_t Struct Reference

### Data Fields

- uint8_t ∗ pEnableNotification

### 8.484.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

**Parameters**

| | |
|---|---|
| *pEnable-Notification* | • Enable Notification or not |

### 8.484.2 Field Documentation

#### 8.484.2.1 uint8_t* pack_wds_DHCPv4ClientLeaseChange_t::pEnableNotification

## 8.485 pack_wds_GetDefaultProfile_t Struct Reference

### Data Fields

- uint32_t profiletype

### 8.485.1 Detailed Description

This structure contains pack get default profile information.

**Parameters**

| | |
|---|---|
| *profiletype* | profile type<br>• 0 - WDS_PROFILE_TYPE_3GPP (0x00) - 3GPP<br>• 1 - WDS_PROFILE_TYPE_3GPP (0x01) - 3GPP2<br>• 2 - WDS_PROFILE_TYPE_EPC (0x02) - EPC |

### 8.485.2 Field Documentation

#### 8.485.2.1 uint32_t pack_wds_GetDefaultProfile_t::profiletype

## 8.486 pack_wds_GetDefaultProfileNum_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t family

### 8.486.1 Detailed Description

This structure contains pack Get Default Profile Number information.

**Parameters**

| | |
|---|---|
| *type* | profile type<br>• 0 - 3GPP<br>• 1 - 3GPP2 |
| *family* | profile family<br>• 0 - Embedded<br>• 1 - Tethered |

### 8.486.2 Field Documentation

#### 8.486.2.1 uint8_t pack_wds_GetDefaultProfileNum_t::family

#### 8.486.2.2 uint8_t pack_wds_GetDefaultProfileNum_t::type

## 8.487 pack_wds_GetDormancyState_t Struct Reference

### 8.487.1 Detailed Description

This structure contains pack get dormancy state information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.488 pack_wds_GetLastMobileIPError_t Struct Reference

### 8.488.1 Detailed Description

This structure contains pack get last mobile IP error information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.489 pack_wds_GetMobileIP_t Struct Reference

### 8.489.1 Detailed Description

This structure contains pack get mobile IP information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.490 pack_wds_GetMobileIPProfile_t Struct Reference

### Data Fields

- uint8_t index

### 8.490.1 Detailed Description

This structure contains pack get mobile IP profile information.

**Parameters**

| | |
|---|---|
| *index* | • Mobile IP profile ID |

### 8.490.2 Field Documentation

#### 8.490.2.1 uint8_t pack_wds_GetMobileIPProfile_t::index

## 8.491 pack_wds_GetPacketStatistics_t Struct Reference

### Data Fields

- uint32_t * pStatMask

### 8.491.1 Detailed Description

This structure contains pack get packet statistics information.

**Parameters**

| | |
|---|---|
| *pStatMask* | • Packet Statistics Mask<br>– 0x00000001 - Tx packets OK<br>– 0x00000002 - Rx packets OK<br>– 0x00000004 - Tx packet errors<br>– 0x00000008 - Rx packet errors<br>– 0x00000010 - Tx overflows<br>– 0x00000020 - Rx overflows<br>– 0x00000040 - Tx bytes OK<br>– 0x00000080 - Rx bytes OK |

### 8.491.2 Field Documentation

#### 8.491.2.1 uint32_t* pack_wds_GetPacketStatistics_t::pStatMask

## 8.492 pack_wds_GetPacketStatus_t Struct Reference

### Data Fields

- uint32_t statmask

### 8.492.1 Detailed Description

This structure contains pack get packet status information.

**Parameters**

| | |
|---|---|
| *statmask* | packet statistics mask<br>• 0x00000001 - Tx packets OK<br>• 0x00000002 - Rx packets OK<br>• 0x00000004 - Tx packet errors<br>• 0x00000008 - Rx packet errors<br>• 0x00000010 - Tx overflows<br>• 0x00000020 - Rx overflows<br>• 0x00000040 - Tx bytes OK<br>• 0x00000080 - Rx bytes OK<br>• 0x00000100 - Tx packets dropped<br>• 0x00000200 - Rx packets dropped |

### 8.492.2 Field Documentation

#### 8.492.2.1 uint32_t pack_wds_GetPacketStatus_t::statmask

## 8.493 pack_wds_GetSessionDuration_t Struct Reference

### 8.493.1 Detailed Description

This structure contains pack get session duration information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.494 pack_wds_RMSetTransferStatistics_t Struct Reference

### Data Fields

- rmTrasnferStaticsReq RmTrasnferStaticsReq

### 8.494.1 Detailed Description

This structure contains pack fetch current data system transfer statistics information.

**Parameters**

| | |
|---|---|
| *RmTrasnfer-StaticsReq[IN]* | • See rmTrasnferStaticsReq for more information |

### 8.494.2 Field Documentation

#### 8.494.2.1 rmTrasnferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasnferStaticsReq

## 8.495 pack_wds_SetAutoconnect_t Struct Reference

### Data Fields

- uint8_t acsetting
- uint8_t acroamsetting

### 8.495.1 Detailed Description

auto connect data session parameters.

**Parameters**

| | |
|---|---|
| *acsetting* | • Current autoconnect setting:<br>– 0x00 - Autoconnect disabled<br>– 0x01 - Autoconnect enabled<br>– 0x02 - Autoconnect paused (resume on powercycle) |
| *acroamsetting* | • Current autoconnect roaming status<br>– 0x00 - Autoconnect always allowed<br>– 0x01 - Autoconnect while in home service area only |

### 8.495.2 Field Documentation

#### 8.495.2.1 uint8_t pack_wds_SetAutoconnect_t::acroamsetting

#### 8.495.2.2 uint8_t pack_wds_SetAutoconnect_t::acsetting

## 8.496 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

### 8.496.1 Detailed Description

Writes the default profile settings to the device. The default profile is used to establish an autoconnect data session.

**Parameters**

| | |
|---|---|
| *profileType* | • Type of profile<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *pdpType* | • Packet Data Protocol (PDP) type specifies the type of data payload exchanged over the air link when the packet data session is established with this profile (optional)<br>– 0 - PDP-IP (IPv4)<br>– 1 - PDP-PPP<br>– 2 - PDP-IPv6<br>– 3 - PDP-IPv4v6 |
| *ipAddress* | • Preferred IPv4 addr to be assigned to device (optional) |
| *primaryDNS* | • Primary DNS Ipv4 address preference (optional) |
| *secondaryDNS* | • Secondary DNS Ipv4 address preference (optional) |
| *authentication* | • Bitmap that indicates authentication algorithm preference (optional)<br>– 0x00000001 - PAP preference<br>* 0 - Never performed<br>* 1 - May be performed<br>– 0x00000002 - CHAP preference<br>* 0 - Never performed<br>* 1 - May be performed<br>– All other bits are reserved and must be set to 0<br>– If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism. |
| *pName* | • profile Name (optional) |
| *pUsername* | • Username used during network authentication (optional) |
| *pAPNName* | • Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network (optional)<br>• If value is NULL or omitted, then subscription default value will be requested. |
| *pPassword* | • Password used during network authentication (optional) |

### 8.496.2 Field Documentation

#### 8.496.2.1 uint32_t pack_wds_SetDefaultProfile_t::authentication

#### 8.496.2.2 uint32_t pack_wds_SetDefaultProfile_t::ipAddress

#### 8.496.2.3 uint8_t* pack_wds_SetDefaultProfile_t::pApnname

#### 8.496.2.4 uint32_t pack_wds_SetDefaultProfile_t::pdpType

#### 8.496.2.5 uint8_t* pack_wds_SetDefaultProfile_t::pName

#### 8.496.2.6 uint8_t* pack_wds_SetDefaultProfile_t::pPassword

#### 8.496.2.7 uint32_t pack_wds_SetDefaultProfile_t::primaryDNS

#### 8.496.2.8 uint32_t pack_wds_SetDefaultProfile_t::profileType

#### 8.496.2.9 uint8_t* pack_wds_SetDefaultProfile_t::pUsername

#### 8.496.2.10 uint32_t pack_wds_SetDefaultProfile_t::secondaryDNS

## 8.497 pack_wds_SetDefaultProfileNum_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t family
- uint8_t index

### 8.497.1 Detailed Description

This structure to hold Set default profile number

**Parameters**

| | |
|---|---|
| *type* | Identifies the technology type of the profile<br>• 0 - 3GPP<br>• 1 - 3GPP2 |
| *family* | Identifies the family of profile<br>• 0 - Embedded<br>• 1 - Tethered |
| *index* | Profile number to be set as default profile. |

### 8.497.2 Field Documentation

#### 8.497.2.1 uint8_t pack_wds_SetDefaultProfileNum_t::family

#### 8.497.2.2 uint8_t pack_wds_SetDefaultProfileNum_t::index

#### 8.497.2.3 uint8_t pack_wds_SetDefaultProfileNum_t::type

## 8.498 pack_wds_SetMobileIP_t Struct Reference

### Data Fields

- uint32_t mode

### 8.498.1 Detailed Description

This structure contains set mobile IP pack information.

**Parameters**

| | |
|---|---|
| *mode* | • Mobile IP setting<br>– 0 - Mobile IP off (simple IP only)<br>– 1 - Mobile IP preferred<br>– 2 - Mobile IP only |

### 8.498.2 Field Documentation

#### 8.498.2.1 uint32_t pack_wds_SetMobileIP_t::mode

## 8.499 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

### 8.499.1 Detailed Description

Mobile IP parameters information.

**Parameters**

| | |
|---|---|
| *pSPC* | • NULL-terminated string representing six digit service programming code. |
| *pMode* | • Mode to be set (optional)<br>– 0 - Mobile IP off (simple IP only)<br>– 1 - Mobile IP preferred<br>– 2 - Mobile IP only |

| | |
|---|---|
| *pRetryLimit* | • Registration retry attempt limit (optional) |
| *pRetryInterval* | • Registration retry attempt interval used to determine the time between registration attempts (optional) |
| *pReRegPeriod* | • Period (in minutes) to attempt re-registration before current registration expires (optional) |
| *pReRegTraffic* | • Re-registration only if traffic since last attempt (optional)<br>– Zero - Disabled<br>– NonZero - Enabled |
| *pHA-Authenticator* | • MH-HA authenticator calculator (optional)<br>– Zero - Disabled<br>– NonZero - Enabled |
| *pHA2002bis* | • MH-HA RFC 2002bis authentication instead of RFC2002 (optional)<br>– Zero - Disabled<br>– NonZero - Enabled |

### 8.499.2 Field Documentation

#### 8.499.2.1 uint8_t∗ pack_wds_SetMobileIPParameters_t::pHA2002bis

#### 8.499.2.2 uint8_t∗ pack_wds_SetMobileIPParameters_t::pHAAuthenticator

#### 8.499.2.3 uint32_t∗ pack_wds_SetMobileIPParameters_t::pMode

#### 8.499.2.4 uint8_t∗ pack_wds_SetMobileIPParameters_t::pReRegPeriod

#### 8.499.2.5 uint8_t∗ pack_wds_SetMobileIPParameters_t::pReRegTraffic

#### 8.499.2.6 uint8_t∗ pack_wds_SetMobileIPParameters_t::pRetryInterval

#### 8.499.2.7 uint8_t∗ pack_wds_SetMobileIPParameters_t::pRetryLimit

#### 8.499.2.8 char∗ pack_wds_SetMobileIPParameters_t::pSPC

## 8.500 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

### 8.500.1 Detailed Description

Sets the mobile IP parameters.

**Parameters**

| | |
|---|---|
| *spc* | • Six digit service programming code string |
| *index* | • Index of the profile to modify |
| *pEnabled* | • (Optional) Enable profile? 0 - Disabled Nonzero - Enabled |
| *pAddress* | • (Optional) Home IPv4 address |
| *pPrimaryHA* | • (Optional) Primary home agent IPv4 address |
| *pSecondaryHA* | • (Optional) Secondary home agent IPv4 address |
| *pRevTunneling* | • (Optional) Enable reverse tunneling? 0 - Disabled Nonzero - Enabled |
| *pNAI* | • (Optional) Network access identifier string |
| *pHASPI* | • (Optional) Home agent security parameter index |
| *pAAASPI* | • (Optional) AAA server security parameter index |
| *pMNHA* | • (Optional) MN-HA key string |
| *pMNAAA* | • (Optional) MN-AAA key string |

### 8.500.2 Field Documentation

#### 8.500.2.1 uint8_t pack_wds_SetMobileIPProfile_t::index

#### 8.500.2.2 uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI

#### 8.500.2.3 uint32_t* pack_wds_SetMobileIPProfile_t::pAddress

#### 8.500.2.4 uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled

#### 8.500.2.5 uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI

#### 8.500.2.6 int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA

#### 8.500.2.7 int8_t* pack_wds_SetMobileIPProfile_t::pMNHA

#### 8.500.2.8 int8_t* pack_wds_SetMobileIPProfile_t::pNAI

#### 8.500.2.9 uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA

#### 8.500.2.10 uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling

#### 8.500.2.11 uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA

#### 8.500.2.12 int8_t pack_wds_SetMobileIPProfile_t::spc[10]

## 8.501 pack_wds_SLQSCreateProfile_t Struct Reference

### Data Fields

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

### 8.501.1 Detailed Description

This structure contains pack create profile.

**Parameters**

| | |
|---|---|
| *pProfileId* | • 1 to 16 for 3GPP profile (EM/MC73xx or earlier)<br>• 1 to 24 for 3GPP profile (EM/MC74xx onwards)<br>• 101 to 106 for 3GPP2 profile |
| *pProfileType* | • Identifies the technology type of the profile<br>– 0x00 - 3GPP<br>– 0x01 - 3GPP2<br>– NULL is not allowed |
| *pCurProfile* | • union of 3GPP and 3GPP2 profile<br>• See wds_profileInfo |

**Note**

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

### 8.501.2 Field Documentation

#### 8.501.2.1 wds_profileInfo* pack_wds_SLQSCreateProfile_t::pCurProfile

#### 8.501.2.2 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileId

#### 8.501.2.3 uint8_t* pack_wds_SLQSCreateProfile_t::pProfileType

## 8.502 pack_wds_SLQSDeleteProfile_t Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileIndex

### 8.502.1 Detailed Description

This structure contains the information about the profile to be deleted.

**Parameters**

| | |
|---|---|
| *profileType* | • Identifies the type of profile<br>– 0x00 - 3GPP<br>• Note: Deletion of 3GPP2 profiles is not supported. |
| *profileIndex* | • Index of the configured profile to be deleted<br>– Value between 1 - 16 (EM/MC73xx or earlier)<br>– Value between 1 - 24 (EM/MC74xx onwards) |

### 8.502.2 Field Documentation

#### 8.502.2.1 uint8_t pack_wds_SLQSDeleteProfile_t::profileIndex

#### 8.502.2.2 uint8_t pack_wds_SLQSDeleteProfile_t::profileType

## 8.503 pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

### 8.503.1 Detailed Description

This structure contains pack get current data system state information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.504 pack_wds_SLQSGetDataBearerTechnology_t Struct Reference

### 8.504.1 Detailed Description

This structure contains pack get data bearer technology information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.505 pack_wds_SLQSGetDUNCallInfo_t Struct Reference

### Data Fields

- uint32_t Mask
- uint8_t * pReportConnStatus
- wds_transferStatInd * pTransferStatInd
- uint8_t * pReportDormStatus
- uint8_t * pReportDataBearerTech
- uint8_t * pReportChannelRate

### 8.505.1 Detailed Description

This structure contains the DUN Call Info Request parameters.

**Parameters**

| | |
|---|---|
| *Mask* | • Mandatory parameter<br>• Set the bits corresponding to the information requested to 1<br>• All other bits must be set to 0.<br>• If any values are not available or applicable, the corresponding TLVs are not returned in the response.<br>– Bit 0 - Connection Status<br>– Bit 1 - Last call end reason<br>– Bit 2 - Tx/Rx bytes OK<br>– Bit 3 - Dormancy status<br>– Bit 4 - Data bearer<br>– Bit 5 - Channel rate<br>– Bit 6 - Call active duration |
| *pReportConnStatus* | • Connect Status Indicator<br>– 0 - Do not report<br>– 1 - Report connection status and call end reason |
| *pTransferStatInd* | • See wds_transferStatInd for more information |

| | |
|---|---|
| *pReportDorm-Status* | • Dormancy Status Indicator<br>– 0 - Do not report<br>– 1 - Report traffic channel state of interface used for data connection |
| *pReportData-BearerTech* | • Current Data Bearer Technology Indicator<br>– 0 - Do not report<br>– 1 - Report radio interface used for data transfer when it changes |
| *pReport-ChannelRate* | • Channel Rate Indicator<br>– 0 - Do not report<br>– 1 - Report channel rate |

### 8.505.2 Field Documentation

#### 8.505.2.1 uint32_t pack_wds_SLQSGetDUNCallInfo_t::Mask

#### 8.505.2.2 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportChannelRate

#### 8.505.2.3 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportConnStatus

#### 8.505.2.4 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDataBearerTech

#### 8.505.2.5 uint8_t* pack_wds_SLQSGetDUNCallInfo_t::pReportDormStatus

#### 8.505.2.6 wds_transferStatInd* pack_wds_SLQSGetDUNCallInfo_t::pTransferStatInd

## 8.506 pack_wds_SLQSGetProfileSettings_t Struct Reference

### Data Fields

- uint8_t ProfileId
- uint8_t ProfileType

### 8.506.1 Detailed Description

This structure contains pack get profile settings information.

**Parameters**

| | |
|---|---|
| *ProfileID* | • 1 to 16 for 3GPP profile (EM/MC73xx or earlier)<br>• 1 to 24 for 3GPP profile (EM/MC74xx onwards)<br>• 101 to 106 for 3GPP2 profile |
| *ProfileType* | • Identifies the technology type of the profile<br>– 0x00 - 3GPP<br>– 0x01 - 3GPP2 |

**Note**

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

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 8.506.2 Field Documentation

#### 8.506.2.1 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileId

#### 8.506.2.2 uint8_t pack_wds_SLQSGetProfileSettings_t::ProfileType

## 8.507 pack_wds_SLQSGetRuntimeSettings_t Struct Reference

### Data Fields

- uint32_t * pReqSettings

### 8.507.1 Detailed Description

This structure contains pack get runtime settings information

**Parameters**

| | |
|---|---|
| *pReqSettings* | Requested Settings (Optional Parameter)<br>• Set bits to 1, corresponding to requested information. All other bits must be set to 0.<br>• If the values are not available, the corresponding TLVs are not returned in the response.<br>• Absence of this mask TLV results in the device returning all of the available information corresponding to bits 0 through 12.<br>• 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.<br>• Values<br>– Bit 0 - Profile identifier<br>– Bit 1 - Profile name<br>– Bit 2 - PDP type<br>– Bit 3 - APN name<br>– Bit 4 - DNS address<br>– Bit 5 - UMTS/GPRS granted QoS<br>– Bit 6 - Username<br>– Bit 7 - Authentication Protocol<br>– Bit 8 - IP address<br>– Bit 9 - Gateway info (address and subnet mask)<br>– Bit 10 - PCSCF address using PCO flag<br>– Bit 11 - PCSCF server address list<br>– Bit 12 - PCSCF domain name list<br>– Bit 13 - MTU<br>– Bit 14 - domain name list<br>– Bit 15 - IP family<br>– Bit 16 - IM_CM flag<br>– Bit 17 - Technology name<br>– Bit 18 - Operator reserved PCO (Not Supported on MC/EM73xx) |

### 8.507.2 Field Documentation

#### 8.507.2.1 uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings

## 8.508 pack_wds_SLQSModifyProfile_t Struct Reference

### Data Fields

- uint8_t * pProfileId
- uint8_t * pProfileType
- wds_profileInfo curProfile

### 8.508.1 Detailed Description

This structure contains pack modify profile information.

**Parameters**

| | |
|---|---|
| *pProfileId* | • 1 to 16 for 3GPP profile (EM/MC73xx or earlier)<br>• 1 to 24 for 3GPP profile (EM/MC74xx onwards)<br>• 101 to 106 for 3GPP2 profile |
| *pProfileType* | • Identifies the technology type of the profile<br>– 0x00 - 3GPP<br>– 0x01 - 3GPP2<br>– NULL is not allowed |
| *curProfile* | • union of 3GPP and 3GPP2 profile<br>• See wds_profileInfo |

**Note**

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

### 8.508.2 Field Documentation

#### 8.508.2.1 wds_profileInfo pack_wds_SLQSModifyProfile_t::curProfile

#### 8.508.2.2 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileId

#### 8.508.2.3 uint8_t* pack_wds_SLQSModifyProfile_t::pProfileType

## 8.509 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

### 8.509.1 Detailed Description

This structure contains pack set 3GPP configuration item information.

**Parameters**

| | |
|---|---|
| *pLTEAttach-Profile* | • Optional parameter<br>• LTE Attach Profile<br>– points to a single WORD Value indicating the attached LTE Profile<br>– Optional parameter with possible values 1-16 (EM/MC73xx or earlier)<br>• This setting is deprecated on MC/EM74xx |

| | |
|---|---|
| *ProfileList* | Profile List<br>• an array of 4 profile configurations<br>• Each element points to a single WORD value indicating profile<br>• Optional parameter with possible values<br>– 1 - 16 (MC/EM73xx and before)<br>– 1 - 24 (MC/EM74xx and onwards)<br>• function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present |
| *pDefaultPDN-Enabled* | • Optional parameter<br>– 0 - disabled<br>– 1 - enabled |
| *p3gppRelease* | 3GPP release<br>• Optional parameter<br>– 0 - Release_99<br>– 1 - Release_5<br>– 2 - Release_6<br>– 3 - Release_7<br>– 4 - Release_8<br>• In 9x30 and onwards<br>– 5 - Release 9<br>– 6 - Release 10<br>– 7 - Release 11 |
| *pLTEAttach-ProfileList* | • pointer to WORD array indicating LTE Attach Profile List<br>– Optional parameter<br>– possible values: 1-24<br>– This setting is only supported for MC/EM74xx onwards<br>– Please provide attach profiles in order of decreasing priority in this list. |
| *LTEAttach-ProfileListLen* | • Number of element in pLTEAttachProfileList<br>– valid range: 1-24<br>– This setting is only supported for MC/EM74xx onwards |

### 8.509.2 Field Documentation

#### 8.509.2.1 uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen

#### 8.509.2.2 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::p3gppRelease

#### 8.509.2.3 uint8_t* pack_wds_SLQSSet3GPPConfigItem_t::pDefaultPDNEnabled

#### 8.509.2.4 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfile

#### 8.509.2.5 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pLTEAttachProfileList

#### 8.509.2.6 uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList

## 8.510 pack_wds_SLQSSetIPFamilyPreference_t Struct Reference

### Data Fields

- uint8_t IPFamilyPreference

### 8.510.1 Detailed Description

This structure contains pack set IP family preference information

**Parameters**

| | |
|---|---|
| *IPFamily-Preference* | IP Family preference<br>• PACK_WDS_IPV4 IP Version 4<br>• PACK_WDS_IPV6 IP Version 6 |

### 8.510.2 Field Documentation

#### 8.510.2.1 uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference

## 8.511 pack_wds_SLQSSetWdsEventCallback_t Struct Reference

### Data Fields

- uint8_t dataBearer
- uint8_t dormancyStatus
- uint8_t mobileIP
- uint8_t transferStats
- uint8_t currentDataBearer
- uint8_t dataSystemStatus
- uint8_t dataBearerTechExt
- uint8_t interval

### 8.511.1 Detailed Description

This structure contains set WDS event callback information.

**Parameters**

| | |
|---|---|
| *dataBearer* | data bearer |
| *dormancyStatus* | dormancy status |
| *mobileIP* | mobile IP |
| *currentData-Bearer* | current data bearer |
| *dataSystem-Status* | data system status |
| *dataBearerTech-Ext* | data Bearer Technology Extended |
| *interval* | interval |

### 8.511.2 Field Documentation

#### 8.511.2.1 uint8_t pack_wds_SLQSSetWdsEventCallback_t::currentDataBearer

#### 8.511.2.2 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearer

#### 8.511.2.3 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataBearerTechExt

#### 8.511.2.4 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dataSystemStatus

#### 8.511.2.5 uint8_t pack_wds_SLQSSetWdsEventCallback_t::dormancyStatus

#### 8.511.2.6 uint8_t pack_wds_SLQSSetWdsEventCallback_t::interval

#### 8.511.2.7 uint8_t pack_wds_SLQSSetWdsEventCallback_t::mobileIP

#### 8.511.2.8 uint8_t pack_wds_SLQSSetWdsEventCallback_t::transferStats

## 8.512 pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

### Data Fields

- wdsDhcpv4ProfileId * pProfileId

### 8.512.1 Detailed Description

This structure contain get DHCPv4 client configure.

**Parameters**

| | |
|---|---|
| *pProfileId* | pointer to Profile Id structure<br>• See wdsDhcpv4ProfileId |

### 8.512.2 Field Documentation

#### 8.512.2.1 wdsDhcpv4ProfileId* pack_wds_SLQSSGetDHCPv4ClientConfig_t::pProfileId

## 8.513 pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference

### Data Fields

- wds_DHCPv4ProfileId * pProfileId
- wds_DHCPv4HWConfig * pHwConfig
- wds_DHCPv4OptionList * pRequestOptionList

### 8.513.1 Detailed Description

WDS SWI DHCPv4 Config Structure

**Parameters**

| | |
|---|---|
| *pProfileId* | • pointer to Profile Id structure |
| *pHWConfig* | • pointer to HW Config structure |
| *pRequestOption-List* | • pointer to Option List structure to be sent in DHCP request |

### 8.513.2 Field Documentation

#### 8.513.2.1 wds_DHCPv4HWConfig∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pHwConfig

#### 8.513.2.2 wds_DHCPv4ProfileId∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pProfileId

#### 8.513.2.3 wds_DHCPv4OptionList∗ pack_wds_SLQSSSetDHCPv4ClientConfig_t::pRequestOptionList

## 8.514 pack_wds_SLQSSSetLoopback_t Struct Reference

### Data Fields

- uint8_t loopbackMode
- uint8_t loopbackMultiplier

### 8.514.1 Detailed Description

This structure contains pack set loopback information.

**Parameters**

| | |
|---|---|
| *loopbackMode* | • Loopback Mode.<br>– 0 - Disable<br>– 1 - Enable |
| *loopback-Multiplier* | • Loopback multiplier. Number of downlink bytes to send for each uplink byte. |

### 8.514.2 Field Documentation

#### 8.514.2.1 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMode

#### 8.514.2.2 uint8_t pack_wds_SLQSSSetLoopback_t::loopbackMultiplier

## 8.515 pack_wds_SLQSStartDataSession_t Struct Reference

### Data Fields

- uint8_t ∗ pTech

- uint32_t * pprofileid3gpp
- uint32_t * pprofileid3gpp2
- uint32_t * pAuth
- char * pUser
- char * pPass

### 8.515.1 Detailed Description

This structure contains pack Start Data Session Information.

**Parameters**

| | |
|---|---|
| *pTech* | • Indicates the technology preference<br>– 1 - UMTS<br>– 2 - CDMA<br>– 3 - eMBMS<br>– 4 - Modem Link Label. Modem Link is an interface for transferring data between entities on AP and modem.<br>• optional |
| *pprofileid3gpp* | • pointer to 3GPP profile id<br>• optional |
| *pprofileid3gpp2* | • pointer to 3GPP2 profile id<br>• optional |
| *pAuth* | • Authentication type, it can be PAP or CHAP<br>• optional |
| *pUser* | • username for authentication process<br>• optional |
| *pPass* | • password for authentication process<br>• optional |

### 8.515.2 Field Documentation

#### 8.515.2.1 uint32_t* pack_wds_SLQSStartDataSession_t::pAuth

#### 8.515.2.2 char* pack_wds_SLQSStartDataSession_t::pPass

#### 8.515.2.3 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp

#### 8.515.2.4 uint32_t* pack_wds_SLQSStartDataSession_t::pprofileid3gpp2

#### 8.515.2.5 uint8_t* pack_wds_SLQSStartDataSession_t::pTech

#### 8.515.2.6 char* pack_wds_SLQSStartDataSession_t::pUser

## 8.516 pack_wds_SLQSStopDataSession_t Struct Reference

### Data Fields

- uint32_t * psid

### 8.516.1 Detailed Description

This structure contains pack stop data session information.

**Parameters**

| | |
|---|---|
| *psid* | session id |

### 8.516.2 Field Documentation

#### 8.516.2.1 uint32_t* pack_wds_SLQSStopDataSession_t::psid

## 8.517 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

### 8.517.1 Detailed Description

This structure contains the information about the Set Event Report Request parameters.

**Parameters**

| | |
|---|---|
| *pCurrChannel-RateInd* | (optional)<br>• Current Channel Rate Indicator.<br>– 0 - Do not report<br>– 1 - Report channel rate when it changes |
| *pTransferStatInd* | (optional)<br>• See wds_TrStatInd for more information. |
| *pDataBearer-TechInd* | (optional)<br>• Data Bearer Technology Indicator.<br>– 0 - Do not report<br>– 1 - Report radio interface used for data transfer when it changes |

| | |
|---|---|
| *pDormancy-StatusInd* | (optional)<br>• Dormancy Status indicator.<br>– 0 - Do not report<br>– 1 - Report traffic channel state of interface used for data connection |
| *pMIPStatusInd* | (optional)<br>• MIP Status Indicator.<br>– 0 - Do not report<br>– 1 - Report MIP status |
| *pCurrData-BearerTechInd* | (optional)<br>• Current Data Bearer Technology Indicator.<br>– 0 - Do not report<br>– 1 - Report current data bearer technology when it changes |
| *pDataCallStatus-ChangeInd* | (optional)<br>• Data Call Status Change Indicator.<br>– 0 - Do not report<br>– 1 - Report data call status change when it changes |
| *pCurrPrefData-SysInd* | (optional)<br>• Current Preferred Data System Indicator.<br>– 0 - Do not report<br>– 1 - Report preferred data system when it changes |
| *pEVDOPage-MonPerChange-Ind* | (optional)<br>• EV-DO Page Monitor Period Change Indicator.<br>– 0 - Do not report<br>– 1 - Report EV-DO page monitor period change event |
| *pDataSystem-StatusChange-Ind* | (optional)<br>• Data System Status Change Indicator.<br>– 0 - Do not report<br>– 1 - Report data system status change event |

**Note**

At least one parameter should be present.

### 8.517.2 Field Documentation

#### 8.517.2.1 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd

#### 8.517.2.2 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd

#### 8.517.2.3 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd

#### 8.517.2.4 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd

#### 8.517.2.5 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd

#### 8.517.2.6 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd

#### 8.517.2.7 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd

#### 8.517.2.8 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd

#### 8.517.2.9 uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd

#### 8.517.2.10 wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd

## 8.518 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

### Data Fields

- uint8_t contextId
- uint8_t contextType

### 8.518.1 Detailed Description

This structure contains the PDP Runtime Settings Request parameters.

**Parameters**

| | |
|---|---|
| *contextId* | • Context Identifier |
| *contextType* | • Identifies technology type<br>– 0 - 3GPP<br>– 1 - 3GPP2 |

### 8.518.2 Field Documentation

#### 8.518.2.1 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

#### 8.518.2.2 uint8_t pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType

## 8.519 PackCreateProfileOut Struct Reference

### Data Fields

- uint8_t ProfileType
- uint8_t ProfileIndex
- uint16_t ExtErrorCode

### 8.519.1 Detailed Description

This structure contains out parameter Information

**Parameters**

| | |
|---|---|
| *ProfileType* | • Identifies the type of profile 0x00 = 3GPP 0x01 = 3GPP2 |
| *ProfileIndex* | • Index identifying the profile that was created |
| *ExtErrorCode* | • The extended error code received from DS Profile subsystem |

### 8.519.2 Field Documentation

#### 8.519.2.1 uint16_t PackCreateProfileOut::ExtErrorCode

#### 8.519.2.2 uint8_t PackCreateProfileOut::ProfileIndex

#### 8.519.2.3 uint8_t PackCreateProfileOut::ProfileType

## 8.520 packgetDyingGaspCfg Struct Reference

### Data Fields

- uint8_t ∗ pDestSMSNum
- uint8_t ∗ pDestSMSContent

### 8.520.1 Detailed Description

This struture contains the TLV required to get the Dysing GASP Config.

**Parameters**

| | |
|---|---|
| *pDestSMSNum[IN]* | • SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.<br>• Optional parameter. |
| *pDestSMSContent[IN]* | • SMS COntent as a string of 8 bit ASCII text characters Max 160 chars.<br>• Optional parameter. |

### 8.520.2 Field Documentation

#### 8.520.2.1 uint8_t∗ packgetDyingGaspCfg::pDestSMSContent

#### 8.520.2.2 uint8_t∗ packgetDyingGaspCfg::pDestSMSNum

## 8.521 packgetDyingGaspStatistics Struct Reference

### Data Fields

- uint32_t ∗ pTimeStamp

- uint8_t * pSMSAttemptedFlag

### 8.521.1 Detailed Description

This structure contains Get Dysing GASP Statistics

**Parameters**

| | |
|---|---|
| *pTimeStamp[OUT]* | • Timestamp of the last time power loss was detected and Dying Gasp feature was triggered.<br>• UTC time in seconds since Jan 06, 1980 (GPS Epoch). |
| *pSMSAttemptedFlag[OUT]* | • Indicates whether device attempted to send SMS in the last power loss event.<br>• 0 - SMS not attempted<br>• 1 - SMS attempted<br>• This only indicates device sent the SMS but does not guarantee network delivery. |

### 8.521.2 Field Documentation

#### 8.521.2.1 uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag

#### 8.521.2.2 uint32_t* packgetDyingGaspStatistics::pTimeStamp

## 8.522 PackSwiAvmsSetSettingsAPNInfo Struct Reference

### Data Fields

- uint8_t bAPNLength
- uint8_t szAPN [49]
- uint8_t bUnameLength
- uint8_t szUname [29]
- uint8_t bPWDLength
- uint8_t szPWD [29]

### 8.522.1 Detailed Description

This structure contains pack set settings APN information.

**Parameters**

| | |
|---|---|
| *bAPNLength* | APN Length. |
| *szAPN* | APN String. |
| *bUnameLength* | Username Length. |
| *szUname* | User Name String. |
| *bPWDLength* | Password Length. |
| *szPWD* | Password String. |

### 8.522.2 Field Documentation

#### 8.522.2.1 uint8_t PackSwiAvmsSetSettingsAPNInfo::bAPNLength

#### 8.522.2.2 uint8_t PackSwiAvmsSetSettingsAPNInfo::bPWDLength

#### 8.522.2.3 uint8_t PackSwiAvmsSetSettingsAPNInfo::bUnameLength

#### 8.522.2.4 uint8_t PackSwiAvmsSetSettingsAPNInfo::szAPN[49]

#### 8.522.2.5 uint8_t PackSwiAvmsSetSettingsAPNInfo::szPWD[29]

#### 8.522.2.6 uint8_t PackSwiAvmsSetSettingsAPNInfo::szUname[29]

## 8.523 PackSwiAvmsSetSettingsConnectionRetryTimers Struct Reference

### Data Fields

- uint16_t Timers [8]

### 8.523.1 Detailed Description

This structure contains the SLQSAVMSSetSettings Polling timer to connect to AVMS server.

**Parameters**

| | |
|---|---|
| *Timers[0]* | - Timer 1.<br>• 0-20160 (min)<br>• 0:retry disabled |
| *Timers[1]* | - Timer 2.<br>• 1-20160 (min) |
| *Timers[2]* | - Timer 3.<br>• 1-20160 (min) |
| *Timers[3]* | - Timer 4.<br>• 1-20160 (min) |
| *Timers[4]* | - Timer 5.<br>• 1-20160 (min) |
| *Timers[5]* | - Timer 6.<br>• 1-20160 (min) |
| *Timers[6]* | - Timer 7.<br>• 1-20160 (min) |
| *Timers[7]* | - Timer 8.<br>• 1-20160 (min) |

### 8.523.2 Field Documentation

#### 8.523.2.1 uint16_t PackSwiAvmsSetSettingsConnectionRetryTimers::Timers[8]

## 8.524 PackSwiAvmsSetSettingsPeriodInfo Struct Reference

### Data Fields

- uint32_t ulMin
- uint32_t ulMax

### 8.524.1 Detailed Description

This structure contain pack set settings period information.

**Parameters**

| | |
|---|---|
| *ulMin* | Min period in seconds. |
| *ulMax* | Max period in seconds. |

### 8.524.2 Field Documentation

#### 8.524.2.1 uint32_t PackSwiAvmsSetSettingsPeriodInfo::ulMax

#### 8.524.2.2 uint32_t PackSwiAvmsSetSettingsPeriodInfo::ulMin

## 8.525 PackSwiAVMSSettingsAPNInfo Struct Reference

### Data Fields

- uint8_t bAPNLength
- uint8_t * pAPN
- uint8_t bUnameLength
- uint8_t * pUname
- uint8_t bPWDLength
- uint8_t * pPWD

### 8.525.1 Detailed Description

This structure contains the SLQSAVMSGetSettings APN Info to connect to AVMS server.

**Parameters**

| | |
|---|---|
| *bAPNLength* | - APN Length(Max 49). |
| *szAPN* | - APN. |
| *bUnameLength* | - User Name Length (Max 29). |
| *szUname* | - User Name. |
| *bPWDLength* | - Password Length (Max 29). |
| *szPWD* | - Password. |

### 8.525.2 Field Documentation

#### 8.525.2.1 uint8_t PackSwiAVMSSettingsAPNInfo::bAPNLength

#### 8.525.2.2 uint8_t PackSwiAVMSSettingsAPNInfo::bPWDLength

#### 8.525.2.3 uint8_t PackSwiAVMSSettingsAPNInfo::bUnameLength

#### 8.525.2.4 uint8_t* PackSwiAVMSSettingsAPNInfo::pAPN

#### 8.525.2.5 uint8_t* PackSwiAVMSSettingsAPNInfo::pPWD

#### 8.525.2.6 uint8_t* PackSwiAVMSSettingsAPNInfo::pUname

## 8.526 PackSwiAVMSSettingsConnectionRetryTimers Struct Reference

### Data Fields

- uint16_t Timers [8]

### 8.526.1 Detailed Description

This structure contains the SLQSAVMSGetSettings Polling timer to connect to AVMS server.

**Parameters**

| | |
|---|---|
| *Timers[0]* | - Timer 1.<br>• 0-20160 (min)<br>• 0:retry disabled |
| *Timers[1]* | - Timer 2.<br>• 1-20160 (min) |
| *Timers[2]* | - Timer 3.<br>• 1-20160 (min) |
| *Timers[3]* | - Timer 4.<br>• 1-20160 (min) |
| *Timers[4]* | - Timer 5.<br>• 1-20160 (min) |
| *Timers[5]* | - Timer 6.<br>• 1-20160 (min) |
| *Timers[6]* | - Timer 7.<br>• 1-20160 (min) |
| *Timers[7]* | - Timer 8.<br>• 1-20160 (min) |

### 8.526.2 Field Documentation

#### 8.526.2.1 uint16_t PackSwiAVMSSettingsConnectionRetryTimers::Timers[8]

## 8.527 PackSwiAVMSSettingsPeriodsInfo Struct Reference

### Data Fields

- uint32_t min
- uint32_t max

### 8.527.1 Detailed Description

This structure contains the SLQSAVMSSettings Min/Max Period of an Observation.

**Parameters**

| | |
|---|---|
| *min* | - Min period in seconds. |
| *max* | - Max period in seconds. |

### 8.527.2 Field Documentation

#### 8.527.2.1 uint32_t PackSwiAVMSSettingsPeriodsInfo::max

#### 8.527.2.2 uint32_t PackSwiAVMSSettingsPeriodsInfo::min

## 8.528 qmiSmsMessageList Struct Reference

### Data Fields

- uint32_t messageIndex
- uint32_t messageTag

### 8.528.1 Detailed Description

This structure contains SMS message list.

**Parameters**

| | |
|---|---|
| *messageIndex* | • Message index of each matched message |
| *messageTag* | • Messagetag |

### 8.528.2 Field Documentation

#### 8.528.2.1 uint32_t qmiSmsMessageList::messageIndex

#### 8.528.2.2 uint32_t qmiSmsMessageList::messageTag

## 8.529 qmiWDSDataBearerTechnology Struct Reference

### Data Fields

- uint8_t currentNetwork
- uint32_t ratMask
- uint32_t soMask

### 8.529.1 Detailed Description

Structure to hold the current data bearer technology values

**Parameters**

| | |
|---|---|
| *currentNetwork[-OUT]* | • current selected network<br>– 0 - UNKNOWN<br>– 1 - 3GPP2<br>– 2 - 3GPP<br>– 0xff - Invalid data. |
| *ratMask[OUT]* | RAT Mask |
| *soMask[OUT]* | SO Mask |

### 8.529.2 Field Documentation

#### 8.529.2.1 uint8_t qmiWDSDataBearerTechnology::currentNetwork

#### 8.529.2.2 uint32_t qmiWDSDataBearerTechnology::ratMask

#### 8.529.2.3 uint32_t qmiWDSDataBearerTechnology::soMask

## 8.530 qos_BindDataPortMuxID_t Struct Reference

### Data Fields

- uint8_t MuxID

### 8.530.1 Detailed Description

Structure that contains the request Binds a control point to a data port Mux ID data.

**Parameters**

| | |
|---|---|
| *MuxID* | Mux ID:<br>• 0x80-0x8F : valid value |

### 8.530.2 Field Documentation

#### 8.530.2.1 uint8_t qos_BindDataPortMuxID_t::MuxID

## 8.531 qos_BindDataPortPeripheralEndPointID_t Struct Reference

### Data Fields

- uint32_t EndPointType
- uint32_t IfaceID

### 8.531.1 Detailed Description

Structure that contains the request Binds a control point to a data port Periapheral End Point Type data.

**Parameters**

| | |
|---|---|
| *EndPointType* | Peripheral end point type. Values:<br>• 0 - Reserved<br>• 1 - HSIC<br>• 2 - HSUSB<br>• 3 - PCIE<br>• 4 - Embedded |
| *IfaceID* | Peripheral interface number. |

### 8.531.2 Field Documentation

#### 8.531.2.1 uint32_t qos_BindDataPortPeripheralEndPointID_t::EndPointType

#### 8.531.2.2 uint32_t qos_BindDataPortPeripheralEndPointID_t::IfaceID

## 8.532 qos_BindDataPortSIODataPort_t Struct Reference

### Data Fields

- uint16_t SIODataPort

### 8.532.1 Detailed Description

Structure that contains the request Binds a control point to a data port SIO Data Port data.

**Parameters**

| | |
|---|---|
| *u16SIODataPort* | SIO Data Port to which the client binds |

### 8.532.2 Field Documentation

#### 8.532.2.1 uint16_t qos_BindDataPortSIODataPort_t::SIODataPort

## 8.533 RFBandInfoElements Struct Reference

### Data Fields

- uint8_t radioInterface
- uint16_t activeBandClass
- uint16_t activeChannel

### 8.533.1 Detailed Description

This structure contains the RFBandInfo response parameters.

**Parameters**

| | |
|---|---|
| *radioInterface* | • Radio interface technology<br>– See Tables for Radio Interface |
| *activeBandClass* | • Active Band Class<br>– See Tables for Band Classes |
| *activeChannel* | • Active channel (0 if channel is not relevant to the reported technology) |

### 8.533.2 Field Documentation

#### 8.533.2.1 uint16_t RFBandInfoElements::activeBandClass

#### 8.533.2.2 uint16_t RFBandInfoElements::activeChannel

#### 8.533.2.3 uint8_t RFBandInfoElements::radioInterface

## 8.534 rmTrasnferStaticsReq Struct Reference

### Data Fields

- uint8_t bResetStatistics
- uint32_t ulMask

### 8.534.1 Detailed Description

RM Transfer Satistics Structure

**Parameters**

| | |
|---|---|
| *bResetStatistics* | • Reset Statistics<br>• Values:<br>• 0 - Not Reset<br>• Other - Reset |
| *ulMask* | • Enable/Disable RM Transfer Satatistics Indiscation Mask<br>• Bit 0: Tx Packet Ok<br>• Bit 1: Rx Packet Ok<br>• Bit 2: Tx Bytes Ok<br>• Bit 3: Rx Bytes Ok<br>• Bit 4: Tx Packets Dropped<br>• Bit 5: Rx Packets Dropped<br>• Value: -0 - Disable -1 - Enable |

### 8.534.2 Field Documentation

#### 8.534.2.1 uint8_t rmTrasnferStaticsReq::bResetStatistics

#### 8.534.2.2 uint32_t rmTrasnferStaticsReq::ulMask

## 8.535 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]

### 8.535.1 Detailed Description

This structure specifies information regarding the 3-Axis Sensor Data. Please check has_<Param_Name> field for presence of optional parameters

**Parameters**

| | |
|---|---|
| *timeOfFirst-Sample* | • Denotes a full 32-bit time stamp of the first (oldest) sample in this message.<br>• The time stamp is in the time reference scale that is used by the sensor time source.<br>• Units - Milliseconds |
| *flags* | • Flags to indicate any deviation from the default measurement assumptions.<br>• All unused bits in this field must be set to 0.<br>• Valid bitmasks<br>– 0x01 - Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples<br>– 0x02 - Bitmask to specify that the sensor time stamp is the same as the modem time stamp |
| *sensorDataLen* | • Number of sets of the following elements<br>– timeOffset<br>– xAxis<br>– yAxis<br>– zAxis |
| *timeOffset* | • Sample time offset<br>• Units - Milliseconds |
| *xAxis* | • Sensor x-axis sample.<br>• Units Accelerometer - Meters/seconds square<br>• Units Gyroscope - Radians/second |

| | |
|---|---|
| *yAxis* | • Sensor Y-axis sample.<br>• Units Accelerometer - Meters/seconds square<br>• Units Gyroscope - Radians/second |
| *xAxis* | • Sensor Z-axis sample.<br>• Units Accelerometer - Meters/seconds square<br>• Units Gyroscope - Radians/second |

### 8.535.2 Field Documentation

#### 8.535.2.1 uint8_t sensorData_t::flags

#### 8.535.2.2 uint8_t sensorData_t::sensorDataLen

#### 8.535.2.3 uint32_t sensorData_t::timeOfFirstSample

#### 8.535.2.4 uint16_t sensorData_t::timeOffset[64]

#### 8.535.2.5 uint32_t sensorData_t::xAxis[64]

#### 8.535.2.6 uint32_t sensorData_t::yAxis[64]

#### 8.535.2.7 uint32_t sensorData_t::zAxis[64]

## 8.536 slot_t Struct Reference

### Data Fields

- uint32_t uPhyCardStatus
- uint32_t uPhySlotStatus
- uint8_t bLogicalSlot
- uint8_t bICCIDLength
- uint8_t bICCID [255]

### 8.536.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *uPhyCardStatus* | • State of the card in the Pyhsical Slot Status.<br>– 0x00 - Unknown.<br>– 0x01 - Absent.<br>– 0x02 - Present. |

| | |
|---|---|
| *uPhySlotStatus* | • State of the Physical Slot status.<br>– 0x00 Inactive.<br>– 0x01 Activate. |
| *bLogicalSlot* | • Logical Slot associated with this physical slot. THis is valid if the physical slot is active.<br>– 1 - Slot 1.<br>– 2 - Slot 2.<br>– 3 - Slot 3.<br>– 4 - Slot 4.<br>– 5 - Slot 5. |
| *bLogicalSlot* | • Number of sets the sets of ICCCID |
| *bICCID[MAX_ICCID_LENGTH]* | • Contains the ICCID of the card in the physical slot. |

### 8.536.2 Field Documentation

#### 8.536.2.1 uint8_t slot_t::bICCID[255]

#### 8.536.2.2 uint8_t slot_t::bICCIDLength

#### 8.536.2.3 uint8_t slot_t::bLogicalSlot

#### 8.536.2.4 uint32_t slot_t::uPhyCardStatus

#### 8.536.2.5 uint32_t slot_t::uPhySlotStatus

## 8.537 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]

### 8.537.1 Detailed Description

This structure contains information about the SLOTS present.

**Parameters**

| | |
|---|---|
| *cardState* | • Indicates the state of the card for each slot.<br>– 0 - Absent<br>– 1 - Present<br>– 2 - Error |
| *upinState* | • Indicates the state of UPIN.<br>– 0 - Unknown<br>– 1 - Enabled and not verified<br>– 2 - Enabled and verified<br>– 3 - Disabled<br>– 4 - Blocked<br>– 5 - Permanently blocked<br>– 0xFF - Not Available |
| *upinRetries* | • Indicates the number of retries remaining to verify the UPIN.<br>• If 0xFF, information not available. |
| *upukRetries* | • Indicates the number of retries remaining to unblock the UPIN.<br>• If 0xFF, information not available. |
| *errorState* | • Indicates the reason for the card error, and is valid only when the card state is Error<br>– 0 - Unknown<br>– 1 - Power down<br>– 2 - Poll error<br>– 3 - No ATR received<br>– 4 - Volt mismatch<br>– 5 - Parity error<br>– 6 - Unknown; possibly removed<br>– 7 - Card returned technical problems<br>– 0xFF - Not Available<br>• Other values are possible and reserved for future use.<br>• When an unknown value is received, it is to be handled as "Unknown". |
| *numApp* | • Indicates the number of applications available on the card.<br>• The following block is repeated for each application. i.e. AppStatus.<br>• If zero(0) then no AppStatus information exists. |
| *AppStatus[MAX_NO_OF_APPLICATIONS]* | • See appStats for more information. |

## 8.537.2 Field Documentation

### 8.537.2.1 appStats slotInf::AppStatus[10]

#### 8.537.2.2 uint8_t slotInf::cardState

#### 8.537.2.3 uint8_t slotInf::errorState

#### 8.537.2.4 uint8_t slotInf::numApp

#### 8.537.2.5 uint8_t slotInf::upinRetries

#### 8.537.2.6 uint8_t slotInf::upinState

#### 8.537.2.7 uint8_t slotInf::upukRetries

## 8.538 slots_t Struct Reference

### Data Fields

- slot_t uimSlotStatus [255]

### 8.538.1 Detailed Description

This structure contains slots status.

**Parameters**

| | |
|---|---|
| *uimSlotStatus* | • See slot_t |

### 8.538.2 Field Documentation

#### 8.538.2.1 slot_t slots_t::uimSlotStatus[255]

## 8.539 sms_BroadcastConfig Struct Reference

### Data Fields

- uint16_t fromServiceId
- uint16_t toServiceId
- uint8_t selected

### 8.539.1 Detailed Description

This structure contains BroadcastConfig parameters

**Parameters**

| | |
|---|---|
| *fromServiceId* | • Starting point of range of CBM message identifiers |
| *toServiceId* | • Ending point of range of CBM message identifiers |

| | |
|---|---|
| *selected* | • Range of CBM message identifiers indicated by from_service_id and to_service_id<br>– 0x00 – Not selected<br>– 0x01 – Selected |

### 8.539.2 Field Documentation

#### 8.539.2.1 uint16_t sms_BroadcastConfig::fromServiceId

#### 8.539.2.2 uint8_t sms_BroadcastConfig::selected

#### 8.539.2.3 uint16_t sms_BroadcastConfig::toServiceId

## 8.540 sms_CDMABroadcastConfig Struct Reference

### Data Fields

- uint16_t serviceCategory
- uint16_t language
- uint8_t selected

### 8.540.1 Detailed Description

This structure contains CDMABroadcastConfig parameters

**Parameters**

| | |
|---|---|
| *serviceCategory* | • Service category |
| *language* | • Language |
| *selected* | • Specified service_category and language<br>– 0x00 – Not selected<br>– 0x01 – Selected |

### 8.540.2 Field Documentation

#### 8.540.2.1 uint16_t sms_CDMABroadcastConfig::language

#### 8.540.2.2 uint8_t sms_CDMABroadcastConfig::selected

#### 8.540.2.3 uint16_t sms_CDMABroadcastConfig::serviceCategory

## 8.541 sms_getIndicationReg Struct Reference

### Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

### 8.541.1 Detailed Description

This structure contains Get Indication Register Response parameters

**Parameters**

| | |
|---|---|
| *pRegTrans-LayerInfoEvt* | -<br>• Optional parameter indicating registration status of transport layer information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled |
| *pRegTransNW-RegInfoEvt* | -<br>• Optional parameter indicating registration status of transport network registration information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled |
| *pRegCallStat-InfoEvt* | -<br>• Optional parameter indicating registration status of call status information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled |

### 8.541.2 Field Documentation

#### 8.541.2.1 uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt

#### 8.541.2.2 uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt

#### 8.541.2.3 uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt

## 8.542 sms_getMsgWaitingInfo Struct Reference

### Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]

### 8.542.1 Detailed Description

This structure contains Get Message Waiting Info Response parameters

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of sets of the elements in structure sms_messageWaitingInfoContent |
| *pMsgWaitInfo* | • Pointer to structure of sms_messageWaitingInfoContent.<br>– See sms_messageWaitingInfoContent for more information. |

### 8.542.2 Field Documentation

#### 8.542.2.1 sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo[0xFF]

#### 8.542.2.2 uint8_t sms_getMsgWaitingInfo::numInstances

## 8.543 sms_getTransLayerInfo Struct Reference

### Data Fields

- uint8_t ∗ pRegInd
- sms_transLayerInfo ∗ pTransLayerInfo

### 8.543.1 Detailed Description

This structure contains Get Transport Layer Info Response parameters

**Parameters**

| | |
|---|---|
| *pRegInd* | -<br>• Optional parameter indicating if transport layer is registered<br>• Values:<br>– 0x00 - Transport layer is not registered<br>– 0x01 - Transport layer is registered<br>• returns a default value 0xFF if no response is received from the device. |
| *pTransLayerInfo* | • Pointer to structure of sms_transLayerInfo.<br>– Optional parameter<br>– See sms_transLayerInfo for more information<br>• returns a default value 0xFF for parameter values if no response is received from the device. |

### 8.543.2 Field Documentation

#### 8.543.2.1 uint8_t∗ sms_getTransLayerInfo::pRegInd

#### 8.543.2.2 sms_transLayerInfo∗ sms_getTransLayerInfo::pTransLayerInfo

## 8.544 sms_getTransNWRegInfo Struct Reference

### Data Fields

- uint8_t * pRegStatus

### 8.544.1 Detailed Description

This structure contains transport network registration info parameter

**Parameters**

| | |
|---|---|
| *pRegStatus* | -<br>• Optional parameter indicating transport layer network registration status<br>• Values:<br>– 0x00 - No service<br>– 0x01 - In progress<br>– 0x02 - Failed<br>– 0x03 - Limited Service<br>– 0x04 - Full Service<br>• returns a default value 0xFF if no response is received from the device. |

### 8.544.2 Field Documentation

#### 8.544.2.1 uint8_t* sms_getTransNWRegInfo::pRegStatus

## 8.545 sms_maxStorageSizeReq Struct Reference

### Data Fields

- uint8_t storageType
- uint8_t * pMessageMode

### 8.545.1 Detailed Description

This structure contains get store max size request parameters

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV |
| *pMessage-Mode(optional* | parameter)<br>• 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS) |

**Note**

The Message Mode TLV must be included if the device is capable of supporting more than one protocol

### 8.545.2 Field Documentation

#### 8.545.2.1 uint8_t* sms_maxStorageSizeReq::pMessageMode

#### 8.545.2.2 uint8_t sms_maxStorageSizeReq::storageType

## 8.546 sms_maxStorageSizeResp Struct Reference

### Data Fields

- uint32_t maxStorageSize
- uint32_t freeSlots

### 8.546.1 Detailed Description

This structure contains get store max size response parameters

**Parameters**

| | |
|---|---|
| *maxStorageSize* | -<br>• Memory Store Size |
| *freeSlots* | -<br>• Optional parameter indicating how much Memory is available<br>• returns a default value 0xFFFFFFFF for parameter values if no response is received from the device. |

### 8.546.2 Field Documentation

#### 8.546.2.1 uint32_t sms_maxStorageSizeResp::freeSlots

#### 8.546.2.2 uint32_t sms_maxStorageSizeResp::maxStorageSize

## 8.547 sms_messageWaitingInfoContent Struct Reference

### Data Fields

- uint8_t msgType
- uint8_t activeInd
- uint8_t msgCount

### 8.547.1 Detailed Description

This structure contains message waiting information per instance

**Parameters**

| | |
|---|---|
| *msgType* | • Message type<br>– 0x00 - MWI_MESSAGE_TYPE_VOICMAIL - Voicemail<br>– 0x01 - MWI_MESSAGE_TYPE_FAX - Fax<br>– 0X02 - MWI_MESSAGE_TYPE_EMAIL - Email<br>– 0x03 - MWI_MESSAGE_TYPE_OTHER - Other<br>– 0x04 - MWI_MESSAGE_TYPE_VIDEOMAIL - Videomail |
| *activeInd* | • Indicates whether the indication is active<br>– 0x00 - Inactive<br>– 0x01 - Active |
| *msgCount* | • Number of messages |

### 8.547.2 Field Documentation

#### 8.547.2.1 uint8_t sms_messageWaitingInfoContent::activeInd

#### 8.547.2.2 uint8_t sms_messageWaitingInfoContent::msgCount

#### 8.547.2.3 uint8_t sms_messageWaitingInfoContent::msgType

## 8.548 sms_msgProtocolResp Struct Reference

### Data Fields

- uint8_t msgProtocol

### 8.548.1 Detailed Description

This structure contains get message protocol response parameters

**Parameters**

| | |
|---|---|
| *msgProtocol* | -<br>• Message Protocol<br>• Values:<br>– 0x00 - MESSAGE_PROTOCOL_CDMA<br>– 0x01 - MESSAGE_PROTOCOL_WCDMA |

### 8.548.2 Field Documentation

#### 8.548.2.1 uint8_t sms_msgProtocolResp::msgProtocol

## 8.549 sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference

### Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_CDMABroadcastConfig CDMABroadcastConfig [0x05]

### 8.549.1 Detailed Description

This structure contains the 3GPP2 Broadcast Configuration Information parameters

**Parameters**

| | |
|---|---|
| *activated_ind* | • Broadcast SMS<br>– 0x00 - Deactivated<br>– 0x01 - Activated |
| *num_instances* | • Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table.<br>– serviceCategory<br>– language<br>– selected |
| *broadcastConfig* | • A CDMABroadcastConfig structure array.<br>• Further defined by the structure sms_CDMABroadcastConfig |

### 8.549.2 Field Documentation

#### 8.549.2.1 uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind

#### 8.549.2.2 sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

#### 8.549.2.3 uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances

## 8.550 sms_qaQmi3GPPBroadcastCfgInfo Struct Reference

### Data Fields

- uint8_t activated_ind
- uint16_t num_instances
- sms_BroadcastConfig broadcastConfig [0x05]

### 8.550.1 Detailed Description

This structure contains the 3GPP Broadcast Configuration Information parameters

**Parameters**

| | |
|---|---|
| *activated_ind* | • Broadcast SMS<br>– 0x00 - Deactivated<br>– 0x01 - Activated |
| *num_instances* | • Number of sets (N) of parameters Following each set describes one entry in the broadcast configuration table.<br>– fromServiceId<br>– toServiceId<br>– selected |
| *broadcastConfig* | • A BroadcastConfig structure array.<br>• Further defined by the structure sms_BroadcastConfig |

### 8.550.2 Field Documentation

#### 8.550.2.1 uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind

#### 8.550.2.2 sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

#### 8.550.2.3 uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances

## 8.551 sms_routeEntry Struct Reference

### Data Fields

- uint8_t messageType
- uint8_t messageClass
- uint8_t routeStorage
- uint8_t receiptAction

### 8.551.1 Detailed Description

This structure contains SMS route entry details

**Parameters**

| | |
|---|---|
| *messageType* | -<br>• Message type matching this route<br>• Values:<br>– 0x00 - MESSAGE_TYPE_POINT_TO_POINT |
| *messageClass* | -<br>• Message Class<br>• Values:<br>– 0x00 - MESSAGE_CLASS_0<br>– 0x01 - MESSAGE_CLASS_1<br>– 0x02 - MESSAGE_CLASS_2<br>– 0x03 - MESSAGE_CLASS_3<br>– 0x04 - MESSAGE_CLASS_NONE<br>– 0x05 - MESSAGE_CLASS_CDMA |
| *routeStorage* | -<br>• If the receiptAction is store where to store the message<br>• Values:<br>– 0x00 - STORAGE_TYPE_UIM<br>– 0x01 - STORAGE_TYPE_NV<br>– 0xFF - STORAGE_TYPE_NONE |
| *receiptAction* | -<br>• Action to be taken on receipt of a message matching the specified type and class for this route<br>• Values:<br>– 0x00 - DISCARD (discarded without notification)<br>– 0x01 - STORE AND NOTIFY (stored and notified to the registered clients)<br>– 0x02 - TRANSFER ONLY (transferred to the client, client expected to send the ACK)<br>– 0x03 - TRANSFER AND ACK (transferred to the client, device expected to send the ACK) |

### 8.551.2 Field Documentation

#### 8.551.2.1 uint8_t sms_routeEntry::messageClass

#### 8.551.2.2 uint8_t sms_routeEntry::messageType

#### 8.551.2.3 uint8_t sms_routeEntry::receiptAction

#### 8.551.2.4 uint8_t sms_routeEntry::routeStorage

## 8.552 sms_sendAsyncsmsParams Struct Reference

### Data Fields

- uint32_t messageFormat
- uint32_t messageSize
- uint8_t * pMessage
- uint8_t * pForceOnDC
- uint8_t * pServiceOption
- uint8_t * pFollowOnDC

- uint8_t * pLinktimer
- uint8_t * pSmsOnIms
- uint8_t * pRetryMessage
- uint32_t * pRetryMessageId
- uint32_t * pUserData

### 8.552.1 Detailed Description

This structure contains SMS parameters

**Parameters**

| | |
|---|---|
| *messageFormat* | • Message format<br>• Values:<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |
| *messageSize* | • The length of the message contents in bytes |
| *pMessage* | • The message contents |
| *pForceOnDC* | • Force the message to be sent on the CDMA dedicated channel.<br>• Values:<br>– 0x00 - Do not care about the channel on which the message is sent<br>– 0x01 - Request to send the message over the dedicated channel |
| *pServiceOption* | • Service option:<br>• Values:<br>– 0x00 - SO_AUTO - AUTO (choose the best service option)<br>– 0x06 - SO_6 - Service option 6<br>– 0x0E - SO_14 - Service option 14 |
| *pFollowOnDC* | • Flag to request not to disconnect the CDMA dedicated channel after the send operation is complete.<br>• This TLV can be included if more messages are expected to follow.<br>• Values:<br>– 0x01 - FOLLOW_ON_DC_ON - On (don't disconnect after send operation) Any value other than 0x01 is treated as an absence of this TLV. |
| *pLinktimer* | • Keeps the GW SMS link open for the specified number of seconds; can be enabled if more messages are expected to follow |

| | |
|---|---|
| *pSmsOnIms* | • Indicates whether the message is to be sent on IMS.<br>• Values:<br>– 0x00 - Message is not to be sent on IMS<br>– 0x01 - Message is to be sent on IMS<br>– 0x02 to 0xFF - Reserved |
| *pRetryMessage* | • Indicates this message is a retry message.<br>• Values:<br>– 0x01 - WMS_MESSAGE_IS_A_RETRY - Message is a retry message Note: Any value other than 0x01 in this field is treated as an absence of this TLV. |
| *pRetryMessageId* | • Message ID to be used in the retry message.<br>• The message ID specified here is used instead of the messsage ID encoded in the raw message. |
| *pUserData* | • Enables the control point to associate the request with the corresponding indication.<br>• The control point might send numerous requests.<br>• This TLV will help the control point to identify the request for which the received indication belongs. |

### 8.552.2 Field Documentation

#### 8.552.2.1 uint32_t sms_sendAsyncsmsParams::messageFormat

#### 8.552.2.2 uint32_t sms_sendAsyncsmsParams::messageSize

#### 8.552.2.3 uint8_t* sms_sendAsyncsmsParams::pFollowOnDC

#### 8.552.2.4 uint8_t* sms_sendAsyncsmsParams::pForceOnDC

#### 8.552.2.5 uint8_t* sms_sendAsyncsmsParams::pLinktimer

#### 8.552.2.6 uint8_t* sms_sendAsyncsmsParams::pMessage

#### 8.552.2.7 uint8_t* sms_sendAsyncsmsParams::pRetryMessage

#### 8.552.2.8 uint32_t* sms_sendAsyncsmsParams::pRetryMessageId

#### 8.552.2.9 uint8_t* sms_sendAsyncsmsParams::pServiceOption

#### 8.552.2.10 uint8_t* sms_sendAsyncsmsParams::pSmsOnIms

#### 8.552.2.11 uint32_t* sms_sendAsyncsmsParams::pUserData

## 8.553 sms_setIndicationReg Struct Reference

### Data Fields

- uint8_t * pRegTransLayerInfoEvt
- uint8_t * pRegTransNWRegInfoEvt
- uint8_t * pRegCallStatInfoEvt

### 8.553.1 Detailed Description

This structure contains Indication Register request parameters

**Parameters**

| | |
|---|---|
| *pRegTrans-LayerInfoEvt* | - <br>• Optional parameter indicating registration status of transport layer information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>– NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device |
| *pRegTransNW-RegInfoEvt* | - <br>• Optional parameter indicating registration status of transport network registration information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>– NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device |
| *pRegCallStat-InfoEvt* | - <br>• Optional parameter indicating registration status of call status information events<br>• Values:<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>– NULL - No change - specifying NULL indicates that the device will continue to use the existing setting (disable/enable) which has been previously set for the device |

### 8.553.2 Field Documentation

#### 8.553.2.1 uint8_t* sms_setIndicationReg::pRegCallStatInfoEvt

#### 8.553.2.2 uint8_t* sms_setIndicationReg::pRegTransLayerInfoEvt

#### 8.553.2.3 uint8_t* sms_setIndicationReg::pRegTransNWRegInfoEvt

## 8.554 sms_setRoutesReq Struct Reference

### Data Fields

- uint16_t numOfRoutes
- sms_routeEntry routeList [0x0A]
- uint8_t * pTransferStatusReport

### 8.554.1 Detailed Description

This structure contains SMS route request parameters

**Parameters**

| | |
|---|---|
| *numOfRoutes* | - <br>• Number of sets of the following element |
| *routeList* | - <br>• Array containing the set of sms_routeEntry |
| *pTransferStatus-Report* | - <br>• 0x01 - Status report are transferred to the client (optional) |

### 8.554.2 Field Documentation

#### 8.554.2.1 uint16_t sms_setRoutesReq::numOfRoutes

#### 8.554.2.2 uint8_t∗ sms_setRoutesReq::pTransferStatusReport

#### 8.554.2.3 sms_routeEntry sms_setRoutesReq::routeList[0x0A]

## 8.555 sms_transLayerInfo Struct Reference

### Data Fields

- uint8_t TransType
- uint8_t TransCap

### 8.555.1 Detailed Description

This structure contains Transport Layer Information

**Parameters**

| | |
|---|---|
| *TransType* | • Transport Type<br>  – 0x00 - IMS |
| *TransCap* | • Transport Capability<br>• Values:<br>  – 0x00 - CDMA<br>  – 0x01 - GW |

### 8.555.2 Field Documentation

#### 8.555.2.1 uint8_t sms_transLayerInfo::TransCap

#### 8.555.2.2 uint8_t sms_transLayerInfo::TransType

## 8.556 sMSCAddressInfo Struct Reference

### Data Fields

- uint8_t length
- uint8_t data [256]

### 8.556.1 Detailed Description

This structure contains SMSC information

**Parameters**

| | |
|---|---|
| *length* | • Number of sets of following element |
| *data* | • SMSC address |

### 8.556.2 Field Documentation

#### 8.556.2.1 uint8_t sMSCAddressInfo::data[256]

#### 8.556.2.2 uint8_t sMSCAddressInfo::length

## 8.557 sMSCAddressTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSCAddressInfo SMSCInfo

### 8.557.1 Detailed Description

This structure contains SMSC TLV information

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *SMSCInfo* | • SMSC Address<br>• See sMSCAddressInfo for more information |

### 8.557.2 Field Documentation

#### 8.557.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

#### 8.557.2.2 uint8_t sMSCAddressTlv::TlvPresent

## 8.558 sMSEtwsMessageInfo Struct Reference

### Data Fields

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

### 8.558.1 Detailed Description

This structure contains ETWS information

**Parameters**

| | |
|---|---|
| *notificationType* | • Message mode 0x00 - Primary 0x01 - Secondary GSM 0x02 - Secondary UMTS |
| *length* | • Number of sets of following elements |
| *data* | • Raw message data |

### 8.558.2 Field Documentation

#### 8.558.2.1 uint8_t sMSEtwsMessageInfo::data[1254]

#### 8.558.2.2 uint16_t sMSEtwsMessageInfo::length

#### 8.558.2.3 uint8_t sMSEtwsMessageInfo::notificationType

## 8.559 sMSEtwsMessageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSEtwsMessageInfo EtwsMessageInfo

### 8.559.1 Detailed Description

This structure contains ETWS TLV information

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *EtwsMessage-Info* | • ETWS Message<br>• See sMSEtwsMessageInfo for more information |

### 8.559.2 Field Documentation

#### 8.559.2.1 sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo

#### 8.559.2.2 uint8_t sMSEtwsMessageTlv::TlvPresent

## 8.560 sMSEtwsPlmnInfo Struct Reference

### Data Fields

- uint16_t mobileCountryCode
- uint16_t mobileNetworkCode

### 8.560.1 Detailed Description

This structure contains ETWS PLMN information

**Parameters**

| | |
|---|---|
| *mobileCountry-Code* | • 16 bit representation of MCC value range : 0 -999 |
| *mobileNetwork-Code* | • 16 bit representation of MNC value range : 0 -999 |

### 8.560.2 Field Documentation

#### 8.560.2.1 uint16_t sMSEtwsPlmnInfo::mobileCountryCode

#### 8.560.2.2 uint16_t sMSEtwsPlmnInfo::mobileNetworkCode

## 8.561 sMSMessageModeInfo Struct Reference

### Data Fields

- uint8_t messageMode

### 8.561.1 Detailed Description

This structure contains message mode information.

**Parameters**

| | |
|---|---|
| *messageMode* | Message Mode |

### 8.561.2 Field Documentation

#### 8.561.2.1 uint8_t sMSMessageModeInfo::messageMode

## 8.562 sMSMTMessageInfo Struct Reference

### Data Fields

- uint32_t storageType
- uint32_t messageIndex

### 8.562.1 Detailed Description

This structure contains MT message information.

**Parameters**

| | |
|---|---|
| *storageType* | memory storage 0x00-UIM 0x01-NV |
| *messageIndex* | MT Message index |

### 8.562.2 Field Documentation

#### 8.562.2.1 uint32_t sMSMTMessageInfo::messageIndex

#### 8.562.2.2 uint32_t sMSMTMessageInfo::storageType

## 8.563 sMSOnIMSInfo Struct Reference

### Data Fields

- uint8_t smsOnIMS

### 8.563.1 Detailed Description

This structure contains SMS on IMS information

**Parameters**

| | |
|---|---|
| *smsOnIMS* | SMS on IMS |

### 8.563.2 Field Documentation

#### 8.563.2.1 uint8_t sMSOnIMSInfo::smsOnIMS

## 8.564 sMSOnIMSTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSOnIMSInfo IMSInfo

### 8.564.1 Detailed Description

This structure contains SMS on IMS TLV information.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *IMSInfo* | • SMS on IMS<br>• See sMSOnIMSInfo for more information |

### 8.564.2 Field Documentation

#### 8.564.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

#### 8.564.2.2 uint8_t sMSOnIMSTlv::TlvPresent

## 8.565 sMSTransferRouteMTMessageInfo Struct Reference

### Data Fields

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

### 8.565.1 Detailed Description

This structure contains Transfer route MT message information.

**Parameters**

| | |
|---|---|
| *ackIndicator* | • Parameter to indicate if ACK must be sent by the control point 0x00 - Send ACK 0x01 - Do not send ACK |
| *transactionID* | • Transaction ID of the message |
| *format* | • Message format 0x00 - CDMA 0x02 - 0x05 - Reserved 0x06 - GW_PP 0x07 - GW_BC |
| *length* | • Length of the raw message. This length should not exceed the maximum WMS payload length of 256 bytes |
| *data* | • Raw message data |

### 8.565.2 Field Documentation

#### 8.565.2.1 uint8_t sMSTransferRouteMTMessageInfo::ackIndicator

#### 8.565.2.2 uint8_t sMSTransferRouteMTMessageInfo::data[256]

#### 8.565.2.3 uint8_t sMSTransferRouteMTMessageInfo::format

#### 8.565.2.4 uint16_t sMSTransferRouteMTMessageInfo::length

#### 8.565.2.5 uint32_t sMSTransferRouteMTMessageInfo::transactionID

## 8.566 swi_uint256_t Struct Reference

### Data Fields

- uint16_t word [16]

### 8.566.1 Detailed Description

SWI 256 bit data type

### 8.566.2 Field Documentation

#### 8.566.2.1 uint16_t swi_uint256_t::word[16]

## 8.567 swiaudio_PCMparams Struct Reference

### Data Fields

- uint8_t iFaceTabLen
- uint8_t iFaceTab [255]

### 8.567.1 Detailed Description

This structure contains the PCM parameters.

**Parameters**

| | |
|---|---|
| *iFaceTabLen* | • Number of sets of iface table |
| *iFaceTab* | • Physical Interface Parameters<br>• See qaGobiApiTableSwiAudio.h for more information on physical interface parameters |

### 8.567.2 Field Documentation

#### 8.567.2.1 uint8_t swiaudio_PCMparams::iFaceTab[255]

#### 8.567.2.2 uint8_t swiaudio_PCMparams::iFaceTabLen

## 8.568 swidms_ehrpdMTUSizeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t ehrpdMTUSize

### 8.568.1 Detailed Description

This structure contains the parameters for EHRPD MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *ehrpdMTUSize* | EHRPD size<br>• Size of EHRPD MTU |

### 8.568.2 Field Documentation

#### 8.568.2.1 uint16_t swidms_ehrpdMTUSizeTlv::ehrpdMTUSize

#### 8.568.2.2 uint8_t swidms_ehrpdMTUSizeTlv::TlvPresent

## 8.569 swidms_hrpdMTUSizeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t hrpdMTUSize

### 8.569.1 Detailed Description

This structure contains the parameters for HRPD MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *hrpdMTUSize* | HRPD size<br>• Size of HRPD MTU |

### 8.569.2 Field Documentation

#### 8.569.2.1 uint16_t swidms_hrpdMTUSizeTlv::hrpdMTUSize

#### 8.569.2.2 uint8_t swidms_hrpdMTUSizeTlv::TlvPresent

## 8.570 swidms_mtuSize3gppTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t MTUSize3gpp

### 8.570.1 Detailed Description

This structure contains the parameters for 3GPP MTU Size.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *MTUSize3gpp* | 3GPP MTU size<br>• Size of 3GPP MTU |

### 8.570.2 Field Documentation

#### 8.570.2.1 uint16_t swidms_mtuSize3gppTlv::MTUSize3gpp

#### 8.570.2.2 uint8_t swidms_mtuSize3gppTlv::TlvPresent

## 8.571 swidms_usbMTUSizeTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t UsbMTUSize

### 8.571.1 Detailed Description

This structure contains the parameters for USB MTU Size

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Tlv Present. |
| *UsbMTUSize* | USB MTU size<br>• Size of USB MTU |

### 8.571.2 Field Documentation

#### 8.571.2.1 uint8_t swidms_usbMTUSizeTlv::TlvPresent

#### 8.571.2.2 uint16_t swidms_usbMTUSizeTlv::UsbMTUSize

## 8.572 tdscdmaSigInfoExt Struct Reference

### Data Fields

- float rssi
- float rscp
- float ecio
- float sinr

### 8.572.1 Detailed Description

This structure contains the TDSCDMA Signal Strength Info Extended

**Parameters**

| | |
|---|---|
| *rssi* | • Measured RSSI in dB |
| *rscp[Optional]* | • Measured RSCP in dBm |
| *ecio[Optional]* | • Measured ECIO in dBm. |
| *sinr[Optional]* | • Measured SINR in dB. -15 dB is sent to clients if the actual SINR is less than -15 dB |

### 8.572.2 Field Documentation

#### 8.572.2.1 float tdscdmaSigInfoExt::ecio

#### 8.572.2.2 float tdscdmaSigInfoExt::rscp

#### 8.572.2.3 float tdscdmaSigInfoExt::rssi

#### 8.572.2.4 float tdscdmaSigInfoExt::sinr

## 8.573 tempData_t Struct Reference

### Data Fields

- uint32_t timeSource
- uint32_t timeOfFirstSample
- uint8_t temperatureDataLen
- uint16_t timeOffset [64]
- uint32_t temperature [64]

### 8.573.1 Detailed Description

This structure specifies information regarding the Temperature Data. Please check has_<Param_Name> field for presence of optional parameters

**Parameters**

| | |
|---|---|
| *timeSource* | • Time source of the sensor data<br>• Valid values<br>– 0 - Sensor time source is unspecified<br>– 1 - Time source is common between the sensors and the location engine |
| *timeOfFirst-Sample* | • Denotes a full 32-bit time stamp of the first (oldest) sample in this message.<br>• The time stamp is in the time reference scale that is used by the sensor time source.<br>• Units - Milliseconds |
| *temperature-DataLen* | • Number of sets of the following elements<br>– timeOffset<br>– temperature |
| *timeOffset* | • Sample time offset<br>• Units - Milliseconds |
| *temperature* | • Sensor temperature.<br>• Type - Floating point<br>• Units - Degrees Celsius<br>• Range -50 to +100.00 |

### 8.573.2 Field Documentation

#### 8.573.2.1 uint32_t tempData_t::temperature[64]

#### 8.573.2.2 uint8_t tempData_t::temperatureDataLen

#### 8.573.2.3 uint32_t tempData_t::timeOfFirstSample

#### 8.573.2.4 uint16_t tempData_t::timeOffset[64]

#### 8.573.2.5 uint32_t tempData_t::timeSource

## 8.574 tmd_mitigationDevList Struct Reference

### Data Fields

- uint8_t mitigationDevIdLen
- char mitigationDevId [255]
- uint8_t maxMitigationLevel

### 8.574.1 Detailed Description

This structure contains mitigation devices list

**Parameters**

| | |
|---|---|
| *mitigationDevId-Len* | • Number of sets of the following elements<br>– mitigation_dev_id |
| *mitigationDevId* | • Mitigation device ID. |
| *maxMitigation-Level* | • Maximum valid mitigation level.<br>• Valid range - 0 to max_mitigation_level. |

### 8.574.2 Field Documentation

#### 8.574.2.1 uint8_t tmd_mitigationDevList::maxMitigationLevel

#### 8.574.2.2 char tmd_mitigationDevList::mitigationDevId[255]

#### 8.574.2.3 uint8_t tmd_mitigationDevList::mitigationDevIdLen

## 8.575 transferRouteMessageTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- sMSTransferRouteMTMessageInfo TransferRouteMTMessageInfo

### 8.575.1 Detailed Description

This structure contains Transfer route MT message TLV information.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • Boolean indicating the presence of the TLV in the QMI response |
| *TransferRouteM-TMessageInfo* | • Transfer Route MT Message<br>• See sMSTransferRouteMTMessageInfo for more information |

### 8.575.2 Field Documentation

#### 8.575.2.1 uint8_t transferRouteMessageTlv::TlvPresent

#### 8.575.2.2 sMSTransferRouteMTMessageInfo transferRouteMessageTlv::TransferRouteMTMessageInfo

## 8.576 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

### 8.576.1 Detailed Description

This structure contains Application Status Information loaded on the card.

**Parameters**

| | |
|---|---|
| *appType* | • Indicates the type of the application.<br>– 0 - Unknown<br>– 1 - SIM card<br>– 2 - USIM application<br>– 3 - RUIM card<br>– 4 - CSIM application<br>– 5 - ISIM application<br>• Other values are reserved for the future and are to be handled as "Unknown". |
| *appState* | • Indicates the state of the application.<br>– 0 - Unknown<br>– 1 - Detected<br>– 2 - PIN1 or UPIN is required<br>– 3 - PUK1 or PUK for UPIN is required<br>– 4 - Personalization state must be checked<br>– 5 - PIN1 is blocked<br>– 6 - Illegal<br>– 7 - Ready |
| *persoState* | • Indicates the state of the personalization for the application.<br>– 0 - Unknown<br>– 1 - Personalization operation is in progress<br>– 2 - Ready<br>– 3 - Personalization code is required<br>– 4 - PUK for personalization code is required<br>– 5 - Permanently blocked |

| | |
|---|---|
| *persoFeature* | • Indicates the personalization feature.<br>• This applies only when a personalization code is required to deactivate or unblock personalization.<br>– 0 - GW network personalization<br>– 1 - GW network subset personalization<br>– 2 - GW service provider personalization<br>– 3 - GW corporate personalization<br>– 4 - GW UIM personalization<br>– 5 - 1X network type 1 personalization<br>– 6 - 1X network type 2 personalization<br>– 7 - 1X HRPD personalization<br>– 8 - 1X service provider personalization<br>– 9 - 1X corporate personalization<br>– 10 - 1X RUIM personalization<br>– 11 - Unknown |
| *persoRetries* | • Indicates the number of retries remaining to disable the personalization. |
| *persoUnblockRetries* | • Indicates the number of retries remaining to unblock the personalization. |
| *aidLength* | • Number of sets of the following elements. i.e. aidVal<br>• If zero(0) then no aidVal information exists. |
| *aidVal[MAX_DESCRIPTION_LENGTH]* | • Application identifier value. |
| *univPin* | • Indicates whether UPIN replaces PIN1.<br>– 0 - PIN1 is used<br>– 1 - UPIN replaces PIN1 |
| *pin1State* | • Indicates the state of PIN1.<br>– 0 - Unknown<br>– 1 - Enabled and not verified<br>– 2 - Enabled and verified<br>– 3 - Disabled<br>– 4 - Blocked<br>– 5 - Permanently blocked |
| *pin1Retries* | • Indicates the number of retries remaining to verify PIN1. |
| *puk1Retries* | • Indicates the number of retries remaining to unblock PIN1. |

| | |
|---|---|
| *pin2State* | • Indicates the state of PIN2.<br>– 0 - Unknown<br>– 1 - Enabled and not verified<br>– 2 - Enabled and verified<br>– 3 - Disabled<br>– 4 - Blocked<br>– 5 - Permanently blocked |
| *pin2Retries* | • Indicates the number of retries remaining to verify PIN2. |
| *puk2Retries* | • Indicates the number of retries remaining to unblock PIN2. |

### 8.576.2 Field Documentation

#### 8.576.2.1 uint8_t uim_appStatus::aidLength

#### 8.576.2.2 uint8_t uim_appStatus::aidVal[255]

#### 8.576.2.3 uint8_t uim_appStatus::appState

#### 8.576.2.4 uint8_t uim_appStatus::appType

#### 8.576.2.5 uint8_t uim_appStatus::persoFeature

#### 8.576.2.6 uint8_t uim_appStatus::persoRetries

#### 8.576.2.7 uint8_t uim_appStatus::persoState

#### 8.576.2.8 uint8_t uim_appStatus::persoUnblockRetries

#### 8.576.2.9 uint8_t uim_appStatus::pin1Retries

#### 8.576.2.10 uint8_t uim_appStatus::pin1State

#### 8.576.2.11 uint8_t uim_appStatus::pin2Retries

#### 8.576.2.12 uint8_t uim_appStatus::pin2State

#### 8.576.2.13 uint8_t uim_appStatus::puk1Retries

#### 8.576.2.14 uint8_t uim_appStatus::puk2Retries

#### 8.576.2.15 uint8_t uim_appStatus::univPin

## 8.577 uim_authenticateResult Struct Reference

### Data Fields

- uint16_t contentLen

- uint8_t content [1024]

### 8.577.1 Detailed Description

This structure contains the information about the authenticate result.

**Parameters**

| | |
|---|---|
| *contentLen* | • Length of the following elements i.e. content. |
| *content[UIM_MAX_CONTENT_LENGTH]* | • Authenticate data.<br>• This value is a sequence of bytes returned from the card. |

### 8.577.2 Field Documentation

#### 8.577.2.1 uint8_t uim_authenticateResult::content[1024]

#### 8.577.2.2 uint16_t uim_authenticateResult::contentLen

## 8.578 uim_authenticationData Struct Reference

### Data Fields

- uint8_t context
- uint16_t dataLen
- uint8_t data [1024]

### 8.578.1 Detailed Description

This structure contains the Session Information.

**Parameters**

| | |
|---|---|
| *context* | • Authenticate context.<br>– 0 - Runs the GSM algorithm (valid only on a 2GSIM card)<br>– 1 - Runs the CAVE algorithm (valid only on a RUIM card)<br>– 2 - GSM security context (valid only on a USIM application)<br>– 3 - 3G security context (valid only on a USIM application)<br>– 4 - VGCS/VBS security context (valid only on aUSIM application)<br>– 5 - GBA security context, Bootstrapping mode (valid only on a USIM or ISIM application)<br>– 6 - GBA security context, NAF Derivation mode (valid only on a USIM or ISIM application)<br>– 7 - MBMS security context, MSK Update mode (valid only on a USIM application)<br>– 8 - MBMS security context, MTK Generation mode (valid only on a USIM application)<br>– 9 - MBMS security context, MSK Deletion mode (valid only on a USIM application)<br>– 10 - MBMS security context, MUK Deletion mode (valid only on a USIM application)<br>– 11 - IMS AKA security context (valid only on aISIM application)<br>– 12 - HTTP-digest security context (valid only onan ISIM application)<br>– 13 - Compute IP authentication, CHAP (valid onlyon RUIM or CSIM)<br>– 14 - Compute IP authentication, MN-HA authenticator (valid only on RUIM or CSIM)<br>– 15 - Compute IP authentication, MIP-RRQ hash (valid only on RUIM or CSIM)<br>– 16 - Compute IP authentication, MN-AAA authenticator (valid only on RUIM or CSIM)<br>– 17 - Compute IP authentication, HRPD access authenticator (valid only on RUIM or CSIM)<br>• Other values are possible and reserved for future use. |
| *dataLen* | • Length of the following elements i.e. data. |
| *data[UIM_MAX_DESCRIPTION_LENGTH]* | • Authenticate Data. |

### 8.578.2 Field Documentation

#### 8.578.2.1 uint8_t uim_authenticationData::context

#### 8.578.2.2 uint8_t uim_authenticationData::data[1024]

#### 8.578.2.3 uint16_t uim_authenticationData::dataLen

## 8.579 uim_cardResult Struct Reference

### Data Fields

- uint8_t sw1
- uint8_t sw2

### 8.579.1 Detailed Description

This structure contains the information about the card result.

**Parameters**

| | |
|---|---|
| *sw1* | • SW1 received from the card. |
| *sw2* | • SW2 received from the card. |

### 8.579.2 Field Documentation

#### 8.579.2.1 uint8_t uim_cardResult::sw1

#### 8.579.2.2 uint8_t uim_cardResult::sw2

## 8.580 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]

### 8.580.1 Detailed Description

This structure contains Card Status Information.

**Parameters**

| | |
|---|---|
| *indexGwPri* | • Index of the primary GW provisioning application.<br>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).<br>• The value 0xFFFF identifies when the session does not exist. |
| *index1xPri* | • Index of the primary 1X provisioning application.<br>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).<br>• The value 0xFFFF identifies when the session does not exist. |
| *indexGwSec* | • Index of the secondary GW provisioning application.<br>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).<br>• The value 0xFFFF identifies when the session does not exist. |

| | |
|---|---|
| *index1xSec* | • Index of the secondary GW provisioning application.<br>• The most significant byte indicates the slot (starting from 0), while the least significant byte indicates the application for that slot (starting from 0).<br>• The value 0xFFFF identifies when the session does not exist. |
| *numSlot* | • Indicates the number of slots available on the device.<br>• The following block is repeated for each slot. i.e. cardState<br>• If zero(0) then no cardState information exists. |
| *SlotInfo* | • See uim_slotInfo for more information. |

### 8.580.2 Field Documentation

#### 8.580.2.1 uint16_t uim_cardStatus::index1xPri

#### 8.580.2.2 uint16_t uim_cardStatus::index1xSec

#### 8.580.2.3 uint16_t uim_cardStatus::indexGwPri

#### 8.580.2.4 uint16_t uim_cardStatus::indexGwSec

#### 8.580.2.5 uint8_t uim_cardStatus::numSlot

#### 8.580.2.6 uim_slotInfo uim_cardStatus::SlotInfo[5]

## 8.581 uim_changeUIMPIN Struct Reference

### Data Fields

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

### 8.581.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pinID* | • Indicates the PIN ID to be changed.<br>– 1 - PIN1 (also called PIN)<br>– 2 - PIN2<br>– 3 - Universal PIN<br>– 4 - Hidden key |

| | |
|---|---|
| *oldPINLen* | • Length of the following elements i.e. old pin value. |
| *oldPINVal[MAX_DESCRIPTION_LENGTH]* | • Old PIN value.<br>• This value is a sequence of ASCII characters. |
| *pinLen* | • Length of the following elements i.e. new pin value. |
| *pinVal[MAX_DESCRIPTION_LENGTH]* | • New PIN value.<br>• This value is a sequence of ASCII characters. |

### 8.581.2 Field Documentation

#### 8.581.2.1 uint8_t uim_changeUIMPIN::oldPINLen

#### 8.581.2.2 uint8_t uim_changeUIMPIN::oldPINVal[255]

#### 8.581.2.3 uint8_t uim_changeUIMPIN::pinID

#### 8.581.2.4 uint8_t uim_changeUIMPIN::pinLen

#### 8.581.2.5 uint8_t uim_changeUIMPIN::pinVal[255]

## 8.582 uim_depersonalizationInformation Struct Reference

### Data Fields

- uint8_t feature
- uint8_t operation
- uint8_t ckLen
- uint8_t ckVal [255]

### 8.582.1 Detailed Description

This structure contains the Depersonalization Information.

**Parameters**

| | |
|---|---|
| *feature* | • Indicates the personalization feature to de-activate or unblock.<br>– 0 - GW network personalization<br>– 1 - GW network subset personalization<br>– 2 - GW service provider personalization<br>– 3 - GW corporate personalization<br>– 4 - GW UIM personalization<br>– 5 - 1X network type 1 personalization<br>– 6 - 1X network type 2 personalization<br>– 7 - 1X HRPD personalization<br>– 8 - 1X service provider personalization<br>– 9 - 1X corporate personalization<br>– 10 - 1X RUIM personalization |
| *operation* | • Indicates the operation to perform.<br>– 0 - Deactivate personalization.<br>– 1 - Unblock personalization. |
| *ckLen* | • Length of the following elements i.e. control key value. |
| *ckVal[MAX_DESCRIPTION_LENGTH]* | • Control key value.<br>• This value is a sequence of ASCII characters. |

### 8.582.2 Field Documentation

#### 8.582.2.1 uint8_t uim_depersonalizationInformation::ckLen

#### 8.582.2.2 uint8_t uim_depersonalizationInformation::ckVal[255]

#### 8.582.2.3 uint8_t uim_depersonalizationInformation::feature

#### 8.582.2.4 uint8_t uim_depersonalizationInformation::operation

## 8.583 uim_encryptedPIN1 Struct Reference

### Data Fields

- uint8_t pin1Len
- uint8_t pin1Val [255]

### 8.583.1 Detailed Description

This structure contains the encrypted PIN1 Information.

**Parameters**

| | |
|---|---|
| *pin1Len* | • Number of sets of the following elements ie encrypted PIN1 value.<br>• If zero(0), no information follows. |
| *pin1Val* | • Encrypted PIN1 value. |

**Note**

This value is returned only when PIN1 is enabled successfully and the feature is supported.

### 8.583.2 Field Documentation

#### 8.583.2.1 uint8_t uim_encryptedPIN1::pin1Len

#### 8.583.2.2 uint8_t uim_encryptedPIN1::pin1Val[255]

## 8.584 uim_fileAttributes Struct Reference

### Data Fields

- uint16_t fileSize
- uint16_t fileID
- uint8_t fileType
- uint16_t recordSize
- uint16_t recordCount
- uint8_t secRead
- uint16_t secReadMask
- uint8_t secWrite
- uint16_t secWriteMask
- uint8_t secIncrease
- uint16_t secIncreaseMask
- uint8_t secDeactivate
- uint16_t secDeactivateMask
- uint8_t secActivate
- uint16_t secActivateMask
- uint16_t rawLen
- uint8_t rawValue [255]

### 8.584.1 Detailed Description

This structure contains the information about the File Attributes.

**Parameters**

| | |
|---|---|
| *fileSize* | • Indicates the size of the file. |
| *fileID* | • Indicates the ID of the file. |

| | |
|---|---|
| *fileType* | • Indicates the type of the file.<br>– 0 - Transparent<br>– 1 - Cyclic<br>– 2 - Linear fixed<br>– 3 - Dedicated file<br>– 4 - Master file |
| *recordSize* | • Indicates the size of the records.<br>• Only for cyclic and linear fixed files |
| *recordCount* | • Indicates the total no. of the records.<br>• Only for linear fixed files |
| *secRead* | • Read security attributes.<br>– 0 - Always<br>– 1 - Never<br>– 2 - AND condition<br>– 3 - OR condition<br>– 4 - Single condition |
| *secReadMask* | • Mask with read security attributes.<br>• This field is valid only when required by security attributes.<br>– Bit 0 - PIN1<br>– Bit 1 - PIN2<br>– Bit 2 - UPIN<br>– Bit 3 - ADM |
| *secWrite* | • Write security attributes.<br>– 0 - Always<br>– 1 - Never<br>– 2 - AND condition<br>– 3 - OR condition<br>– 4 - Single condition |
| *secWriteMask* | • Mask with write security attributes.<br>• This field is valid only when required by security attributes.<br>– Bit 0 - PIN1<br>– Bit 1 - PIN2<br>– Bit 2 - UPIN<br>– Bit 3 - ADM |

| | |
|---|---|
| *secIncrease* | • Increase security attributes.<br>– 0 - Always<br>– 1 - Never<br>– 2 - AND condition<br>– 3 - OR condition<br>– 4 - Single condition |
| *secIncrease-Mask* | • Mask with increase security attributes.<br>• This field is valid only when required by security attributes.<br>– Bit 0 - PIN1<br>– Bit 1 - PIN2<br>– Bit 2 - UPIN<br>– Bit 3 - ADM |
| *secDeactivate* | • Deactivate security attributes.<br>– 0 - Always<br>– 1 - Never<br>– 2 - AND condition<br>– 3 - OR condition<br>– 4 - Single condition |
| *secDeactivate-Mask* | • Mask with deactivate security attributes.<br>• This field is valid only when required by security attributes.<br>– Bit 0 - PIN1<br>– Bit 1 - PIN2<br>– Bit 2 - UPIN<br>– Bit 3 - ADM |
| *secActivate* | • Activate security attributes.<br>– 0 - Always<br>– 1 - Never<br>– 2 - AND condition<br>– 3 - OR condition<br>– 4 - Single condition |
| *secActivateMask* | • Mask with activate security attributes.<br>• This field is valid only when required by security attributes.<br>– Bit 0 - PIN1<br>– Bit 1 - PIN2<br>– Bit 2 - UPIN<br>– Bit 3 - ADM |
| *rawLen* | • Length of the following elements i.e. raw value. |

| | |
|---|---|
| *rawValue[MAX_-DESCRIPTION-_LENGTH]* | • Raw value of file attributes. |

### 8.584.2 Field Documentation

#### 8.584.2.1 uint16_t uim_fileAttributes::fileID

#### 8.584.2.2 uint16_t uim_fileAttributes::fileSize

#### 8.584.2.3 uint8_t uim_fileAttributes::fileType

#### 8.584.2.4 uint16_t uim_fileAttributes::rawLen

#### 8.584.2.5 uint8_t uim_fileAttributes::rawValue[255]

#### 8.584.2.6 uint16_t uim_fileAttributes::recordCount

#### 8.584.2.7 uint16_t uim_fileAttributes::recordSize

#### 8.584.2.8 uint8_t uim_fileAttributes::secActivate

#### 8.584.2.9 uint16_t uim_fileAttributes::secActivateMask

#### 8.584.2.10 uint8_t uim_fileAttributes::secDeactivate

#### 8.584.2.11 uint16_t uim_fileAttributes::secDeactivateMask

#### 8.584.2.12 uint8_t uim_fileAttributes::secIncrease

#### 8.584.2.13 uint16_t uim_fileAttributes::secIncreaseMask

#### 8.584.2.14 uint8_t uim_fileAttributes::secRead

#### 8.584.2.15 uint16_t uim_fileAttributes::secReadMask

#### 8.584.2.16 uint8_t uim_fileAttributes::secWrite

#### 8.584.2.17 uint16_t uim_fileAttributes::secWriteMask

## 8.585 uim_fileInfo Struct Reference

### Data Fields

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

### 8.585.1 Detailed Description

This structure contains paramaters for file Information

**Parameters**

| | |
|---|---|
| *fileID* | • This is Identifier to SIM files; e.g. in UIM "6F07" is Identifier of IMSI File |
| *pathLen* | • Length of file Path |
| *path* | • Path value. This value must be the complete path of the file, which is a sequence block of 2 bytes (e.g., 0x3F00 0x7FFF) is for LTE (0x3F00 ,0x7F20) is for GSM. |

### 8.585.2 Field Documentation

#### 8.585.2.1 uint16_t uim_fileInfo::fileID

#### 8.585.2.2 uint16_t uim_fileInfo::path[255]

#### 8.585.2.3 uint8_t uim_fileInfo::pathLen

## 8.586 uim_hotSwapStatus Struct Reference

### Data Fields

- uint8_t hotSwapLength
- uint8_t hotSwap [255]

### 8.586.1 Detailed Description

This structure contains Hot Swap Status Information.

**Parameters**

| | |
|---|---|
| *hotSwapLength* | • Number of sets of the following elements. i.e. hot_swap |
| *hotSwap* | • Indicates the status of the hot-swap switch.<br>– 0 - Hot-swap is not supported<br>– 1 - Hot-swap is supported, but the status of the switch is not supported<br>– 2 - Switch indicates that the card is present<br>– 3 - Switch indicates that the card is not present |

### 8.586.2 Field Documentation

#### 8.586.2.1 uint8_t uim_hotSwapStatus::hotSwap[255]

#### 8.586.2.2 uint8_t uim_hotSwapStatus::hotSwapLength

## 8.587 uim_personalizationStatus Struct Reference

### Data Fields

- uint8_t numFeatures
- uint8_t feature [12]
- uint8_t verifyLeft [12]
- uint8_t unblockLeft [12]

### 8.587.1 Detailed Description

This structure contains the information about the card result.

**Parameters**

| | |
|---|---|
| *numFeatures* | • Number of active personalization features. The following block is repeated for each feature. |
| *feature* | • Indicates the personalization feature to deactivate or unblock. Valid values:<br>– 0 - GW network personalization<br>– 1 - GW network subset personalization<br>– 2 - GW service provider personalization<br>– 3 - GW corporate personalization<br>– 4 - GW UIM personalization<br>– 5 - 1X network type 1 personalization<br>– 6 - 1X network type 2 personalization<br>– 7 - 1X HRPD personalization<br>– 8 - 1X service provider personalization<br>– 9 - 1X corporate personalization<br>– 10 - 1X RUIM personalization |
| *verifyLeft* | • Number of the remaining attempts to verify the personalization feature. |
| *unblockLeft* | • Number of the remaining attempts to unblock the personalization feature. |

### 8.587.2 Field Documentation

#### 8.587.2.1 uint8_t uim_personalizationStatus::feature[12]

#### 8.587.2.2 uint8_t uim_personalizationStatus::numFeatures

#### 8.587.2.3 uint8_t uim_personalizationStatus::unblockLeft[12]

#### 8.587.2.4 uint8_t uim_personalizationStatus::verifyLeft[12]

## 8.588 uim_readResult Struct Reference

### Data Fields

- uint16_t contentLen
- uint8_t content [255]

### 8.588.1 Detailed Description

This structure contains the information for write operation.

**Parameters**

| | |
|---|---|
| *contentLen* | • Number of sets of content. |
| *content[255]* | • Read content.<br>• The content is the sequence of bytes as read from the card. |

### 8.588.2 Field Documentation

#### 8.588.2.1 uint8_t uim_readResult::content[255]

#### 8.588.2.2 uint16_t uim_readResult::contentLen

## 8.589 uim_readTransparentInfo Struct Reference

### Data Fields

- uint16_t offset
- uint16_t length

### 8.589.1 Detailed Description

This structure contains the information for read operation.

**Parameters**

| | |
|---|---|
| *offset* | • Offset for the read operation. |
| *length* | • Length of the content to be read.<br>• The value 0 is used to read the complete file. |

### 8.589.2 Field Documentation

#### 8.589.2.1 uint16_t uim_readTransparentInfo::length

#### 8.589.2.2 uint16_t uim_readTransparentInfo::offset

## 8.590 uim_refreshevent Struct Reference

### Data Fields

- uint8_t stage
- uint8_t mode

- uint8_t sessionType
- uint8_t aidLength
- uint8_t aid [255]
- uint16_t numOfFiles
- uim_fileInfo arrfileInfo [255]

### 8.590.1 Detailed Description

This structure contains information of parameters associated with the Refresh Event.

**Parameters**

| | |
|---|---|
| *stage* | • Indicates the stage of the Refresh procedure.<br>– 0 - Waiting for OK to refresh<br>– 1 - Refresh started<br>– 2 - Refresh ended successfully<br>– 3 - Refresh failed |
| *mode* | • Indicates the Refresh mode.<br>– 0 - Reset<br>– 1 - Init<br>– 2 - Init and FCN<br>– 3 - FCN<br>– 4 - Init and Full FCN<br>– 5 - Application reset<br>– 6 - 3G session reset |
| *sessionType* | • Indicates the session type.<br>– 0 - Primary GW provisioning<br>– 1 - Primary 1X provisioning<br>– 2 - Secondary GW provisioning<br>– 3 - Secondary 1X provisioning<br>– 4 - Nonprovisioning on slot 1<br>– 5 - Nonprovisioning on slot 2<br>– 6 - Card on slot 1<br>– 7 - Card on slot 2<br>– 8 - Logical channel on slot 1<br>– 9 - Logical channel on slot 2 |
| *aidLength* | • Number of sets of the following elements<br>– Application Identifier |
| *aid* | • Application identifier value or channel ID. This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases |

| | |
|---|---|
| *numFiles* | • Number of sets of the following elements:<br>– file_id<br>– path_len<br>– path |
| *arrfileInfo* | • Array of file Information struct |

### 8.590.2 Field Documentation

#### 8.590.2.1 uint8_t uim_refreshevent::aid[255]

#### 8.590.2.2 uint8_t uim_refreshevent::aidLength

#### 8.590.2.3 uim_fileInfo uim_refreshevent::arrfileInfo[255]

#### 8.590.2.4 uint8_t uim_refreshevent::mode

#### 8.590.2.5 uint16_t uim_refreshevent::numOfFiles

#### 8.590.2.6 uint8_t uim_refreshevent::sessionType

#### 8.590.2.7 uint8_t uim_refreshevent::stage

## 8.591 uim_registerRefresh Struct Reference

### Data Fields

- uint8_t registerFlag
- uint8_t voteForInit
- uint16_t numFiles
- uim_fileInfo arrfileInfo [255]

### 8.591.1 Detailed Description

This structure contains parameters of refresh Information

**Parameters**

| | |
|---|---|
| *registerFlag* | • Flag that indicates whether to register or deregister for refresh indications. Valid values:<br>– 0 - Deregister<br>– 1 - Register |
| *voteForInit* | • Flag that indicates whether to vote for the init when there is a refresh. Valid values:<br>– 0 - Client does not vote for initialization<br>– 1 - Client votes for initialization |

| | |
|---|---|
| *numFiles* | • Number of sets of the following elements:<br>– file_id<br>– path_len<br>– path |
| *arrfileInfo* | • Array of file Information structure.<br>• See uim_fileInfo for more information |

### 8.591.2 Field Documentation

#### 8.591.2.1 uim_fileInfo uim_registerRefresh::arrfileInfo[255]

#### 8.591.2.2 uint16_t uim_registerRefresh::numFiles

#### 8.591.2.3 uint8_t uim_registerRefresh::registerFlag

#### 8.591.2.4 uint8_t uim_registerRefresh::voteForInit

## 8.592 uim_remainingRetries Struct Reference

### Data Fields

- uint8_t verifyLeft
- uint8_t unblockLeft

### 8.592.1 Detailed Description

This structure contains the information about the retries remaining.

**Parameters**

| | |
|---|---|
| *verifyLeft* | • Number of remaining attempts to verify the PIN.<br>• 0xFF, if unavailable. |
| *unblockLeft* | • Number of remaining attempts to unblock the PIN.<br>• 0xFF, if unavailable. |

**Note**

This value is returned only when the enable/disable operation has failed. This information is not sent for a hidden key PIN type.

### 8.592.2 Field Documentation

#### 8.592.2.1 uint8_t uim_remainingRetries::unblockLeft

#### 8.592.2.2 uint8_t uim_remainingRetries::verifyLeft

## 8.593 uim_sessionInformation Struct Reference

### Data Fields

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

### 8.593.1 Detailed Description

This structure contains the Session Information.

**Parameters**

| | |
|---|---|
| *sessionType* | • Indicates the session type.<br>– 0 - Primary GW provisioning<br>– 1 - Primary 1X provisioning<br>– 2 - Secondary GW provisioning<br>– 3 - Secondary 1X provisioning<br>– 4 - Non-provisioning on slot 1<br>– 5 - Non-provisioning on slot 2<br>– 6 - Card on slot 1<br>– 7 - Card on slot 2<br>– 8 - Logical channel on slot 1<br>– 9 - Logical channel on slot 2 |
| *aidLength* | • Length of the following elements i.e. Application Identifier. |
| *aid* | • Application identifier value or channel ID.<br>• This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases. |

### 8.593.2 Field Documentation

#### 8.593.2.1 uint8_t uim_sessionInformation::aid[255]

#### 8.593.2.2 uint8_t uim_sessionInformation::aidLength

#### 8.593.2.3 uint8_t uim_sessionInformation::sessionType

## 8.594 uim_setPINProtection Struct Reference

### Data Fields

- uint8_t pinID
- uint8_t pinOperation

- uint8_t pinLength
- uint8_t pinValue [255]

### 8.594.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pinID* | • Indicates the PIN ID to be enabled or disabled.<br>– 1 - PIN1 (also called PIN)<br>– 2 - PIN2<br>– 3 - Universal PIN<br>– 4 - Hidden key |
| *pinOperation* | • Indicates whether the PIN is enabled or disabled.<br>– 0 - Disable the PIN<br>– 1 - Enable the PIN |
| *pinLength* | • Length of the following elements i.e. pin value. |
| *pinValue[MAX_-DESCRIPTION-_LENGTH]* | • PIN value.<br>• This value is a sequence of ASCII characters. |

### 8.594.2 Field Documentation

#### 8.594.2.1 uint8_t uim_setPINProtection::pinID

#### 8.594.2.2 uint8_t uim_setPINProtection::pinLength

#### 8.594.2.3 uint8_t uim_setPINProtection::pinOperation

#### 8.594.2.4 uint8_t uim_setPINProtection::pinValue[255]

## 8.595 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]

### 8.595.1 Detailed Description

This structure contains information about the SLOTS present.

**Parameters**

| | |
|---|---|
| *cardState* | • Indicates the state of the card for each slot.<br>– 0 - Absent<br>– 1 - Present<br>– 2 - Error |
| *upinState* | • Indicates the state of UPIN.<br>– 0 - Unknown<br>– 1 - Enabled and not verified<br>– 2 - Enabled and verified<br>– 3 - Disabled<br>– 4 - Blocked<br>– 5 - Permanently blocked<br>– 0xFF - Not Available |
| *upinRetries* | • Indicates the number of retries remaining to verify the UPIN.<br>• If 0xFF, information not available. |
| *upukRetries* | • Indicates the number of retries remaining to unblock the UPIN.<br>• If 0xFF, information not available. |
| *errorState* | • Indicates the reason for the card error, and is valid only when the card state is Error<br>– 0 - Unknown<br>– 1 - Power down<br>– 2 - Poll error<br>– 3 - No ATR received<br>– 4 - Volt mismatch<br>– 5 - Parity error<br>– 6 - Unknown; possibly removed<br>– 7 - Card returned technical problems<br>– 0xFF - Not Available<br>• Other values are possible and reserved for future use.<br>• When an unknown value is received, it is to be handled as "Unknown". |
| *numApp* | • Indicates the number of applications available on the card.<br>• The following block is repeated for each application. i.e. AppStatus.<br>• If zero(0) then no AppStatus information exists. |
| *AppStatus* | • See uim_appStatus for more information. |

### 8.595.2 Field Documentation

#### 8.595.2.1 uim_appStatus uim_slotInfo::AppStatus[10]

#### 8.595.2.2 uint8_t uim_slotInfo::cardState

#### 8.595.2.3 uint8_t uim_slotInfo::errorState

#### 8.595.2.4 uint8_t uim_slotInfo::numApp

#### 8.595.2.5 uint8_t uim_slotInfo::upinRetries

#### 8.595.2.6 uint8_t uim_slotInfo::upinState

#### 8.595.2.7 uint8_t uim_slotInfo::upukRetries

## 8.596 uim_UIMSessionInformation Struct Reference

### Data Fields

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

### 8.596.1 Detailed Description

This structure contains the Session Information.

**Parameters**

| | |
|---|---|
| *sessionType* | • Indicates the session type.<br>– 0 - Primary GW provisioning<br>– 1 - Primary 1X provisioning<br>– 2 - Secondary GW provisioning<br>– 3 - Secondary 1X provisioning<br>– 4 - Non-provisioning on slot 1<br>– 5 - Non-provisioning on slot 2<br>– 6 - Card on slot 1<br>– 7 - Card on slot 2<br>– 8 - Logical channel on slot 1<br>– 9 - Logical channel on slot 2 |
| *aidLength* | • Length of the following elements i.e. Application Identifier. |
| *aid* | • Application identifier value or channel ID.<br>• This value is required for non-provisioning and for logical channel session types. It is ignored in all other cases. |

### 8.596.2 Field Documentation

#### 8.596.2.1 uint8_t uim_UIMSessionInformation::aid[255]

#### 8.596.2.2 uint8_t uim_UIMSessionInformation::aidLength

#### 8.596.2.3 uint8_t uim_UIMSessionInformation::sessionType

## 8.597 uim_unblockUIMPIN Struct Reference

### Data Fields

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

### 8.597.1 Detailed Description

This structure contains the information about the unblock pin parameters.

**Parameters**

| | |
|---|---|
| *pinID* | • Indicates the PIN ID to be changed.<br>– 1 - PIN1 (also called PIN)<br>– 2 - PIN2<br>– 3 - Universal PIN |
| *pukLen* | • Length of the following elements i.e. puk value. |
| *pukVal[UIM_MAX_DESCRIPTION_LENGTH]* | • PIN Unlock Key value.<br>• This value is a sequence of ASCII characters. |
| *newPINLen* | • Length of the following elements i.e. new pin value. |
| *newPINVal[UIM_MAX_DESCRIPTION_LENGTH]* | • New PIN value.<br>• This value is a sequence of ASCII characters. |

### 8.597.2 Field Documentation

#### 8.597.2.1 uint8_t uim_unblockUIMPIN::newPINLen

#### 8.597.2.2 uint8_t uim_unblockUIMPIN::newPINVal[255]

#### 8.597.2.3 uint8_t uim_unblockUIMPIN::pinID

#### 8.597.2.4 uint8_t uim_unblockUIMPIN::pukLen

#### 8.597.2.5 uint8_t uim_unblockUIMPIN::pukVal[255]

## 8.598 uim_verifyUIMPIN Struct Reference

### Data Fields

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

### 8.598.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pinID* | • Indicates the PIN ID to be verified.<br>– 1 - PIN1 (also called PIN)<br>– 2 - PIN2<br>– 3 - Universal PIN<br>– 4 - Hidden key |
| *pinLen* | • Length of the following elements i.e. pin value. |
| *pinVal[MAX_DESCRIPTION_LENGTH]* | • PIN value.<br>• This value is a sequence of ASCII characters. |

### 8.598.2 Field Documentation

#### 8.598.2.1 uint8_t uim_verifyUIMPIN::pinID

#### 8.598.2.2 uint8_t uim_verifyUIMPIN::pinLen

#### 8.598.2.3 uint8_t uim_verifyUIMPIN::pinVal[255]

## 8.599 unpack_audio_SLQSGetAudioPathConfig_t Struct Reference

### Data Fields

- uint8_t * pECMode
- uint8_t * pNSEnable
- uint16_t * pTXGain
- uint16_t * pDTMFTXGain
- uint16_t * pCodecSTGain
- audio_TXPCMIIRFltr * pTXPCMIIRFltr
- audio_RXPCMIIRFltr * pRXPCMIIRFltr
- uint8_t * pMICGainSelect

- uint8_t * pRXAVCAGCSwitch
- uint8_t * pTXAVCSwitch
- audio_RXAGCList * pRXAGCList
- audio_RXAVCList * pRXAVCList
- audio_TXAGCList * pTXAGCList
- swi_uint256_t ParamPresenceMask

### 8.599.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig response parameters.

**Parameters**

| | |
|---|---|
| *pECMode* | [Optional]<br>• AV_EC<br>– 0 - Echo cancellation off<br>– 1 - Handset mode<br>– 2 - Headset mode<br>– 3 - Car kit mode<br>– 4 - Speaker Mode<br>• Bit to check in ParamPresenceMask - **33** |
| *pNSEnable* | [Optional]<br>• AV_NS<br>– 0 - Noise suppression off<br>– 1 - Noise suppression on<br>• Bit to check in ParamPresenceMask - **34** |
| *pTXGain* | [Optional]<br>• AV_TXVOL<br>– 0x0000 - 0xffff<br>• Bit to check in ParamPresenceMask - **35** |
| *pDTMFTXGain* | [Optional]<br>• AV_DTMFTXG<br>– 0x0000 - 0xffff<br>• Bit to check in ParamPresenceMask - **36** |
| *pCodecSTGain* | [Optional]<br>• AV_CODECSTG<br>– 0x0000 - 0xffff<br>• Bit to check in ParamPresenceMask - **37** |
| *pTXPCMIIRFltr* | [Optional]<br>• See audio_TXPCMIIRFltr for more information<br>• Bit to check in ParamPresenceMask - **38** |
| *pRXPCMIIRFltr* | [Optional]<br>• See audio_RXPCMIIRFltr for more information<br>• Bit to check in ParamPresenceMask - **39** |

| | |
|---|---|
| *pMICGainSelect* | [Optional]<br>• AV_MICGAIN<br>• Bit to check in ParamPresenceMask - **40** |
| *pRXAVCAGC-Switch* | [Optional]<br>• RX AVC/AGC Switch<br>• Bit to check in ParamPresenceMask - **41** |
| *pTXAVCSwitch* | [Optional]<br>• TX AVC Switch<br>• Bit to check in ParamPresenceMask - **42** |
| *pRXAGCList* | [Optional]<br>• See audio_RXAGCList for more information<br>• Bit to check in ParamPresenceMask - **43** |
| *pRXAVCList* | [Optional]<br>• See audio_RXAVCList for more information<br>• Bit to check in ParamPresenceMask - **44** |
| *pTXAGCList* | [Optional]<br>• See audio_TXAGCList for more information<br>• Bit to check in ParamPresenceMask - **45** |

### 8.599.2 Field Documentation

#### 8.599.2.1 swi_uint256_t unpack_audio_SLQSGetAudioPathConfig_t::ParamPresenceMask

#### 8.599.2.2 uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pCodecSTGain

#### 8.599.2.3 uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pDTMFTXGain

#### 8.599.2.4 uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pECMode

#### 8.599.2.5 uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pMICGainSelect

#### 8.599.2.6 uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pNSEnable

#### 8.599.2.7 audio_RXAGCList* unpack_audio_SLQSGetAudioPathConfig_t::pRXAGCList

#### 8.599.2.8 uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pRXAVCAGCSwitch

#### 8.599.2.9 audio_RXAVCList* unpack_audio_SLQSGetAudioPathConfig_t::pRXAVCList

#### 8.599.2.10 audio_RXPCMIIRFltr* unpack_audio_SLQSGetAudioPathConfig_t::pRXPCMIIRFltr

#### 8.599.2.11 audio_TXAGCList* unpack_audio_SLQSGetAudioPathConfig_t::pTXAGCList

#### 8.599.2.12 uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pTXAVCSwitch

#### 8.599.2.13 uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pTXGain

#### 8.599.2.14 audio_TXPCMIIRFltr* unpack_audio_SLQSGetAudioPathConfig_t::pTXPCMIIRFltr

## 8.600 unpack_audio_SLQSGetAudioProfile_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t EarMute
- uint8_t MicMute
- uint8_t Volume
- swi_uint256_t ParamPresenceMask

### 8.600.1 Detailed Description

This structure contains the unpack parameters to Get Audio Profile.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>– 0 - Handset<br>– 1 - Headset<br>– 2 - Car Kit<br>– 3 - Speaker phone<br>– 4 - Auxiliary<br>– 5 - TTY<br>– 6 - Auxiliary external PCM<br>– 7 - Primary external PCM<br>– 8 - External slave PCM<br>– 9 - I2S<br>• Bit to check in ParamPresenceMask - **16** |
| *EarMute* | • Ear Mute Setting<br>– 0 - unmuted<br>– 1 - muted<br>• Bit to check in ParamPresenceMask - **17** |
| *MicMute* | • MIC Mute Setting<br>– 0 - unmuted<br>– 1 - muted<br>• Bit to check in ParamPresenceMask - **18** |
| *Volume* | • Audio Volume Level<br>– 0 to 7<br>• Bit to check in ParamPresenceMask - **20** |

### 8.600.2 Field Documentation

#### 8.600.2.1 uint8_t unpack_audio_SLQSGetAudioProfile_t::EarMute

#### 8.600.2.2 uint8_t unpack_audio_SLQSGetAudioProfile_t::MicMute

#### 8.600.2.3 swi_uint256_t unpack_audio_SLQSGetAudioProfile_t::ParamPresenceMask

#### 8.600.2.4 uint8_t unpack_audio_SLQSGetAudioProfile_t::Profile

#### 8.600.2.5 uint8_t unpack_audio_SLQSGetAudioProfile_t::Volume

## 8.601 unpack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference

### Data Fields

- uint16_t ResCode
- swi_uint256_t ParamPresenceMask

### 8.601.1 Detailed Description

This structure contains the unpack parameters for SLQSGetAudioVolTLBConfig.

**Parameters**

| | |
|---|---|
| *ResCode* | • Result of requested item<br>• Bit to check in ParamPresenceMask - **33** |

### 8.601.2 Field Documentation

#### 8.601.2.1 swi_uint256_t unpack_audio_SLQSGetAudioVolTLBConfig_t::ParamPresenceMask

#### 8.601.2.2 uint16_t unpack_audio_SLQSGetAudioVolTLBConfig_t::ResCode

## 8.602 unpack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference

### Data Fields

- uint16_t ResCode
- swi_uint256_t ParamPresenceMask

### 8.602.1 Detailed Description

This structure contains the unpack parameters for SLQSSetAudioVolTLBConfig.

**Parameters**

| | |
|---|---|
| *ResCode* | • Result of requested item<br>• Bit to check in ParamPresenceMask - **33** |

### 8.602.2 Field Documentation

#### 8.602.2.1 swi_uint256_t unpack_audio_SLQSSetAudioVolTLBConfig_t::ParamPresenceMask

#### 8.602.2.2 uint16_t unpack_audio_SLQSSetAudioVolTLBConfig_t::ResCode

## 8.603 unpack_cat_SetCatEventCallback_ind_t Struct Reference

### Data Fields

- uint8_t event_Index
- struct cat_commonEventTlv CCETlv [11]
- swi_uint256_t ParamPresenceMask

### 8.603.1 Detailed Description

Structure used to store all CAT Notification Parameters.

**Parameters**

| | |
|---|---|
| *event_Index* | - Event Index |
| *CCETlv* | - CAT common event Tlv cat_commonEventTlv for more info.<br>• [0] - Bit to check in ParamPresenceMask for Display Text - **16**<br>• [1] - Bit to check in ParamPresenceMask for Get In-Key - **17**<br>• [2] - Bit to check in ParamPresenceMask for Get Input - **18**<br>• [3] - Bit to check in ParamPresenceMask for Setup Menu - **19**<br>• [4] - Bit to check in ParamPresenceMask for Select Item - **20**<br>• [5] - Bit to check in ParamPresenceMask for Alpha Identifier - **21**<br>• [6] - Bit to check in ParamPresenceMask for Setup Event List - **22**<br>• [7] - Bit to check in ParamPresenceMask for Setup Idle Mode Text - **23**<br>• [8] - Bit to check in ParamPresenceMask for Language Notification - **24**<br>• [9] - Bit to check in ParamPresenceMask for Refresh - **25**<br>• [10] - Bit to check in ParamPresenceMask for End Proactive Session - **26** |

### 8.603.2 Field Documentation

#### 8.603.2.1 struct cat_commonEventTlv unpack_cat_SetCatEventCallback_ind_t::CCETlv[11]

#### 8.603.2.2 uint8_t unpack_cat_SetCatEventCallback_ind_t::event_Index

#### 8.603.2.3 swi_uint256_t unpack_cat_SetCatEventCallback_ind_t::ParamPresenceMask

## 8.604 unpack_cat_SetCATEventCallback_t Struct Reference

### Data Fields

- uint32_t errorMask
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.604.1 Detailed Description

This structure contains sEnables/disables the CAT event callback unpack variable.

**Parameters**

| | |
|---|---|
| *errorMask* | • error bitmask. Each bit set indicates the proactive command that caused the error<br>– 0x00000001 - Display Text<br>– 0x00000002 - Get In-Key<br>– 0x00000004 - Get Input<br>– 0x00000008 - Setup Menu<br>– 0x00000010 - Select Item<br>– 0x00000020 - Send SMS - Alpha Identifier<br>– 0x00000040 - Setup Event: User Activity<br>– 0x00000080 - Setup Event: Idle Screen Notify<br>– 0x00000100 - Setup Event: Language Sel Notify<br>– 0x00000200 - Setup Idle Mode Text<br>– 0x00000400 - Language Notification<br>– 0x00000800 - Refresh<br>– 0x00001000 - End Proactive Session<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult* | • Unpack Result |

### 8.604.2 Field Documentation

#### 8.604.2.1 uint32_t unpack_cat_SetCATEventCallback_t::errorMask

#### 8.604.2.2 swi_uint256_t unpack_cat_SetCATEventCallback_t::ParamPresenceMask

#### 8.604.2.3 uint16_t unpack_cat_SetCATEventCallback_t::Tlvresult

## 8.605 unpack_dms_GetActivationState_t Struct Reference

### Data Fields

- uint8_t state
- swi_uint256_t ParamPresenceMask

### 8.605.1 Detailed Description

This structure contains unpack_dms_GetActivationState parameters.

**Parameters**

| | |
|---|---|
| *state[OUT]* | • Service Activation Code<br>0 - Service not activated<br>1 - Service activated<br>2 - Activation connecting<br>3 - Activation connected<br>4 - OTASP security authenticated<br>5 - OTASP NAM downloaded<br>6 - OTASP MDN downloaded<br>7 - OTASP IMSI downloaded<br>8 - OTASP PRL downloaded<br>9 - OTASP SPC downloaded<br>10 - OTASP settings committed<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.605.2 Field Documentation

#### 8.605.2.1 swi_uint256_t unpack_dms_GetActivationState_t::ParamPresenceMask

#### 8.605.2.2 uint8_t unpack_dms_GetActivationState_t::state

## 8.606 unpack_dms_GetBandCapability_t Struct Reference

### Data Fields

- uint32_t BandCapability
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.606.1 Detailed Description

Structure to store the band capability of the device.

**Parameters**

| | |
|---|---|
| *BandCapability* | Bitmask of bands supported by the device<br>• Bit 0 - Band class 0, A-system<br>• Bit 1 - Band class 0, B-system<br>• Bit 2 - Band class 1, all blocks<br>• Bit 3 - Band class 2<br>• Bit 4 - Band class 3, A-system<br>• Bit 5 - Band class 4, all blocks<br>• Bit 6 - Band class 5, all blocks<br>• Bit 7 - GSM DCS band (1800)<br>• Bit 8 - GSM Extended GSM (E-GSM) band (900)<br>• Bit 9 - GSM Primary GSM (P-GSM) band (900)<br>• Bit 10 - Band class 6<br>• Bit 11 - Band class 7<br>• Bit 12 - Band class 8<br>• Bit 13 - Band class 9<br>• Bit 14 - Band class 10<br>• Bit 15 - Band class 11<br>• Bit 16 - GSM 450 band<br>• Bit 17 - GSM 480 band<br>• Bit 18 - GSM 750 band<br>• Bit 19 - GSM 850 band<br>• Bit 20 - GSM railways GSM band (900)<br>• Bit 21 - GSM PCS band (1900)<br>• Bit 22 - WCDMA (Europe, Japan, and China) 2100 band<br>• Bit 23 - WCDMA US PCS 1900 band<br>• Bit 24 - WCDMA (Europe and China) DCS 1800 band<br>• Bit 25 - WCDMA US 1700 band<br>• Bit 26 - WCDMA US 850 band<br>• Bit 27 - WCDMA Japan 800 band<br>• Bit 28 - Band class 12<br>• Bit 29 - Band class 14<br>• Bit 30 - Reserved<br>• Bit 31 - Band class 15<br>• Bits 32 through 47 - Reserved<br>• Bit 48 - WCDMA Europe 2600 band<br>• Bit 49 - WCDMA Europe and Japan 900 band<br>• Bit 50 - WCDMA Japan 1700 band<br>• Bits 51 through 55 - Reserved<br>• Bit 56 - Band class 16<br>• Bit 57 - Band class 17<br>• Bit 58 - Band class 18<br>• Bit 59 - Band class 19<br>• Bit 60 - WCDMA Japan 850 band<br>• Bit 61 - WCDMA 1500 band<br>• Bits 62 and 63 - Reserved |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.606.2 Field Documentation

#### 8.606.2.1 uint32_t unpack_dms_GetBandCapability_t::BandCapability

#### 8.606.2.2 swi_uint256_t unpack_dms_GetBandCapability_t::ParamPresenceMask

#### 8.606.2.3 uint16_t unpack_dms_GetBandCapability_t::Tlvresult

## 8.607 unpack_dms_GetCrashAction_t Struct Reference

### Data Fields

- uint8_t DevCrashState
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.607.1 Detailed Description

This structure is used to store the Crash State from the device.

**Parameters**

| | |
|---|---|
| *DevCrashState* | • Device Crash State<br>• Values:<br>– 0 - USB Memory Download Modem will reset after a crash and will stay in USB download mode with only ttyUSB0 enumerated. ramdump tool is to be used to recover the crash dump. Modem needs to be reset again to come back in ONLINE mode.<br>– 1 - Reset Modem will reset and come back in ONLINE mode. Minimal crash data will be available and can be extracted with at!gcdump? AT command or SLQSSwiGetCrashInfo() SDK API<br>– 2 - No action<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.607.2 Field Documentation

#### 8.607.2.1 uint8_t unpack_dms_GetCrashAction_t::DevCrashState

#### 8.607.2.2 swi_uint256_t unpack_dms_GetCrashAction_t::ParamPresenceMask

#### 8.607.2.3 uint16_t unpack_dms_GetCrashAction_t::Tlvresult

## 8.608 unpack_dms_GetCustFeature_t Struct Reference

### Data Fields

- uint32_t GpsEnable
- uint8_t DisableIMSI
- uint16_t IPFamSupport
- uint8_t RMAutoConnect
- uint8_t GPSSel
- uint8_t SMSSupport
- uint8_t IsVoiceEnabled
- uint8_t DHCPRelayEnabled
- uint8_t GPSLPM
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.608.1 Detailed Description

This structure contains current settings of custom features

**Parameters**

| | |
|---|---|
| *GpsEnable* | • describes if GPS is enabled or disabled<br>• values:<br>– 0x00 - GPS is disabled<br>– 0x01 - GPS is enabled<br>• Bit to check in ParamPresenceMask - **16** |
| *DisableIMSI* | • optional 1 byte parameter<br>• describes if IMSI display is enabled or disabled<br>• values:<br>– 0x00 - Allow display of IMSI<br>– 0x01 - Do not display IMSI<br>• Bit to check in ParamPresenceMask - **17** |
| *IPFamSupport* | • optional 2 byte BitMask<br>• bitmask representing the IP families supported<br>• values:<br>– 0x01 - IPv4<br>– 0x02 - IPv6<br>– 0x04 - IPv4v6<br>• Bit to check in ParamPresenceMask - **18** |

| | |
|---|---|
| *RMAutoConnect* | • optional 1 byte parameter<br>• QMI Mode RM Net Auto Connect Support<br>• values:<br>– 0x00 - Not Supported<br>– 0x01 - Supported<br>• Bit to check in ParamPresenceMask - **19** |
| *GPSSel* | • optional 1 byte parameter<br>• GPS Antenna Select<br>• values:<br>– 0x00 - Dedicated GPS Port<br>– 0x01 - GPS Rx over AUX Port<br>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied<br>• Bit to check in ParamPresenceMask - **20** |
| *SMSSupport* | • optional 1 byte parameter<br>• SMS support<br>• values:<br>– 0x00 - Not supported<br>– 0x01 - supported<br>• Bit to check in ParamPresenceMask - **21**<br>• Used to determine whether or not to hide SMS from user |
| *IsVoiceEnabled* | • optional 1 byte parameter<br>• Voice support<br>• values:<br>– 0x00 - Enable voice on both AT and QMI interface (default)<br>– 0x01 - Reserved<br>– 0x02 - Disable voice on both AT and QMI interface<br>• Bit to check in ParamPresenceMask - **22** |
| *DHCPRelay-Enabled* | • optional 1 byte parameter<br>• DHCP Relay support<br>• values:<br>– 0x00 - Disable DHCP relay<br>– 0x01 - Enable DHCP relay<br>• Bit to check in ParamPresenceMask - **23** |
| *GPSLPM* | • optional 1 byte parameter<br>• GPSLPM support<br>• values:<br>– 0x00 - Enable GPS in Low Power Mode<br>– 0x01 - Disable GPS in Low Power Mode<br>• Bit to check in ParamPresenceMask - **24** |

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.608.2 Field Documentation

#### 8.608.2.1 uint8_t unpack_dms_GetCustFeature_t::DHCPRelayEnabled

#### 8.608.2.2 uint8_t unpack_dms_GetCustFeature_t::DisableIMSI

#### 8.608.2.3 uint32_t unpack_dms_GetCustFeature_t::GpsEnable

#### 8.608.2.4 uint8_t unpack_dms_GetCustFeature_t::GPSLPM

#### 8.608.2.5 uint8_t unpack_dms_GetCustFeature_t::GPSSel

#### 8.608.2.6 uint16_t unpack_dms_GetCustFeature_t::IPFamSupport

#### 8.608.2.7 uint8_t unpack_dms_GetCustFeature_t::IsVoiceEnabled

#### 8.608.2.8 swi_uint256_t unpack_dms_GetCustFeature_t::ParamPresenceMask

#### 8.608.2.9 uint8_t unpack_dms_GetCustFeature_t::RMAutoConnect

#### 8.608.2.10 uint8_t unpack_dms_GetCustFeature_t::SMSSupport

#### 8.608.2.11 uint16_t unpack_dms_GetCustFeature_t::Tlvresult

## 8.609 unpack_dms_GetCustFeaturesV2_t Struct Reference

### Data Fields

- DMSgetCustomFeatureV2 GetCustomFeatureV2
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.609.1 Detailed Description

This structure contains customization settings set to modem unpack

**Parameters**

| | |
|---|---|
| *GetCustom-FeatureV2* | • See DMSgetCustomFeatureV2 for more information<br>• pCustSettingInfo<br>  – Bit to check in ParamPresenceMask - **32**<br>• pCustSettingList<br>  – Bit to check in ParamPresenceMask - **33** |

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.609.2 Field Documentation

#### 8.609.2.1 DMSgetCustomFeatureV2 unpack_dms_GetCustFeaturesV2_t::GetCustomFeatureV2

#### 8.609.2.2 swi_uint256_t unpack_dms_GetCustFeaturesV2_t::ParamPresenceMask

#### 8.609.2.3 uint16_t unpack_dms_GetCustFeaturesV2_t::Tlvresult

## 8.610 unpack_dms_GetDeviceCap_t Struct Reference

### Data Fields

- uint32_t MaxTXChannelRate
- uint32_t MaxRXChannelRate
- uint32_t DataServiceCapability
- uint32_t SimCapability
- uint32_t RadioIfacesSize
- uint8_t RadioIfaces [64]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.610.1 Detailed Description

Structure to store the device capabilities

**Parameters**

| | |
|---|---|
| *MaxTXChannel-Rate* | • Maximum transmission rate (in bps) supported by the device<br>• In multi-technology devices, this value will be the greatest rate among all supported technologies<br>• Bit to check in ParamPresenceMask - **1** |
| *MaxRXChannel-Rate* | • Maximum reception rate (in bps) supported by the device<br>• In multi-technology devices, this value will be the greatest rate among all supported technologies<br>• Bit to check in ParamPresenceMask - **1** |
| *DataService-Capability* | • CS/PS data service capability<br>– 0 - No data services supported<br>– 1 - Only Circuit Switched (CS) services supported<br>– 2 - Only Packet Switched (PS) services supported<br>– 3 - Simultaneous CS and PS<br>– 4 - Non-simultaneous CS and PS<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *SimCapability* | Device SIM capability<br>• 0 - SIM not supported<br>• 1 - SIM supported |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *RadioIfacesSize* | • Upon input, the maximum number of elements that the radio interface array can contain<br>• Upon successful output, actual number of elements in the radio interface array<br>• Bit to check in ParamPresenceMask - **1** |
| *RadioIfaces* | • Radio interface array. This is a structure of array containing the elements below. uint8_t RadioIfaces<br>– See qaGobiApiTableRadioInterfaces.h for Radio Interfaces<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.610.2 Field Documentation

#### 8.610.2.1 uint32_t unpack_dms_GetDeviceCap_t::DataServiceCapability

#### 8.610.2.2 uint32_t unpack_dms_GetDeviceCap_t::MaxRXChannelRate

#### 8.610.2.3 uint32_t unpack_dms_GetDeviceCap_t::MaxTXChannelRate

#### 8.610.2.4 swi_uint256_t unpack_dms_GetDeviceCap_t::ParamPresenceMask

#### 8.610.2.5 uint8_t unpack_dms_GetDeviceCap_t::RadioIfaces[64]

#### 8.610.2.6 uint32_t unpack_dms_GetDeviceCap_t::RadioIfacesSize

#### 8.610.2.7 uint32_t unpack_dms_GetDeviceCap_t::SimCapability

#### 8.610.2.8 uint16_t unpack_dms_GetDeviceCap_t::Tlvresult

## 8.611 unpack_dms_GetDeviceCapabilities_t Struct Reference

### Data Fields

- uint32_t maxTxChannelRate
- uint32_t maxRxChannelRate
- uint32_t dataServiceCaCapability

- uint32_t simCapability
- uint32_t radioIfacesSize
- uint8_t RadioIfaces [255]
- swi_uint256_t ParamPresenceMask

### 8.611.1 Detailed Description

Gets the device capabilities structure

**Parameters**

| | |
|---|---|
| *maxTxChannelRate* | • Maximum transmission rate (in bps) supported by the device<br>• In multi-technology devices, this value will be the greatest rate among all supported technologies<br>• Bit to check in ParamPresenceMask - **1** |
| *maxRxChannelRate* | • Maximum reception rate (in bps) supported by the device<br>• In multi-technology devices, this value will be the greatest rate among all supported technologies<br>• Bit to check in ParamPresenceMask - **1** |
| *dataServiceCaCapability* | • CS/PS data service capability<br>– 0 - No data services supported<br>– 1 - Only Circuit Switched (CS) services supported<br>– 2 - Only Packet Switched (PS) services supported<br>– 3 - Simultaneous CS and PS<br>– 4 - Non-simultaneous CS and PS<br>• Bit to check in ParamPresenceMask - **1** |
| *simCapability* | • Device SIM capability<br>– 0 - SIM not supported<br>– 1 - SIM supported<br>• Bit to check in ParamPresenceMask - **1** |
| *radioIfacesSize* | • Upon input, the maximum number of elements that the radio interface array can contain<br>• Upon successful output, actual number of elements in the radio interface array<br>• Bit to check in ParamPresenceMask - **1** |
| *RadioIfaces[OUT]* | • Radio interface array. This is a structure of array containing the elements below.<br>– See qaGobiApiTableRadioInterfaces.h for Radio Interfaces<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.611.2 Field Documentation

#### 8.611.2.1 uint32_t unpack_dms_GetDeviceCapabilities_t::dataServiceCaCapability

#### 8.611.2.2 uint32_t unpack_dms_GetDeviceCapabilities_t::maxRxChannelRate

#### 8.611.2.3 uint32_t unpack_dms_GetDeviceCapabilities_t::maxTxChannelRate

#### 8.611.2.4 swi_uint256_t unpack_dms_GetDeviceCapabilities_t::ParamPresenceMask

#### 8.611.2.5 uint8_t unpack_dms_GetDeviceCapabilities_t::RadioIfaces[255]

#### 8.611.2.6 uint32_t unpack_dms_GetDeviceCapabilities_t::radioIfacesSize

#### 8.611.2.7 uint32_t unpack_dms_GetDeviceCapabilities_t::simCapability

## 8.612 unpack_dms_GetDeviceHardwareRev_t Struct Reference

### Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.612.1 Detailed Description

Structure to store the hardware revision of the device

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain<br>• Bit to check in ParamPresenceMask - **1** |
| *String* | • NULL terminated Hardware Revision string<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.612.2 Field Documentation

#### 8.612.2.1 swi_uint256_t unpack_dms_GetDeviceHardwareRev_t::ParamPresenceMask

#### 8.612.2.2 char unpack_dms_GetDeviceHardwareRev_t::String[255]

#### 8.612.2.3 uint8_t unpack_dms_GetDeviceHardwareRev_t::stringSize

#### 8.612.2.4 uint16_t unpack_dms_GetDeviceHardwareRev_t::Tlvresult

## 8.613 unpack_dms_GetDeviceMfr_t Struct Reference

### Data Fields

- uint8_t stringSize
- char String [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.613.1 Detailed Description

This structure is used to store unpack_dms_GetDeviceMfr parameters

**Parameters**

| | |
|---|---|
| *stringSize* | • The maximum number of characters (including NULL terminator) that the string array can contain |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *String* | • NULL terminated Device Manufacture string<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.613.2 Field Documentation

#### 8.613.2.1 swi_uint256_t unpack_dms_GetDeviceMfr_t::ParamPresenceMask

#### 8.613.2.2 char unpack_dms_GetDeviceMfr_t::String[255]

#### 8.613.2.3 uint8_t unpack_dms_GetDeviceMfr_t::stringSize

#### 8.613.2.4 uint16_t unpack_dms_GetDeviceMfr_t::Tlvresult

## 8.614 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
- swi_uint256_t ParamPresenceMask

### 8.614.1 Detailed Description

This structure used to store unpack_dms_GetDeviceSerialNumbers parameters

**Parameters**

| | |
|---|---|
| *esnSize* | • The maximum number of characters (including NULL terminator) that the ESN string array can contain<br>• Bit to check in ParamPresenceMask - **16** |
| *ESNString* | • NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **16** |
| *imeiSize* | • The maximum number of characters (including NULL terminator) that the IMEI string array can contain<br>• Bit to check in ParamPresenceMask - **17** |
| *IMEIString* | • NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **17** |
| *meidSize* | • The maximum number of characters (including NULL terminator) that the MEID string array can contain<br>• Bit to check in ParamPresenceMask - **18** |
| *MEIDString* | • NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **18** |
| *imeiSvnSize* | • The maximum number of characters (including NULL terminator) that the IMEI SVN string array can contain.<br>• Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *ImeiSvnString* | • NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **19** |
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.614.2 Field Documentation

#### 8.614.2.1 uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize

#### 8.614.2.2 char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]

#### 8.614.2.3 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize

#### 8.614.2.4 char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]

#### 8.614.2.5 uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize

#### 8.614.2.6 char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]

#### 8.614.2.7 uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize

#### 8.614.2.8 char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]

#### 8.614.2.9 swi_uint256_t unpack_dms_GetDeviceSerialNumbers_t::ParamPresenceMask

#### 8.614.2.10 uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult

## 8.615 unpack_dms_GetFirmwareInfo_t Struct Reference

### Data Fields

- char modelid_str [20]
- char bootversion_str [85]
- char appversion_str [85]
- char sku_str [15]
- char packageid_str [85]
- char carrier_str [20]
- char priversion_str [16]
- char cur_carr_name [17]
- char cur_carr_rev [13]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.615.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareInfo parameters

**Parameters**

| | |
|---|---|
| *modelid_str* | - Mode ID String.<br>• NULL-terminated Mode ID String.<br>• Bit to check in ParamPresenceMask - **16** |
| *bootversion_str* | - Boot Version.<br>• NULL-terminated Boot Version String.<br>• Bit to check in ParamPresenceMask - **17** |
| *appversion_str* | - Application Version String.<br>• NULL-terminated Application Version String.<br>• Bit to check in ParamPresenceMask - **18** |
| *sku_str* | - SKU String.<br>• NULL-terminated SKU String.<br>• Bit to check in ParamPresenceMask - **19** |
| *packageid_str* | - Package ID String.<br>• NULL-terminated Package ID String.<br>• deprecated on EM/MC74xx(9x30) devices<br>• Bit to check in ParamPresenceMask - **20** |
| *carrier_str* | - Carrier String.<br>• NULL-terminated Carrier String.<br>• Bit to check in ParamPresenceMask - **21** |
| *priversion_str* | - PRI Version String.<br>• NULL-terminated PRI Version String.<br>• Bit to check in ParamPresenceMask - **22** |
| *cur_carr_name* | - Current Carrier Name String.<br>• NULL-terminated Current Carrier Name String.<br>• Bit to check in ParamPresenceMask - **23** |
| *cur_carr_rev* | - Current Carrier Revision String.<br>• NULL-terminated Current Carrier Revision String.<br>• Bit to check in ParamPresenceMask - **24** |
| *Tlvresult* | - unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.615.2 Field Documentation

#### 8.615.2.1 char unpack_dms_GetFirmwareInfo_t::appversion_str[85]

#### 8.615.2.2 char unpack_dms_GetFirmwareInfo_t::bootversion_str[85]

#### 8.615.2.3 char unpack_dms_GetFirmwareInfo_t::carrier_str[20]

#### 8.615.2.4 char unpack_dms_GetFirmwareInfo_t::cur_carr_name[17]

#### 8.615.2.5 char unpack_dms_GetFirmwareInfo_t::cur_carr_rev[13]

#### 8.615.2.6 char unpack_dms_GetFirmwareInfo_t::modelid_str[20]

#### 8.615.2.7 char unpack_dms_GetFirmwareInfo_t::packageid_str[85]

#### 8.615.2.8 swi_uint256_t unpack_dms_GetFirmwareInfo_t::ParamPresenceMask

#### 8.615.2.9 char unpack_dms_GetFirmwareInfo_t::priversion_str[16]

#### 8.615.2.10 char unpack_dms_GetFirmwareInfo_t::sku_str[15]

#### 8.615.2.11 uint16_t unpack_dms_GetFirmwareInfo_t::Tlvresult

## 8.616 unpack_dms_GetFirmwareRevision_t Struct Reference

### Data Fields

- uint8_t amssSize
- char AMSSString [255]
- char PRIString [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.616.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareRevision parameters

**Parameters**

| | |
|---|---|
| *amssSize* | • Maximum number of characters (including NULL terminator) that the AMSS string array can contain.<br>• Bit to check in ParamPresenceMask - **1** |
| *AMSSString* | • NULL-terminated AMSS revision string.<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *PRIString[OUT]* | • NULL-terminated PRI revision string.<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.616.2 Field Documentation

#### 8.616.2.1 uint8_t unpack_dms_GetFirmwareRevision_t::amssSize

#### 8.616.2.2 char unpack_dms_GetFirmwareRevision_t::AMSSString[255]

#### 8.616.2.3 swi_uint256_t unpack_dms_GetFirmwareRevision_t::ParamPresenceMask

#### 8.616.2.4 char unpack_dms_GetFirmwareRevision_t::PRIString[255]

#### 8.616.2.5 uint16_t unpack_dms_GetFirmwareRevision_t::Tlvresult

## 8.617 unpack_dms_GetFirmwareRevisions_t Struct Reference

### Data Fields

- uint8_t amssSize
- char AMSSString [255]
- uint8_t bootSize
- char BootString [255]
- uint8_t priSize
- char PRIString [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.617.1 Detailed Description

This structure used to store unpack_dms_GetFirmwareRevisions parameters

**Parameters**

| | |
|---|---|
| *amssSize* | • Maximum number of characters (including NULL terminator) that the AMSS string array can contain<br>• Bit to check in ParamPresenceMask - **1** |
| *AMSSString* | • NULL-terminated AMSS revision string<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *bootSize* | • Maximum number of characters (including NULL terminator) that the boot string array can contain<br>• Bit to check in ParamPresenceMask - **16** |
| *BootString* | • NULL-terminated boot code revision string<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **16** |
| *priSize* | • Maximum number of characters (including NULL terminator) that the PRI string array can contain<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *PRIString* | • NULL-terminated PRI revision string<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | • unpack Tlv Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.617.2 Field Documentation

#### 8.617.2.1 uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize

#### 8.617.2.2 char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]

#### 8.617.2.3 uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize

#### 8.617.2.4 char unpack_dms_GetFirmwareRevisions_t::BootString[255]

#### 8.617.2.5 swi_uint256_t unpack_dms_GetFirmwareRevisions_t::ParamPresenceMask

#### 8.617.2.6 uint8_t unpack_dms_GetFirmwareRevisions_t::priSize

#### 8.617.2.7 char unpack_dms_GetFirmwareRevisions_t::PRIString[255]

#### 8.617.2.8 uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult

## 8.618 unpack_dms_GetFSN_t Struct Reference

### Data Fields

- char String [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.618.1 Detailed Description

This structure used to store Factory Sequence Number parameter

**Parameters**

| | |
|---|---|
| *String* | • Facorty Sequence Number<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack Result |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.618.2 Field Documentation

#### 8.618.2.1 swi_uint256_t unpack_dms_GetFSN_t::ParamPresenceMask

#### 8.618.2.2 char unpack_dms_GetFSN_t::String[255]

#### 8.618.2.3 uint16_t unpack_dms_GetFSN_t::Tlvresult

## 8.619 unpack_dms_GetHardwareRevision_t Struct Reference

### Data Fields

- char hwVer [255]
- swi_uint256_t ParamPresenceMask

### 8.619.1 Detailed Description

This structure used to store unpack_dms_GetHardwareRevision parameters

**Parameters**

| | |
|---|---|
| *hwVer* | - Hardware version<br>• NULL-terminated string<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.619.2 Field Documentation

#### 8.619.2.1 char unpack_dms_GetHardwareRevision_t::hwVer[255]

#### 8.619.2.2 swi_uint256_t unpack_dms_GetHardwareRevision_t::ParamPresenceMask

## 8.620 unpack_dms_GetIMSI_t Struct Reference

### Data Fields

- char imsi [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.620.1 Detailed Description

This structure used to store unpack_dms_GetIMSI parameters

**Parameters**

| | |
|---|---|
| *imsi* | - IMSI no.<br>• NULL-terminated String.<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | - unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.620.2 Field Documentation

#### 8.620.2.1 char unpack_dms_GetIMSI_t::imsi[255]

#### 8.620.2.2 swi_uint256_t unpack_dms_GetIMSI_t::ParamPresenceMask

#### 8.620.2.3 uint16_t unpack_dms_GetIMSI_t::Tlvresult

## 8.621 unpack_dms_GetManufacturer_t Struct Reference

### Data Fields

- char manufacturer [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.621.1 Detailed Description

This structure is used to store device manufacturer information.

**Parameters**

| | |
|---|---|
| *manufacturer[OUT]* | • NULL terminated string<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.621.2 Field Documentation

#### 8.621.2.1 char unpack_dms_GetManufacturer_t::manufacturer[255]

#### 8.621.2.2 swi_uint256_t unpack_dms_GetManufacturer_t::ParamPresenceMask

#### 8.621.2.3 uint16_t unpack_dms_GetManufacturer_t::Tlvresult

## 8.622 unpack_dms_GetModelID_t Struct Reference

### Data Fields

- char modelid [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.622.1 Detailed Description

This structure used to store unpack_dms_GetModelID parameters

**Parameters**

| | |
|---|---|
| *modelid* | - Device model id.<br>• NULL-terminated String.<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | - unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.622.2 Field Documentation

#### 8.622.2.1 char unpack_dms_GetModelID_t::modelid[255]

#### 8.622.2.2 swi_uint256_t unpack_dms_GetModelID_t::ParamPresenceMask

#### 8.622.2.3 uint16_t unpack_dms_GetModelID_t::Tlvresult

## 8.623 unpack_dms_GetNetworkTime_t Struct Reference

### Data Fields

- uint16_t source
- uint64_t timestamp
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.623.1 Detailed Description

This structure used to store unpack_dms_GetNetworkTime parameters

**Parameters**

| | |
|---|---|
| *source* | • Source of timestamp<br>0 - 32 kHz device clock<br>1 - CDMA network<br>2 - cdma2000 1xEV-DO network<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *timestamp* | • Count of 1.25 ms that have elapsed from the start of GPS time (Jan 6, 1980)<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

**Note**

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

### 8.623.2 Field Documentation

#### 8.623.2.1 swi_uint256_t unpack_dms_GetNetworkTime_t::ParamPresenceMask

#### 8.623.2.2 uint16_t unpack_dms_GetNetworkTime_t::source

#### 8.623.2.3 uint64_t unpack_dms_GetNetworkTime_t::timestamp

#### 8.623.2.4 uint16_t unpack_dms_GetNetworkTime_t::Tlvresult

## 8.624 unpack_dms_GetOfflineReason_t Struct Reference

### Data Fields

- uint32_t ∗ pReasonMask
- uint32_t ∗ pbPlatform
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.624.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pReasonMask[-OUT]* | • Optional parameter<br>• Bitmask of offline reasons<br>– 0x00000001 - Host image configuration issue<br>– 0x00000002 - PRI image configuration issue<br>– 0x00000004 - PRI version incompatible<br>– 0x00000008 - PRI copy issue<br>– All others - Reserved<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pbPlatform[OUT]* | • Optional parameter<br>• Is the device offline due to a platform restriction?<br>– 0 - No<br>– 1 - Yes<br>• Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.624.2 Field Documentation

#### 8.624.2.1 swi_uint256_t unpack_dms_GetOfflineReason_t::ParamPresenceMask

#### 8.624.2.2 uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform

#### 8.624.2.3 uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask

#### 8.624.2.4 uint16_t unpack_dms_GetOfflineReason_t::Tlvresult

## 8.625 unpack_dms_GetPower_t Struct Reference

### Data Fields

- uint32_t OperationMode
- uint32_t OfflineReason
- uint32_t HardwareControlledMode
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.625.1 Detailed Description

This structure used to store unpack_dms_GetPower parameters

**Parameters**

| | |
|---|---|
| *OperationMode* | - operating mode.<br>• 0 - Online (default).<br>• 1 - Low power (airplane) mode.<br>• 2 - Factory test mode.<br>• 3 - Offline.<br>• 4 - Resetting.<br>• 5 - Power off.<br>• 6 - Persistent low power (airplane) mode.<br>• 7 - Mode - only low power.<br>• Bit to check in ParamPresenceMask - **1** |

**Note**

Valid transitions for Power Modes

- Online to Low Power, Persistent low power, Factory test, Offline or Shut Down
- Low power to online, Persistent low power, Offline, or Shut Down
- Persistent low power to Online, Low power, Offline or Shut down
- Factory test to online
- Offline to Reset

**Parameters**

| | |
|---|---|
| *OfflineReason* | - offline reason.<br>• 0x0001 - Host image misconfiguration.<br>• 0x0002 - PRI image misconfiguration.<br>• 0x0004 - PRI version incompatible.<br>• 0x0008 - Device memory is full,cannot copy PRI information.<br>• Bit to check in ParamPresenceMask - **16** |
| *HardwareControlledMode* | - hardware restricted mode.<br>• 0x00 - FALSE.<br>• 0x01 - TRUE.<br>• Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | - unpack Tlv Result. |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.625.2 Field Documentation

#### 8.625.2.1 uint32_t unpack_dms_GetPower_t::HardwareControlledMode

#### 8.625.2.2 uint32_t unpack_dms_GetPower_t::OfflineReason

#### 8.625.2.3 uint32_t unpack_dms_GetPower_t::OperationMode

#### 8.625.2.4 swi_uint256_t unpack_dms_GetPower_t::ParamPresenceMask

#### 8.625.2.5 uint16_t unpack_dms_GetPower_t::Tlvresult

## 8.626 unpack_dms_GetPRLVersion_t Struct Reference

### Data Fields

- uint8_t u8PRLPreference
- uint16_t u16PRLVersion
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.626.1 Detailed Description

This structure used to store unpack_dms_GetPRLVersion parameters

**Parameters**

| | |
|---|---|
| *u8PRL-Preference* | • PRL Preference<br>– 0 - Unset<br>– 1 - Set<br>• Bit to check in ParamPresenceMask - **17** |
| *u16PRLVersion* | • PRL version of device.<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult* | • unpack Tlv Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.626.2 Field Documentation

#### 8.626.2.1 swi_uint256_t unpack_dms_GetPRLVersion_t::ParamPresenceMask

#### 8.626.2.2 uint16_t unpack_dms_GetPRLVersion_t::Tlvresult

#### 8.626.2.3 uint16_t unpack_dms_GetPRLVersion_t::u16PRLVersion

#### 8.626.2.4 uint8_t unpack_dms_GetPRLVersion_t::u8PRLPreference

## 8.627 unpack_dms_GetSerialNumbers_t Struct Reference

### Data Fields

- char esn [255]
- char imei_no [255]
- char meid [255]
- char imeisv_svn [255]
- swi_uint256_t ParamPresenceMask

### 8.627.1 Detailed Description

This structure used to store unpack_dms_GetSerialNumbers parameters

**Parameters**

| | |
|---|---|
| *esn* | - Electronic Serial Number of the device.<br>• NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed.<br>• Bit to check in ParamPresenceMask - **16** |
| *imei_no* | - International Mobile Equipment Identity of the device.<br>• NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed.<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *meid* | - Mobile Equipment Identifier of the device.<br>• NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed.<br>• Bit to check in ParamPresenceMask - **18** |
| *imeisv_svn* | - NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.imei software version revision.<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.627.2 Field Documentation

#### 8.627.2.1 char unpack_dms_GetSerialNumbers_t::esn[255]

#### 8.627.2.2 char unpack_dms_GetSerialNumbers_t::imei_no[255]

#### 8.627.2.3 char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]

#### 8.627.2.4 char unpack_dms_GetSerialNumbers_t::meid[255]

#### 8.627.2.5 swi_uint256_t unpack_dms_GetSerialNumbers_t::ParamPresenceMask

## 8.628 unpack_dms_GetUSBComp_t Struct Reference

### Data Fields

- uint8_t USBComp [255]
- uint8_t NumSupUSBComps
- uint8_t SupUSBComps
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.628.1 Detailed Description

This structure is used to store retrieved USB Composition

**Parameters**

| | |
|---|---|
| *USBComp* | • Current USB Composition(optional parameter)<br>• Values:<br>– 0..5 - Reserved (non-QMI)<br>– 6 - DM NMEA AT QMI<br>– 7 - DM NMEA AT QMI1 QMI2 QMI3<br>– 8 - DM NMEA AT MBIM<br>– 9 - MBIM<br>– 10 - NMEA MBIM<br>– 11 - DM MBIM<br>– 12 - DM NMEA MBIM 13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces<br>– 13 - 6 for QMI, 8 for MBIM<br>– 14 - 6 for QMI, 9 for MBIM<br>– 15 - 6 for QMI, 10 for MBIM<br>– 16 - 6 for QMI, 11 for MBIM<br>– 17 - 6 for QMI, 12 for MBIM<br>– 18 - 7 for QMI, 8 for MBIM<br>– 19 - 7 for QMI, 9 for MBIM<br>– 20 - 7 for QMI, 10 for MBIM<br>– 21 - 7 for QMI, 11 for MBIM<br>– 22 - 7 for QMI, 12 for MBIM<br>• Bit to check in ParamPresenceMask - **16** |
| *NumSupUSB-Comps* | • Number of supported USB compositions in the parameter to follow<br>• Range - 0-255<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *SupUSBComps* | • Optional parameter<br>• List of supported USB compositions( 1 Byte each - Max 255 )<br>• Total length is defined by pNumSupUSBComps parameter<br>• Values:<br>– 0..5 - Reserved (non-QMI)<br>– 6 - DM NMEA AT QMI<br>– 7 - DM NMEA AT QMI1 QMI2 QMI3<br>– 8 - DM NMEA AT MBIM<br>– 9 - MBIM<br>– 10 - NMEA MBIM<br>– 11 - DM MBIM<br>– 12 - DM NMEA MBIM<br>13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces<br>– 13 - 6 for QMI, 8 for MBIM<br>– 14 - 6 for QMI, 9 for MBIM<br>– 15 - 6 for QMI, 10 for MBIM<br>– 16 - 6 for QMI, 11 for MBIM<br>– 17 - 6 for QMI, 12 for MBIM<br>– 18 - 7 for QMI, 8 for MBIM<br>– 19 - 7 for QMI, 9 for MBIM<br>– 20 - 7 for QMI, 10 for MBIM<br>– 21 - 7 for QMI, 11 for MBIM<br>– 22 - 7 for QMI, 12 for MBIM<br>• Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.628.2 Field Documentation

#### 8.628.2.1 uint8_t unpack_dms_GetUSBComp_t::NumSupUSBComps

#### 8.628.2.2 swi_uint256_t unpack_dms_GetUSBComp_t::ParamPresenceMask

#### 8.628.2.3 uint8_t unpack_dms_GetUSBComp_t::SupUSBComps

#### 8.628.2.4 uint16_t unpack_dms_GetUSBComp_t::Tlvresult

#### 8.628.2.5 uint8_t unpack_dms_GetUSBComp_t::USBComp[255]

## 8.629 unpack_dms_GetVoiceNumber_t Struct Reference

### Data Fields

- uint8_t voiceNumberSize

- char VoiceNumber [255]
- uint8_t minSize
- char MIN [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.629.1 Detailed Description

Structure to store the voice number in use by the device

**Parameters**

| | |
|---|---|
| *voiceNumber-Size* | • Maximum number of characters (including NULL terminator) that the voice number array can contain.<br>• Bit to check in ParamPresenceMask - **1** |
| *VoiceNumber* | • Voice number string: MDN or MS ISDN<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *minSize* | • Maximum number of characters (including NULL terminator) that the MIN array can contain.<br>• Bit to check in ParamPresenceMask - **16** |
| *MIN* | • Optional Parameter<br>• MIN string: Empty string returned when MIN is not supported/ programmed.<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.629.2 Field Documentation

#### 8.629.2.1 char unpack_dms_GetVoiceNumber_t::MIN[255]

#### 8.629.2.2 uint8_t unpack_dms_GetVoiceNumber_t::minSize

#### 8.629.2.3 swi_uint256_t unpack_dms_GetVoiceNumber_t::ParamPresenceMask

#### 8.629.2.4 uint16_t unpack_dms_GetVoiceNumber_t::Tlvresult

#### 8.629.2.5 char unpack_dms_GetVoiceNumber_t::VoiceNumber[255]

#### 8.629.2.6 uint8_t unpack_dms_GetVoiceNumber_t::voiceNumberSize

## 8.630 unpack_dms_PSMCfgChange_ind_t Struct Reference

### Data Fields

- dms_PSMEnableStateIndTlv EnableState
- dms_PSMActiveTimerIndTlv ActiveTimer
- dms_PSMPeriodicUpdateTimerIndTlv PeriodicUpdateTimer
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.630.1 Detailed Description

DMS Event Report indication structure

**Parameters**

| | |
|---|---|
| *EnableState* | • See dms_PSMEnableStateIndTlv<br>• Bit to check in ParamPresenceMask - **16** |
| *ActiveTimer* | • See dms_PSMActiveTimerIndTlv<br>• Bit to check in ParamPresenceMask - **17** |
| *PeriodicUpdate-Timer* | • See dms_PSMPeriodicUpdateTimerIndTlv<br>• Bit to check in ParamPresenceMask - **18** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.630.2 Field Documentation

#### 8.630.2.1 dms_PSMActiveTimerIndTlv unpack_dms_PSMCfgChange_ind_t::ActiveTimer

#### 8.630.2.2 dms_PSMEnableStateIndTlv unpack_dms_PSMCfgChange_ind_t::EnableState

#### 8.630.2.3 swi_uint256_t unpack_dms_PSMCfgChange_ind_t::ParamPresenceMask

#### 8.630.2.4 dms_PSMPeriodicUpdateTimerIndTlv unpack_dms_PSMCfgChange_ind_t::PeriodicUpdateTimer

#### 8.630.2.5 uint16_t unpack_dms_PSMCfgChange_ind_t::Tlvresult

## 8.631 unpack_dms_ResetToFactoryDefaults_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.631.1 Detailed Description

This structure contains reset to factory default unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.631.2 Field Documentation

#### 8.631.2.1 swi_uint256_t unpack_dms_ResetToFactoryDefaults_t::ParamPresenceMask

#### 8.631.2.2 uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult

## 8.632 unpack_dms_SetActivationStatusCallback_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.632.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter unpack.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.632.2 Field Documentation

#### 8.632.2.1 swi_uint256_t unpack_dms_SetActivationStatusCallback_t::ParamPresenceMask

#### 8.632.2.2 uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult

## 8.633 unpack_dms_SetCrashAction_t Struct Reference

### Data Fields

- uint8_t notused
- swi_uint256_t ParamPresenceMask

### 8.633.1 Detailed Description

This structure is used to store unpack_dms_SetCrashAction parameters

**Parameters**

| | |
|---|---|
| *notused* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.633.2 Field Documentation

#### 8.633.2.1 uint8_t unpack_dms_SetCrashAction_t::notused

#### 8.633.2.2 swi_uint256_t unpack_dms_SetCrashAction_t::ParamPresenceMask

## 8.634 unpack_dms_SetCustFeature_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.634.1 Detailed Description

This structure is used to store unpack_dms_SetCustFeature parameters

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.634.2 Field Documentation

#### 8.634.2.1 swi_uint256_t unpack_dms_SetCustFeature_t::ParamPresenceMask

#### 8.634.2.2 uint16_t unpack_dms_SetCustFeature_t::Tlvresult

## 8.635 unpack_dms_SetCustFeaturesV2_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.635.1 Detailed Description

This structure contains customization settings set to modem unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.635.2 Field Documentation

#### 8.635.2.1 swi_uint256_t unpack_dms_SetCustFeaturesV2_t::ParamPresenceMask

#### 8.635.2.2 uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult

## 8.636 unpack_dms_SetEventReport_ind_t Struct Reference

### Data Fields

- dms_ActivationStatusTlv ActivationStatusTlv
- dms_OperatingModeTlv OperatingModeTlv
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.636.1 Detailed Description

DMS Event Report indication structure

**Parameters**

| | |
|---|---|
| *ActivationStatus-Tlv* | • See dms_ActivationStatusTlv<br>• Bit to check in ParamPresenceMask - **19** |
| *OperatingMode-Tlv* | • See dms_OperatingModeTlv<br>• Bit to check in ParamPresenceMask - **20** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.636.2 Field Documentation

#### 8.636.2.1 dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv

#### 8.636.2.2 dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv

#### 8.636.2.3 swi_uint256_t unpack_dms_SetEventReport_ind_t::ParamPresenceMask

#### 8.636.2.4 uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult

## 8.637 unpack_dms_SetEventReport_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.637.1 Detailed Description

This structure is used to store unpack_dms_SetEventReport parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.637.2 Field Documentation

#### 8.637.2.1 swi_uint256_t unpack_dms_SetEventReport_t::ParamPresenceMask

#### 8.637.2.2 uint16_t unpack_dms_SetEventReport_t::Tlvresult

## 8.638 unpack_dms_SetFirmwarePreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.638.1 Detailed Description

This structure is used to store unpack_dms_SetFirmwarePreference parameters

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.638.2 Field Documentation

#### 8.638.2.1 swi_uint256_t unpack_dms_SetFirmwarePreference_t::ParamPresenceMask

#### 8.638.2.2 uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult

## 8.639 unpack_dms_SetIndicationRegister_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.639.1 Detailed Description

This structure is used to store unpack_dms_SetIndicationRegister parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.639.2 Field Documentation

#### 8.639.2.1 swi_uint256_t unpack_dms_SetIndicationRegister_t::ParamPresenceMask

#### 8.639.2.2 uint16_t unpack_dms_SetIndicationRegister_t::Tlvresult

## 8.640 unpack_dms_SetPower_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.640.1 Detailed Description

Structure to store unpack the operating mode of the device.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.640.2 Field Documentation

#### 8.640.2.1 swi_uint256_t unpack_dms_SetPower_t::ParamPresenceMask

#### 8.640.2.2 uint16_t unpack_dms_SetPower_t::Tlvresult

## 8.641 unpack_dms_SetUSBComp_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.641.1 Detailed Description

This structure is used to store unpack USB composition information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv Result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.641.2 Field Documentation

#### 8.641.2.1 swi_uint256_t unpack_dms_SetUSBComp_t::ParamPresenceMask

#### 8.641.2.2 uint16_t unpack_dms_SetUSBComp_t::Tlvresult

## 8.642 unpack_dms_SLQSDmsSwiGetPCInfo_t Struct Reference

### Data Fields

- uint8_t opMode
- int has_LpmFlag
- uint32_t LpmFlag
- int has_Wdisable
- uint8_t Wdisable
- int has_PowerOffMode
- uint8_t PowerOffMode
- int has_PersistentLpm
- uint8_t PersistentLpm
- swi_uint256_t ParamPresenceMask

### 8.642.1 Detailed Description

This structure contains the TLV required to Get device power control status information.

**Parameters**

| | |
|---|---|
| *opMode* | • Selected operating mode.<br>• Values<br>– 0 - Online<br>– 1 - Low power<br>– 2 - Factory Test mode<br>– 3 - Offline<br>– 4 - Resetting<br>– 5 - Shutting down<br>– 6 - Persistent low power<br>– 8 - Conducting network test for GSM/WCDMA<br>• Bit to check in ParamPresenceMask - **1** |
| *has_LpmFlag* | LPM Flag Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

- Bit to check in ParamPresenceMask - **16**

**Parameters**

| | |
|---|---|
| *LpmFlag[-Optional]* | • LPM Force Flags<br>• Bitmask<br>– bit0 - Tracks the state of W_DISABLE TLV, indicating state of the W_DISABLE switch.<br>– bit1 - Set if LPM is requested by a host request, such as AT command, QMI or MBIM request. Cleared by a host request to return to online mode. Also set when the device is cold or warm booted in persistent LPM.<br>– bit2 - Set when the device temperature is outside the valid operating range. Cleared if the temperature returns to the normal range.<br>– bit3 - Set when the device voltage is outside the valid operating range. Cleared if the voltage returns to the normal range.<br>– bit4 - Set on power up when BIOS locking is enabled. Cleared when the host has disabled the BIOS lock.<br>– bit5 - Set if the current device configuration does not match the GOBI image preference.<br>– bit6-31 - Additional LPM causes may be added to future products.<br>• Bit to check in ParamPresenceMask - **16** |
| *has_Wdisable* | W-Disable Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

- Bit to check in ParamPresenceMask - **17**

**Parameters**

| | |
|---|---|
| *Wdisable[-Optional]* | • W_DISABLE<br>• Values<br>– 0 - Switch set to ON position<br>– 1 - Switch set to OFF position<br>• Bit to check in ParamPresenceMask - **17** |
| *has_PowerOff-Mode* | Power off mode Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

- Bit to check in ParamPresenceMask - **18**

**Parameters**

| | |
|---|---|
| *PowerOffMode[-Optional]* | • Power-off Mode<br>• Action taken when W_DISABLE is switched to the OFF position<br>– 0 - LPM<br>– 1 - Shutdown<br>– 2 - Ignore<br>• Bit to check in ParamPresenceMask - **18** |
| *has_Persistent-Lpm* | Persistent LPM Availability.<br>• 0 : Unavailable.<br>• 1 : Available. |

- Bit to check in ParamPresenceMask - **19**

**Parameters**

| | |
|---|---|
| *PersistentLpm[-Optional]* | • LPM Persistence<br>• Values<br>– 0 - Non-persistent LPM<br>– 1 - Persistent LPM<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

## 8.642.2 Field Documentation

### 8.642.2.1 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_LpmFlag

### 8.642.2.2 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_PersistentLpm

### 8.642.2.3 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_PowerOffMode

#### 8.642.2.4 int unpack_dms_SLQSDmsSwiGetPCInfo_t::has_Wdisable

#### 8.642.2.5 uint32_t unpack_dms_SLQSDmsSwiGetPCInfo_t::LpmFlag

#### 8.642.2.6 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::opMode

#### 8.642.2.7 swi_uint256_t unpack_dms_SLQSDmsSwiGetPCInfo_t::ParamPresenceMask

#### 8.642.2.8 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::PersistentLpm

#### 8.642.2.9 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::PowerOffMode

#### 8.642.2.10 uint8_t unpack_dms_SLQSDmsSwiGetPCInfo_t::Wdisable

## 8.643 unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.643.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

**Parameters**

| | |
|---|---|
| *type[OUT]* | • type of reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - warm<br>– 2 - hard<br>– 3 - crash<br>– 4 - power down<br>• Bit to check in ParamPresenceMask - **1** |
| *source[OUT]* | • entity which initiated the reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - user requested ( AT!RESET, AT!BOOTHOLD, FW/PRI download - including host-initiated image switching)<br>– 2 - hardware switch (W_DISABLE)<br>– 3 - temperature critical<br>– 4 - voltage critical<br>– 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset)<br>– 6 - LWM2M (Light Weight M2M client (internal process for LWM2M))<br>– 7 - OMA-DM<br>– 8 - FOTA |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *Tlvresult[OUT]* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.643.2 Field Documentation

#### 8.643.2.1 swi_uint256_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::ParamPresenceMask

#### 8.643.2.2 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source

#### 8.643.2.3 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult

#### 8.643.2.4 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type

## 8.644 unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference

### Data Fields

- uint8_t type
- uint8_t source
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.644.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

**Parameters**

| | |
|---|---|
| *type[OUT]* | • type of reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - warm<br>– 2 - hard<br>– 3 - crash<br>– 4 - power down<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *source[OUT]* | • entity which initiated the reset or power down, possible values listed below:<br>– 0 - unknown<br>– 1 - user requested ( AT!RESET, AT!BOOTHOLD, FW/PRI download - including host-initiated image switching)<br>– 2 - hardware switch (W_DISABLE)<br>– 3 - temperature critical<br>– 4 - voltage critical<br>– 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset)<br>– 6 - LWM2M (Light Weight M2M client (internal process for LWM2M))<br>– 7 - OMA-DM<br>– 8 - FOTA |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.644.2 Field Documentation

#### 8.644.2.1 swi_uint256_t unpack_dms_SLQSDmsSwiGetResetInfo_t::ParamPresenceMask

#### 8.644.2.2 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::source

#### 8.644.2.3 uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_t::Tlvresult

#### 8.644.2.4 uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_t::type

## 8.645 unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.645.1 Detailed Description

This structure contains set registration state for different indication unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.645.2 Field Documentation

#### 8.645.2.1 swi_uint256_t unpack_dms_SLQSDmsSwiIndicationRegister_t::ParamPresenceMask

#### 8.645.2.2 uint16_t unpack_dms_SLQSDmsSwiIndicationRegister_t::Tlvresult

## 8.646 unpack_dms_SLQSGetBandCapability_t Struct Reference

### Data Fields

- uint64_t bandCapability
- int is_LteBandCapability_Available
- uint64_t LteBandCapability
- int is_TdsBandCapability_Available
- uint64_t TdsBandCapability
- swi_uint256_t ParamPresenceMask

### 8.646.1 Detailed Description

This structure contains the Band Capabilities response.

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

**Parameters**

| | |
|---|---|
| *bandCapability[-OUT]* | Bitmask of bands supported by the device<br>• Bit 0 - Band class 0, A-system<br>• Bit 1 - Band class 0, B-system<br>• Bit 2 - Band class 1, all blocks<br>• Bit 3 - Band class 2<br>• Bit 4 - Band class 3, A-system<br>• Bit 5 - Band class 4, all blocks<br>• Bit 6 - Band class 5, all blocks<br>• Bit 7 - GSM DCS band (1800)<br>• Bit 8 - GSM Extended GSM (E-GSM) band (900)<br>• Bit 9 - GSM Primary GSM (P-GSM) band (900)<br>• Bit 10 - Band class 6<br>• Bit 11 - Band class 7<br>• Bit 12 - Band class 8<br>• Bit 13 - Band class 9<br>• Bit 14 - Band class 10<br>• Bit 15 - Band class 11<br>• Bit 16 - GSM 450 band<br>• Bit 17 - GSM 480 band<br>• Bit 18 - GSM 750 band<br>• Bit 19 - GSM 850 band<br>• Bit 20 - GSM railways GSM band (900)<br>• Bit 21 - GSM PCS band (1900)<br>• Bit 22 - WCDMA (Europe, Japan, and China) 2100 band<br>• Bit 23 - WCDMA US PCS 1900 band<br>• Bit 24 - WCDMA (Europe and China) DCS 1800 band<br>• Bit 25 - WCDMA US 1700 band<br>• Bit 26 - WCDMA US 850 band<br>• Bit 27 - WCDMA Japan 800 band<br>• Bit 28 - Band class 12<br>• Bit 29 - Band class 14<br>• Bit 30 - Reserved<br>• Bit 31 - Band class 15<br>• Bits 32 through 47 - Reserved<br>• Bit 48 - WCDMA Europe 2600 band<br>• Bit 49 - WCDMA Europe and Japan 900 band<br>• Bit 50 - WCDMA Japan 1700 band<br>• Bits 51 through 55 - Reserved<br>• Bit 56 - Band class 16<br>• Bit 57 - Band class 17<br>• Bit 58 - Band class 18<br>• Bit 59 - Band class 19<br>• Bit 60 - WCDMA Japan 850 band<br>• Bit 61 - WCDMA 1500 band<br>• Bits 62 and 63 - Reserved<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *is_LteBand-Capability_-Available[OUT]* | LTE Band Capabilities Availability.<br>• 0 : Unavailable.<br>• 1 : Available.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *LteBand-Capability[OUT]* | Bitmask of LTE bands supported by the device<br>• Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170<br>• Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990<br>• Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880<br>• Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155<br>• Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894<br>• Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885<br>• Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690<br>• Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960<br>• Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9<br>• Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170<br>• Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9<br>• Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746<br>• Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756<br>• Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768<br>• Bits 14 and 15 - Reserved<br>• Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746<br>• Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875<br>• Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890<br>• Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821<br>• Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9<br>• Bit 21 - Reserved<br>• Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200<br>• Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559<br>• Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995<br>• Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894<br>• Bit 26 - Reserved<br>• Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803<br>• Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728<br>• Bit 29 - LTE EUTRAN Band 30 UL: 2350-2360; DL: 2305-2315<br>• Bit 30 - Reserved<br>• Bit 31 - LTE EUTRAN Band 32 DL: 9920-10359<br>• Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920<br>• Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025<br>• Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910<br>• Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990<br>• Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930<br>• Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620<br>• Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920<br>• Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400<br>• Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690<br>• Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600<br>• Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800<br>• Bits 43 through 64 - Reserved<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *is_TdsBand-Capability_-Available[OUT]* | TDS Band Capabilities Availability.<br>• 0 : Unavailable.<br>• 1 : Available.<br>• Bit to check in ParamPresenceMask - **17** |
| *TdsBand-Capability[OUT]* | Bitmask of TDS bands supported by the device.<br>• Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz<br>• Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz<br>• Bit 2 - TDS Band C 1910 to 1930 MHz<br>• Bit 3 - TDS Band D 2570 to 2620 MHz<br>• Bit 4 - TDS Band E 2300 to 2400 MHz<br>• Bit 5 - TDS Band F 1880 to 1920 MHz<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.646.2 Field Documentation

#### 8.646.2.1 uint64_t unpack_dms_SLQSGetBandCapability_t::bandCapability

#### 8.646.2.2 int unpack_dms_SLQSGetBandCapability_t::is_LteBandCapability_Available

#### 8.646.2.3 int unpack_dms_SLQSGetBandCapability_t::is_TdsBandCapability_Available

#### 8.646.2.4 uint64_t unpack_dms_SLQSGetBandCapability_t::LteBandCapability

#### 8.646.2.5 swi_uint256_t unpack_dms_SLQSGetBandCapability_t::ParamPresenceMask

#### 8.646.2.6 uint64_t unpack_dms_SLQSGetBandCapability_t::TdsBandCapability

## 8.647 unpack_dms_SLQSGetERIFile_t Struct Reference

### Data Fields

- eriDataparams eriFile
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.647.1 Detailed Description

This structure contains Get ERI file unpack

**Parameters**

| | |
|---|---|
| *eriFile* | • Pointer to structure ERIFileparams<br>• See eriDataparams for more information<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.647.2 Field Documentation

#### 8.647.2.1 eriDataparams unpack_dms_SLQSGetERIFile_t::eriFile

#### 8.647.2.2 swi_uint256_t unpack_dms_SLQSGetERIFile_t::ParamPresenceMask

#### 8.647.2.3 uint16_t unpack_dms_SLQSGetERIFile_t::Tlvresult

## 8.648 unpack_dms_SLQSGetPowerSaveModeConfig_t Struct Reference

### Data Fields

- dms_PSMEnableStateTlv * pPsmEnableState
- dms_PSMDurationThresholdTlv * pDurationThreshold
- dms_PSMDurationDueToOOSTlv * pDurationDueToOOS
- dms_PSMRandomizationWindowTlv * pRandomizationWindow
- dms_PSMActiveTimerTlv * pActiveTimer
- dms_PSMPeriodicUpdateTimerTlv * pPeriodicUpdateTimer
- dms_PSMEarlyWakeupTimeTlv * pEarlyWakeupTime
- swi_uint256_t ParamPresenceMask

### 8.648.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

**Parameters**

| | |
|---|---|
| *pPsmEnable-State* | [Optional]<br>• See dms_PSMEnableStateTlv for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pDuration-Threshold* | [Optional]<br>• See dms_PSMDurationThresholdTlv for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pDurationDueTo-OOS* | [Optional]<br>• See dms_PSMDurationDueToOOSTlv for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *pRandomization-Window* | [Optional]<br>• See dms_PSMRandomizationWindowTlv for more information<br>• Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *pActiveTimer* | [Optional]<br>• See dms_PSMActiveTimerTlv for more information<br>• Bit to check in ParamPresenceMask - **20** |
| *pPeriodic-UpdateTimer* | [Optional]<br>• See dms_PSMPeriodicUpdateTimerTlv for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *pEarlyWakeup-Time* | [Optional]<br>• See dms_PSMEarlyWakeupTimeTlv for more information<br>• Bit to check in ParamPresenceMask - **22** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.648.2 Field Documentation

#### 8.648.2.1 dms_PSMActiveTimerTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pActiveTimer

#### 8.648.2.2 swi_uint256_t unpack_dms_SLQSGetPowerSaveModeConfig_t::ParamPresenceMask

#### 8.648.2.3 dms_PSMDurationDueToOOSTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pDurationDueToOOS

#### 8.648.2.4 dms_PSMDurationThresholdTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pDurationThreshold

#### 8.648.2.5 dms_PSMEarlyWakeupTimeTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pEarlyWakeupTime

#### 8.648.2.6 dms_PSMPeriodicUpdateTimerTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pPeriodicUpdateTimer

#### 8.648.2.7 dms_PSMEnableStateTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pPsmEnableState

#### 8.648.2.8 dms_PSMRandomizationWindowTlv* unpack_dms_SLQSGetPowerSaveModeConfig_t::pRandomization-Window

## 8.649 unpack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.649.1 Detailed Description

This structure is used to store unpack_dms_SLQSSetPowerSaveModeConfig parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.649.2 Field Documentation

#### 8.649.2.1 swi_uint256_t unpack_dms_SLQSSetPowerSaveModeConfig_t::ParamPresenceMask

#### 8.649.2.2 uint16_t unpack_dms_SLQSSetPowerSaveModeConfig_t::Tlvresult

## 8.650 unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.650.1 Detailed Description

This structure contains Clear Dying GASP unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.650.2 Field Documentation

#### 8.650.2.1 swi_uint256_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::ParamPresenceMask

#### 8.650.2.2 uint16_t unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult

## 8.651 unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- crashInfoParams crashInfoParam
- swi_uint256_t ParamPresenceMask

### 8.651.1 Detailed Description

This structure contains SWI get crash information unpack information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *crashInfoParam[-OUT]* | • See crashInfoParams<br>• crashStatus<br>– Bit to check in ParamPresenceMask - **1**<br>• crashInfo<br>– Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.651.2 Field Documentation

#### 8.651.2.1 crashInfoParams unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam

#### 8.651.2.2 swi_uint256_t unpack_dms_SLQSSwiGetCrashInfo_t::ParamPresenceMask

#### 8.651.2.3 uint16_t unpack_dms_SLQSSwiGetCrashInfo_t::Tlvresult

## 8.652 unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference

### Data Fields

- packgetDyingGaspCfg * pGetDyingGaspCfg
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.652.1 Detailed Description

This structure contains Get Dying GASP Config unpack

**Parameters**

| | |
|---|---|
| *pGetDyingGasp-Cfg* | • See packgetDyingGaspCfg<br>• pDestSMSNum.<br>– Bit to check in ParamPresenceMask - **16**<br>• pDestSMSContent.<br>– Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.652.2 Field Documentation

#### 8.652.2.1 swi_uint256_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::ParamPresenceMask

#### 8.652.2.2 packgetDyingGaspCfg* unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg

#### 8.652.2.3 uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult

## 8.653 unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference

### Data Fields

- packgetDyingGaspStatistics * pGetDyingGaspStatistics
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.653.1 Detailed Description

This structure contains Get Dying GASP Statistics unpack.

**Parameters**

| | |
|---|---|
| *pGetDyingGasp-Statistics* | • See packgetDyingGaspStatistics<br>• pTimeStamp.<br>– Bit to check in ParamPresenceMask - **16**<br>• pSMSAttemptedFlag.<br>– Bit to check in ParamPresenceMask - **17** |
| *Tlvresult* | • Unpack Result |

### 8.653.2 Field Documentation

#### 8.653.2.1 swi_uint256_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::ParamPresenceMask

#### 8.653.2.2 packgetDyingGaspStatistics* unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics

#### 8.653.2.3 uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult

## 8.654 unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference

### Data Fields

- uint8_t numEntries
- image_info_t * pCurrImgInfo
- char priver [16]
- char pkgver [16]
- char fwvers [16]
- char carrier [16]
- swi_uint256_t ParamPresenceMask

### 8.654.1 Detailed Description

This structure is used to store unpack_dms_SLQSSwiGetFirmwareCurr parameters

**Parameters**

| | |
|---|---|
| *numEntries[IN/-OUT]* | • Number of entries in the image list to follow<br>• The size of the list pCurrImgInfo must be specified when calling the API<br>– Bit to check in ParamPresenceMask - **1** |
| *pCurrImgInfo[O-UT]* | • Currently Active Image List<br>– Bit to check in ParamPresenceMask - **1** |
| *priver[OUT]* | • PRI version of the currently running firmware<br>• Bit to check in ParamPresenceMask - **1** |
| *pkgver[OUT]* | • Package version of the currently running firmware<br>• Bit to check in ParamPresenceMask - **1** |
| *fwvers[OUT]* | • firmware version of the currently running firmware<br>• Bit to check in ParamPresenceMask - **1** |
| *carrier[OUT]* | • Carrier string of the currently running firmware<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.654.2 Field Documentation

#### 8.654.2.1 char unpack_dms_SLQSSwiGetFirmwareCurr_t::carrier[16]

#### 8.654.2.2 char unpack_dms_SLQSSwiGetFirmwareCurr_t::fwvers[16]

#### 8.654.2.3 uint8_t unpack_dms_SLQSSwiGetFirmwareCurr_t::numEntries

#### 8.654.2.4 swi_uint256_t unpack_dms_SLQSSwiGetFirmwareCurr_t::ParamPresenceMask

#### 8.654.2.5 image_info_t* unpack_dms_SLQSSwiGetFirmwareCurr_t::pCurrImgInfo

#### 8.654.2.6 char unpack_dms_SLQSSwiGetFirmwareCurr_t::pkgver[16]

#### 8.654.2.7 char unpack_dms_SLQSSwiGetFirmwareCurr_t::priver[16]

## 8.655 unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference

### Data Fields

- uint32_t ResCode
- uint8_t imgType
- uint32_t refData
- uint8_t refString [15]
- uint8_t logString [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.655.1 Detailed Description

This structure is used to store Firmware Update Status

**Parameters**

| | |
|---|---|
| *ResCode* | • FW Update Result Code<br>• Values:<br>– 0x00000001 - Successful<br>– 0xFFFFFFFF - Unknown (due to power off reset after firmware update )<br>– 0x100000nn - File update errors while nn will be the exact error number:<br>* 00 - General error<br>– 0x200000nn - NVUP update errors while nn will be the exact error number:<br>* 00 - General error<br>– 0x40000nnn - FOTA update agent errors while nnn will be the exact error number:<br>* 000 ∼ 0FF - Insignia defined error code<br>* 100 ∼ 1FF - Sierra defined error code<br>* See qaGobiApiTableFwDldErrorCodes.h for more detailed information<br>– 0x800000nn - FDT/SSDP reported errors while nn will be the exact error number<br>* See qaGobiApiTableFwDldErrorCodes.h for more detailed information<br>• Bit to check in ParamPresenceMask - **16** |
| *imgType* | • Optional parameter<br>• Firmware image type that failed the update<br>• Bit to check in ParamPresenceMask - **17** |
| *refData* | • Optional parameter<br>• Failed image reference data<br>• This is normally the offset of the image that caused the failure<br>• Bit to check in ParamPresenceMask - **18** |
| *refString* | • Optional parameter<br>• Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.<br>• Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *logString* | • Optional parameter<br>• Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.<br>• Bit to check in ParamPresenceMask - **20** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.655.2 Field Documentation

#### 8.655.2.1 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType

#### 8.655.2.2 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]

#### 8.655.2.3 swi_uint256_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ParamPresenceMask

#### 8.655.2.4 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData

#### 8.655.2.5 uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]

#### 8.655.2.6 uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode

#### 8.655.2.7 uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult

## 8.656 unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- char manString [255]
- char modelString [255]
- char swVerString [255]
- char plasmaIDString [255]
- swi_uint256_t ParamPresenceMask

### 8.656.1 Detailed Description

This structure contains SWI get host device info unpack information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Result |
| *manString[OUT]* | • Host Device Manufacturer Name<br>• Null terminated ASCII String<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *modelString[OUT]* | • Host Device Model String<br>• Null terminated ASCII String<br>• Bit to check in ParamPresenceMask - **17** |
| *swVerString[OUT]* | • Host Device Software Version String<br>• Null terminated ASCII string<br>• Bit to check in ParamPresenceMask - **18** |
| *plasmaIDString[OUT]* | • Host Device Plasma ID String<br>• Null terminated alphanumeric ASCII String.<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresenceMask[OUT]* | • bitmask representation to indicate valid parameters. |

### 8.656.2 Field Documentation

#### 8.656.2.1 char unpack_dms_SLQSSwiGetHostDevInfo_t::manString[255]

#### 8.656.2.2 char unpack_dms_SLQSSwiGetHostDevInfo_t::modelString[255]

#### 8.656.2.3 swi_uint256_t unpack_dms_SLQSSwiGetHostDevInfo_t::ParamPresenceMask

#### 8.656.2.4 char unpack_dms_SLQSSwiGetHostDevInfo_t::plasmaIDString[255]

#### 8.656.2.5 char unpack_dms_SLQSSwiGetHostDevInfo_t::swVerString[255]

#### 8.656.2.6 uint16_t unpack_dms_SLQSSwiGetHostDevInfo_t::Tlvresult

## 8.657 unpack_dms_SLQSSwiGetOSInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- char nameString [255]
- char versionString [255]
- swi_uint256_t ParamPresenceMask

### 8.657.1 Detailed Description

This structure contains SWI get host os info unpack information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

| | |
|---|---|
| *nameString[OUT]* | • Host operating system name<br>• Bit to check in ParamPresenceMask - **16** |
| *VersionString[OUT]* | • Host operating system version<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.657.2 Field Documentation

#### 8.657.2.1 char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]

#### 8.657.2.2 swi_uint256_t unpack_dms_SLQSSwiGetOSInfo_t::ParamPresenceMask

#### 8.657.2.3 uint16_t unpack_dms_SLQSSwiGetOSInfo_t::Tlvresult

#### 8.657.2.4 char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]

## 8.658 unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- char meidString [8]
- swi_uint256_t ParamPresenceMask

### 8.658.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

**Parameters**

| | |
|---|---|
| *Tlvresult* | • pack Result |
| *meidString[OUT]* | • optional parameter, mobile equipment identifier<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.658.2 Field Documentation

#### 8.658.2.1 char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]

#### 8.658.2.2 swi_uint256_t unpack_dms_SLQSSwiGetSerialNoExt_t::ParamPresenceMask

#### 8.658.2.3 uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult

## 8.659 unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.659.1 Detailed Description

This structure contains set Dying GASP Config unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.659.2 Field Documentation

#### 8.659.2.1 swi_uint256_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::ParamPresenceMask

#### 8.659.2.2 uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult

## 8.660 unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.660.1 Detailed Description

This structure contains SWI set host dev info unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.660.2 Field Documentation

#### 8.660.2.1 swi_uint256_t unpack_dms_SLQSSwiSetHostDevInfo_t::ParamPresenceMask

#### 8.660.2.2 uint16_t unpack_dms_SLQSSwiSetHostDevInfo_t::Tlvresult

## 8.661 unpack_dms_SLQSSwiSetOSInfo_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.661.1 Detailed Description

This structure contains SWI set host OS info unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.661.2 Field Documentation

#### 8.661.2.1 swi_uint256_t unpack_dms_SLQSSwiSetOSInfo_t::ParamPresenceMask

#### 8.661.2.2 uint16_t unpack_dms_SLQSSwiSetOSInfo_t::Tlvresult

## 8.662 unpack_dms_SLQSUIMGetState_t Struct Reference

### Data Fields

- uint8_t state
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.662.1 Detailed Description

This structure contains UIM get state unpack information

**Parameters**

| | |
|---|---|
| *state[OUT]* | • UIM state<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.662.2 Field Documentation

#### 8.662.2.1 swi_uint256_t unpack_dms_SLQSUIMGetState_t::ParamPresenceMask

#### 8.662.2.2 uint8_t unpack_dms_SLQSUIMGetState_t::state

#### 8.662.2.3 uint16_t unpack_dms_SLQSUIMGetState_t::Tlvresult

## 8.663 unpack_dms_SwiEventReportCallBack_ind_t Struct Reference

### Data Fields

- dms_TemperatureTlv TempTlv
- dms_VoltageTlv VoltTlv
- swi_uint256_t ParamPresenceMask

### 8.663.1 Detailed Description

Structure used to SWI event report indication parameters.

**Parameters**

| | |
|---|---|
| *TempTlv* | • Temperature status tlv<br>• See dms_TemperatureTlv for more information |
| *VoltTlv* | • Voltage status tlv<br>• see dms_VoltageTlv for more information |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

**Note**

: None

### 8.663.2 Field Documentation

#### 8.663.2.1 swi_uint256_t unpack_dms_SwiEventReportCallBack_ind_t::ParamPresenceMask

#### 8.663.2.2 dms_TemperatureTlv unpack_dms_SwiEventReportCallBack_ind_t::TempTlv

#### 8.663.2.3 dms_VoltageTlv unpack_dms_SwiEventReportCallBack_ind_t::VoltTlv

## 8.664 unpack_dms_SwiSetEventReport_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.664.1 Detailed Description

This structure is used to store unpack_dms_SwiSetEventReport parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.664.2 Field Documentation

#### 8.664.2.1 swi_uint256_t unpack_dms_SwiSetEventReport_t::ParamPresenceMask

#### 8.664.2.2 uint16_t unpack_dms_SwiSetEventReport_t::Tlvresult

## 8.665 unpack_dms_UIMGetControlKeyStatus_t Struct Reference

### Data Fields

- uint8_t facilityState
- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.665.1 Detailed Description

This structure contains PIN retries status

**Parameters**

| | |
|---|---|
| *facilityState[OUT]* | • Control key status<br>– 0 - Deactivated<br>– 1 - Activated<br>– 2 - Blocked<br>• Bit to check in ParamPresenceMask - **1** |
| *verifyRetriesLeft[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked.<br>• Bit to check in ParamPresenceMask - **1** |
| *unblockRetriesLeft[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable.<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.665.2 Field Documentation

#### 8.665.2.1 uint8_t unpack_dms_UIMGetControlKeyStatus_t::facilityState

#### 8.665.2.2 swi_uint256_t unpack_dms_UIMGetControlKeyStatus_t::ParamPresenceMask

#### 8.665.2.3 uint16_t unpack_dms_UIMGetControlKeyStatus_t::Tlvresult

#### 8.665.2.4 uint8_t unpack_dms_UIMGetControlKeyStatus_t::unblockRetriesLeft

#### 8.665.2.5 uint8_t unpack_dms_UIMGetControlKeyStatus_t::verifyRetriesLeft

## 8.666 unpack_dms_UIMGetICCID_t Struct Reference

### Data Fields

- uint8_t stringSize
- uint8_t String [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.666.1 Detailed Description

This structure is used to store unpack_dms_UIMGetICCID parameters.

**Parameters**

| | |
|---|---|
| *stringSize* | • Size of String.<br>• Bit to check in ParamPresenceMask - **1** |
| *String* | • ICCID String.<br>• Maximum Length is 255 Bytes<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unack result. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.666.2 Field Documentation

#### 8.666.2.1 swi_uint256_t unpack_dms_UIMGetICCID_t::ParamPresenceMask

#### 8.666.2.2 uint8_t unpack_dms_UIMGetICCID_t::String[255]

#### 8.666.2.3 uint8_t unpack_dms_UIMGetICCID_t::stringSize

#### 8.666.2.4 uint16_t unpack_dms_UIMGetICCID_t::Tlvresult

## 8.667 unpack_dms_UIMGetPINStatus_t Struct Reference

### Data Fields

- uint8_t p1Status
- uint8_t p1VerifyRetriesLeft
- uint8_t p1UnblockRetriesLeft
- uint8_t p2Status
- uint8_t p2VerifyRetriesLeft
- uint8_t p2UnblockRetriesLeft
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.667.1 Detailed Description

This structure contains Get PIN Status unpack information

**Parameters**

| | |
|---|---|
| *p1Status[OUT]* | • PIN1 status(0xFFFFFFFF - Unknown)<br>– 0 - PIN not initialized<br>– 1 - PIN enabled, not verified<br>– 2 - PIN enabled, verified<br>– 3 - PIN disabled<br>– 4 - PIN blocked<br>– 5 - PIN permanently blocked<br>• Bit to check in ParamPresenceMask - **17** |
| *p1VerifyRetries-Left[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left of PIN1, after which PIN will be blocked.<br>• Bit to check in ParamPresenceMask - **17** |
| *p1Unblock-RetriesLeft[OUT]* | • Optional parameter<br>• 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.<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *p2Status[OUT]* | • PIN2 status(0xFFFFFFFF - Unknown)<br>– 0 - PIN not initialized<br>– 1 - PIN enabled, not verified<br>– 2 - PIN enabled, verified<br>– 3 - PIN disabled<br>– 4 - PIN blocked<br>– 5 - PIN permanently blocked<br>• Bit to check in ParamPresenceMask - **18** |
| *p2VerifyRetries-Left[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left of PIN2, after which PIN will be blocked.<br>• Bit to check in ParamPresenceMask - **18** |
| *p2Unblock-RetriesLeft[OUT]* | • Optional parameter<br>• 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.<br>• Bit to check in ParamPresenceMask - **18** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.667.2 Field Documentation

#### 8.667.2.1 uint8_t unpack_dms_UIMGetPINStatus_t::p1Status

#### 8.667.2.2 uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft

#### 8.667.2.3 uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft

#### 8.667.2.4 uint8_t unpack_dms_UIMGetPINStatus_t::p2Status

#### 8.667.2.5 uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft

#### 8.667.2.6 uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft

#### 8.667.2.7 swi_uint256_t unpack_dms_UIMGetPINStatus_t::ParamPresenceMask

#### 8.667.2.8 uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult

## 8.668 unpack_dms_UIMSetControlKeyProtection_t Struct Reference

### Data Fields

- uint8_t verifyRetriesLeft

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.668.1 Detailed Description

This structure contains UIM Set control key protection unpack information

**Parameters**

| | |
|---|---|
| *verifyRetries-Left[OUT]* | • Optional parameter<br>• number of retries left after which the control key is blocked<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.668.2 Field Documentation

#### 8.668.2.1 swi_uint256_t unpack_dms_UIMSetControlKeyProtection_t::ParamPresenceMask

#### 8.668.2.2 uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult

#### 8.668.2.3 uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft

## 8.669 unpack_dms_UIMSetPINProtection_t Struct Reference

### Data Fields

- uint8_t verifyRetriesLeft
- uint8_t unblockRetriesLeft
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.669.1 Detailed Description

This structure contains PIN retries status

**Parameters**

| | |
|---|---|
| *verifyRetries-Left[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of retries left, after which PIN will be blocked.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *unblockRetries-Left[OUT]* | • Optional parameter<br>• Upon operational failure, this will indicate number of unblock retries left, after which the PIN will be permanently blocked i.e. UIM is unusable.<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult[OUT]* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.669.2 Field Documentation

#### 8.669.2.1 swi_uint256_t unpack_dms_UIMSetPINProtection_t::ParamPresenceMask

#### 8.669.2.2 uint16_t unpack_dms_UIMSetPINProtection_t::Tlvresult

#### 8.669.2.3 uint8_t unpack_dms_UIMSetPINProtection_t::unblockRetriesLeft

#### 8.669.2.4 uint8_t unpack_dms_UIMSetPINProtection_t::verifyRetriesLeft

## 8.670 unpack_dms_UIMUnblockControlKey_t Struct Reference

### Data Fields

- uint8_t unblockRetriesLeft
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.670.1 Detailed Description

This structure contains UIM Set control key protection unpack information

**Parameters**

| | |
|---|---|
| *unblockRetries-Left[OUT]* | • Optional parameter<br>• number of unblock retries left after which the control key is permanently blocked<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult[OUT]* | • Unpack Result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.670.2 Field Documentation

#### 8.670.2.1 swi_uint256_t unpack_dms_UIMUnblockControlKey_t::ParamPresenceMask

#### 8.670.2.2 uint16_t unpack_dms_UIMUnblockControlKey_t::Tlvresult

#### 8.670.2.3 uint8_t unpack_dms_UIMUnblockControlKey_t::unblockRetriesLeft

## 8.671 unpack_fms_GetImagesPreference_t Struct Reference

### Data Fields

- uint32_t ImageListSize
- FMSPrefImageList * pImageList
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.671.1 Detailed Description

This structure contains the Get Image Preference information unpack

**Parameters**

| | |
|---|---|
| *listSize* | • The number of elements in the image list |
| *pListEntries* | • Array of Image entries with size provided by previous field<br>• See FMSImageElement<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack result |

### 8.671.2 Field Documentation

#### 8.671.2.1 uint32_t unpack_fms_GetImagesPreference_t::ImageListSize

#### 8.671.2.2 swi_uint256_t unpack_fms_GetImagesPreference_t::ParamPresenceMask

#### 8.671.2.3 FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList

#### 8.671.2.4 uint16_t unpack_fms_GetImagesPreference_t::Tlvresult

## 8.672 unpack_fms_GetStoredImages_t Struct Reference

### Data Fields

- uint32_t imagelistSize
- FMSImageList imageList
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.672.1 Detailed Description

This structure contains the Get Stored Images unpack

**Parameters**

| | |
|---|---|
| *listSize* | • The number of elements in the image list |
| *imageList* | • Array of Image entries with size provided by previous field<br>• See FMSImageElement<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack result |

### 8.672.2 Field Documentation

#### 8.672.2.1 FMSImageList unpack_fms_GetStoredImages_t::imageList

#### 8.672.2.2 uint32_t unpack_fms_GetStoredImages_t::imagelistSize

#### 8.672.2.3 swi_uint256_t unpack_fms_GetStoredImages_t::ParamPresenceMask

#### 8.672.2.4 uint16_t unpack_fms_GetStoredImages_t::Tlvresult

## 8.673 unpack_fms_SetImagesPreference_t Struct Reference

### Data Fields

- uint32_t ImageTypesSize
- uint8_t ImageTypes [255]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.673.1 Detailed Description

This structure contains the Set Images Preference unpack

**Parameters**

| | |
|---|---|
| *ImageTypesSize* | • Image Type Size |
| *ImageTypes* | • Image Type<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • Unpack result |

### 8.673.2 Field Documentation

#### 8.673.2.1 uint8_t unpack_fms_SetImagesPreference_t::ImageTypes[255]

#### 8.673.2.2 uint32_t unpack_fms_SetImagesPreference_t::ImageTypesSize

#### 8.673.2.3 swi_uint256_t unpack_fms_SetImagesPreference_t::ParamPresenceMask

#### 8.673.2.4 uint16_t unpack_fms_SetImagesPreference_t::Tlvresult

## 8.674 unpack_ims_SLQSGetIMSSMSConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- uint8_t * pSMSFormat
- uint8_t * pSMSOverIPNwInd
- uint8_t * pPhoneCtxtURILen
- uint8_t * pPhoneCtxtURI
- swi_uint256_t ParamPresenceMask

### 8.674.1 Detailed Description

This structure contains the SLQSGetIMSSMSConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response<br>• Bit to check in ParamPresenceMask - **16** |
| *pSMSFormat* | • SMS format<br>– 0 - 3GPP<br>– 1 - 3GPP2<br>• Bit to check in ParamPresenceMask - **17** |
| *pSMSOverIPNw-Ind* | • SMS over IP Network Indication Flag<br>– TRUE - Turn on mobile-originated SMS<br>– FALSE - Turn off mobile-originated SMS<br>• Bit to check in ParamPresenceMask - **18** |
| *pPhoneCtxtURI-Len[IN/OUT]* | • Size in bytes assigned to the Phone context Universal Resource Identifier to follow |
| *pPhoneCtxtURI* | • Phone context universal resource identifier<br>• Length of this string must be specified in pPhoneCtxtURILen parameter<br>• Bit to check in ParamPresenceMask - **19** |

### 8.674.2 Field Documentation

#### 8.674.2.1 swi_uint256_t unpack_ims_SLQSGetIMSSMSConfig_t::ParamPresenceMask

#### 8.674.2.2 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pPhoneCtxtURI

#### 8.674.2.3 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pPhoneCtxtURILen

#### 8.674.2.4 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pSettingResp

#### 8.674.2.5 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pSMSFormat

#### 8.674.2.6 uint8_t* unpack_ims_SLQSGetIMSSMSConfig_t::pSMSOverIPNwInd

## 8.675 unpack_ims_SLQSGetIMSUserConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- uint8_t * pIMSDomainLen
- uint8_t * pIMSDomain
- swi_uint256_t ParamPresenceMask

### 8.675.1 Detailed Description

This structure contains the SLQSGetIMSUserConfig pack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response<br>• Bit to check in ParamPresenceMask - **16** |
| *pIMSDomain-Len[IN/OUT]* | • Length of IMS Domain Name to follow |
| *pIMSDomain* | • IMS domain name<br>• Length of this string must be specified in pIMSDomainLen parameter<br>• Bit to check in ParamPresenceMask - **17** |

### 8.675.2 Field Documentation

#### 8.675.2.1 swi_uint256_t unpack_ims_SLQSGetIMSUserConfig_t::ParamPresenceMask

#### 8.675.2.2 uint8_t* unpack_ims_SLQSGetIMSUserConfig_t::pIMSDomain

#### 8.675.2.3 uint8_t* unpack_ims_SLQSGetIMSUserConfig_t::pIMSDomainLen

#### 8.675.2.4 uint8_t* unpack_ims_SLQSGetIMSUserConfig_t::pSettingResp

## 8.676 unpack_ims_SLQSGetIMSVoIPConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- uint16_t * pSessionExpiryTimer
- uint16_t * pMinSessionExpiryTimer
- uint8_t * pAmrWbEnable
- uint8_t * pScrAmrEnable
- uint8_t * pScrAmrWbEnable
- uint8_t * pAmrMode
- uint16_t * pAmrWBMode
- uint8_t * pAmrOctetAligned
- uint8_t * pAmrWBOctetAligned
- uint16_t * pRingingTimer
- uint16_t * pRingBackTimer
- uint16_t * pRTPRTCPInactTimer
- swi_uint256_t ParamPresenceMask

### 8.676.1 Detailed Description

This structure contains the SLQSGetIMSVoIPConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE<br>• Bit to check in ParamPresenceMask - **16** |
| *pSessionExpiry-Timer* | • Session duration, in seconds<br>• Bit to check in ParamPresenceMask - **17** |
| *pMinSession-ExpiryTimer* | • Minimum allowed value for session expiry timer, in seconds<br>• Bit to check in ParamPresenceMask - **18** |
| *pAmrWbEnable* | • Flag to enable/disable Adaptive Multirate Codec(AMR) WideBand(WB) audio<br>• Values:<br>– True - Enable<br>– False - Disable<br>• Bit to check in ParamPresenceMask - **19** |
| *pScrAmrEnable* | • Flag to enable/disable Source Control Rate(SCR) for AMR NarrowBand (NB)<br>• Values:<br>– True - Enable<br>– False - Disable<br>• Bit to check in ParamPresenceMask - **20** |

| | |
|---|---|
| *pScrAmrWb-Enable* | • Flag to enable/disable SCR for AMR WB Audio<br>• Values:<br>– True - Enable<br>– False - Disable<br>• Bit to check in ParamPresenceMask - **21** |
| *pAmrMode* | • BitMask for AMR NB modes allowed<br>• Values:<br>– 0x1 - 4.75 kbps<br>– 0x2 - 5.15 kbps<br>– 0x4 - 5.9 kbps<br>– 0x8 - 6.17 kbps<br>– 0x10 - 7.4 kbps<br>– 0x20 - 7.95 kbps<br>– 0x40 - 10.2 kbps<br>– 0x80 - 12.2 kbps<br>• Bit to check in ParamPresenceMask - **22** |
| *pAmrWBMode* | • BitMask for AMR WB modes allowed<br>• Values:<br>– 0x1 - 6.60 kbps<br>– 0x2 - 8.85 kbps<br>– 0x4 - 12.65 kbps<br>– 0x8 - 14.25 kbps<br>– 0x10 - 15.85 kbps<br>– 0x20 - 18.25 kbps<br>– 0x40 - 19.85 kbps<br>– 0x80 - 23.05 kbps<br>– 0x100 - 23.85 kbps<br>• Bit to check in ParamPresenceMask - **23** |
| *pAmrOctet-Aligned* | • Flag to indicate if the octet is aligned for AMR NB Audio<br>• Values:<br>– True - Aligned<br>– False - Not aligned, Bandwidth Efficient mode<br>• Bit to check in ParamPresenceMask - **24** |
| *pAmrWBOctet-Aligned* | • Flag to indicate if the octet is aligned for AMR WB Audio<br>• Values:<br>– True - Aligned<br>– False - Not aligned, Bandwidth Efficient mode<br>• Bit to check in ParamPresenceMask - **25** |

| | |
|---|---|
| *pRingingTimer* | • Duration of ringing timer, in seconds. The ringing timer starts on the ringing event. If the call is not answered within the duration of this timer, the call is disconnected.<br>• Bit to check in ParamPresenceMask - **26** |
| *pRingBackTimer* | • Duration of ringback timer, in seconds. The ringback timer starts on the ringback event. If the call is not answered within the duration of this timer, the call is disconnected.<br>• Bit to check in ParamPresenceMask - **27** |
| *pRTPRTCP-InactTimer* | • Duration of RTP/RTCP inactivity timer, in seconds. If no RTP/RTCP packet is received prior to the expiry of this timer, the call is disconnected.<br>• Bit to check in ParamPresenceMask - **28** |

### 8.676.2 Field Documentation

#### 8.676.2.1 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pAmrMode

#### 8.676.2.2 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pAmrOctetAligned

#### 8.676.2.3 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pAmrWbEnable

#### 8.676.2.4 uint16_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pAmrWBMode

#### 8.676.2.5 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pAmrWBOctetAligned

#### 8.676.2.6 swi_uint256_t unpack_ims_SLQSGetIMSVoIPConfig_t::ParamPresenceMask

#### 8.676.2.7 uint16_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pMinSessionExpiryTimer

#### 8.676.2.8 uint16_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pRingBackTimer

#### 8.676.2.9 uint16_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pRingingTimer

#### 8.676.2.10 uint16_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pRTPRTCPInactTimer

#### 8.676.2.11 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pScrAmrEnable

#### 8.676.2.12 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pScrAmrWbEnable

#### 8.676.2.13 uint16_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pSessionExpiryTimer

#### 8.676.2.14 uint8_t∗ unpack_ims_SLQSGetIMSVoIPConfig_t::pSettingResp

## 8.677 unpack_ims_SLQSGetRegMgrConfig_t Struct Reference

### Data Fields

- uint8_t ∗ pSettingResp
- uint16_t ∗ pPCSCFPort
- uint8_t ∗ pPriCSCFPortNameLen

- uint8_t * pPriCSCFPortName
- uint8_t * pIMSTestMode
- swi_uint256_t ParamPresenceMask

### 8.677.1 Detailed Description

This structure contains the SLQSGetRegMgrConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response<br>• Bit to check in ParamPresenceMask - **16** |
| *pPCSCFPort* | • Proxy call session control function port<br>• Bit to check in ParamPresenceMask - **17** |
| *pPriCSCFPort-NameLen(IN/O-UT)* | • Size in bytes assigned to the primary CSCF Port name parameter to follow |
| *pPriCSCFPort-Name* | • Call Session control port, fully qualified domain name<br>• Length of this string must be specified in pPriCSCFPortNameLen parameter<br>• Bit to check in ParamPresenceMask - **18** |
| *pIMSTestMode* | • IMS Test mode Enabled.<br>  – TRUE - Enabled<br>  – FALSE - Disabled<br>• Bit to check in ParamPresenceMask - **19** |

**Note**

pPriCSCFPortNameLen must be set to a valid value during API call to retrieve pPriCSCFPortName.

### 8.677.2 Field Documentation

#### 8.677.2.1 swi_uint256_t unpack_ims_SLQSGetRegMgrConfig_t::ParamPresenceMask

#### 8.677.2.2 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pIMSTestMode

#### 8.677.2.3 uint16_t* unpack_ims_SLQSGetRegMgrConfig_t::pPCSCFPort

#### 8.677.2.4 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pPriCSCFPortName

#### 8.677.2.5 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pPriCSCFPortNameLen

#### 8.677.2.6 uint8_t* unpack_ims_SLQSGetRegMgrConfig_t::pSettingResp

## 8.678 unpack_ims_SLQSGetSIPConfig_t Struct Reference

## Data Fields

- uint8_t * pSettingResp
- uint16_t * pSIPLocalPort
- uint32_t * pTimerSIPReg
- uint32_t * pSubscribeTimer
- uint32_t * pTimerT1
- uint32_t * pTimerT2
- uint32_t * pTimerTf
- uint8_t * pSigCompEnabled
- swi_uint256_t ParamPresenceMask

### 8.678.1 Detailed Description

This structure contains the SLQSGetSIPConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings Response<br>• Bit to check in ParamPresenceMask - **16** |
| *pSIPLocalPort* | • Primary call session control function SIP port number<br>• Bit to check in ParamPresenceMask - **17** |
| *pTimerSIPReg* | • Initial SIP registration duration from the User equipment, in seconds<br>• Bit to check in ParamPresenceMask - **18** |
| *pSubscribeTimer* | • Duration of the subscription by the UE for IMS registration notifications, in seconds<br>• Bit to check in ParamPresenceMask - **19** |
| *pTimerT1* | • RTT estimate, in milliseconds<br>• Bit to check in ParamPresenceMask - **20** |
| *pTimerT2* | • The maximum retransmit interval for non-invite requests and invite responses, in milliseconds<br>• Bit to check in ParamPresenceMask - **21** |
| *pTimerTf* | • Non-invite transaction timeout timer, in milliseconds<br>• Bit to check in ParamPresenceMask - **22** |
| *pSigComp-Enabled* | • Sig Comp Status<br>– TRUE - Sig Comp Enabled<br>– FALSE - Sig Comp Disabled<br>• Bit to check in ParamPresenceMask - **23** |

### 8.678.2 Field Documentation

#### 8.678.2.1 swi_uint256_t unpack_ims_SLQSGetSIPConfig_t::ParamPresenceMask

#### 8.678.2.2 uint8_t* unpack_ims_SLQSGetSIPConfig_t::pSettingResp

#### 8.678.2.3 uint8_t* unpack_ims_SLQSGetSIPConfig_t::pSigCompEnabled

#### 8.678.2.4 uint16_t* unpack_ims_SLQSGetSIPConfig_t::pSIPLocalPort

#### 8.678.2.5 uint32_t* unpack_ims_SLQSGetSIPConfig_t::pSubscribeTimer

#### 8.678.2.6 uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerSIPReg

#### 8.678.2.7 uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerT1

#### 8.678.2.8 uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerT2

#### 8.678.2.9 uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerTf

## 8.679 unpack_ims_SLQSRegMgrCfgCallBack_ind_t Struct Reference

### Data Fields

- ims_PCSCFPortInfo PCTlv
- ims_CSCFPortNameInfo PNTlv
- ims_IMSTestModeInfo TMTlv
- swi_uint256_t ParamPresenceMask

### 8.679.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

**Parameters**

| | |
|---|---|
| *PCTlv* | • Primary PCSCF Port info<br>• See ims_PCSCFPortInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *PNTlv* | • CSCF Port info<br>• see ims_CSCFPortNameInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *TMTlv* | • IMS test mode info<br>• see ims_IMSTestModeInfo for more information<br>• Bit to check in ParamPresenceMask - **18** |

**Note**

: None

### 8.679.2 Field Documentation

#### 8.679.2.1 swi_uint256_t unpack_ims_SLQSRegMgrCfgCallBack_ind_t::ParamPresenceMask

#### 8.679.2.2 ims_PCSCFPortInfo unpack_ims_SLQSRegMgrCfgCallBack_ind_t::PCTlv

#### 8.679.2.3 ims_CSCFPortNameInfo unpack_ims_SLQSRegMgrCfgCallBack_ind_t::PNTlv

#### 8.679.2.4 ims_IMSTestModeInfo unpack_ims_SLQSRegMgrCfgCallBack_ind_t::TMTlv

## 8.680 unpack_ims_SLQSSetIMSSMSConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- swi_uint256_t ParamPresenceMask

### 8.680.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE<br>• Bit to check in ParamPresenceMask - **16** |

### 8.680.2 Field Documentation

#### 8.680.2.1 swi_uint256_t unpack_ims_SLQSSetIMSSMSConfig_t::ParamPresenceMask

#### 8.680.2.2 uint8_t* unpack_ims_SLQSSetIMSSMSConfig_t::pSettingResp

## 8.681 unpack_ims_SLQSSetIMSUserConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- swi_uint256_t ParamPresenceMask

### 8.681.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE<br>• Bit to check in ParamPresenceMask - **16** |

### 8.681.2 Field Documentation

#### 8.681.2.1 swi_uint256_t unpack_ims_SLQSSetIMSUserConfig_t::ParamPresenceMask

#### 8.681.2.2 uint8_t* unpack_ims_SLQSSetIMSUserConfig_t::pSettingResp

## 8.682 unpack_ims_SLQSSetIMSVoIPConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- swi_uint256_t ParamPresenceMask

### 8.682.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE<br>• Bit to check in ParamPresenceMask - **16** |

### 8.682.2 Field Documentation

#### 8.682.2.1 swi_uint256_t unpack_ims_SLQSSetIMSVoIPConfig_t::ParamPresenceMask

#### 8.682.2.2 uint8_t* unpack_ims_SLQSSetIMSVoIPConfig_t::pSettingResp

## 8.683 unpack_ims_SLQSSetRegMgrConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- swi_uint256_t ParamPresenceMask

### 8.683.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE<br>• Bit to check in ParamPresenceMask - **16** |

### 8.683.2 Field Documentation

#### 8.683.2.1 swi_uint256_t unpack_ims_SLQSSetRegMgrConfig_t::ParamPresenceMask

#### 8.683.2.2 uint8_t* unpack_ims_SLQSSetRegMgrConfig_t::pSettingResp

## 8.684 unpack_ims_SLQSSetSIPConfig_t Struct Reference

### Data Fields

- uint8_t * pSettingResp
- swi_uint256_t ParamPresenceMask

### 8.684.1 Detailed Description

This structure contains the SLQSSetSIPConfig unpack parameters.

**Parameters**

| | |
|---|---|
| *pSettingResp* | • Settings standard response type. A settings specific error code is returned when the standard response error type is QMI_ERR_CAUSE_CODE<br>• Bit to check in ParamPresenceMask - **16** |

### 8.684.2 Field Documentation

#### 8.684.2.1 swi_uint256_t unpack_ims_SLQSSetSIPConfig_t::ParamPresenceMask

#### 8.684.2.2 uint8_t* unpack_ims_SLQSSetSIPConfig_t::pSettingResp

## 8.685 unpack_ims_SLQSSIPCfgCallBack_ind_t Struct Reference

### Data Fields

- ims_SIPPortInfo SPTlv
- ims_SIPRegnTmrInfo SRTlv
- ims_SubscrTmrInfo STTlv
- ims_TmrT1Info TT1Tlv
- ims_TmrT2Info TT2Tlv
- ims_TmrTfInfo TTfTlv
- ims_SigCompEnInfo SCTlv
- swi_uint256_t ParamPresenceMask

### 8.685.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

**Parameters**

| | |
|---|---|
| *SPTlv* | • SIP port tlv<br>• See ims_SIPPortInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *SRTlv* | • SIP Registration Timer tlv<br>• see ims_SIPRegnTmrInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *STTlv* | • Subscriber timer tlv<br>• see ims_SubscrTmrInfo for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *TT1Tlv* | • Timer T1 tlv<br>• see ims_TmrT1Info for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *TT2Tlv* | • Timer T2 tlv<br>• see ims_TmrT2Info for more information<br>• Bit to check in ParamPresenceMask - **20** |
| *TTfTlv* | • Timer Tf tlv<br>• see ims_TmrTfInfo for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *SCTlv* | • SigComp Status Tlv<br>• see ims_SigCompEnInfo for more information<br>• Bit to check in ParamPresenceMask - **22** |

**Note**

: None

## 8.685.2 Field Documentation

### 8.685.2.1 swi_uint256_t unpack_ims_SLQSSIPCfgCallBack_ind_t::ParamPresenceMask

### 8.685.2.2 ims_SigCompEnInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SCTlv

### 8.685.2.3 ims_SIPPortInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SPTlv

### 8.685.2.4 ims_SIPRegnTmrInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SRTlv

### 8.685.2.5 ims_SubscrTmrInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::STTlv

### 8.685.2.6 ims_TmrT1Info unpack_ims_SLQSSIPCfgCallBack_ind_t::TT1Tlv

### 8.685.2.7 ims_TmrT2Info unpack_ims_SLQSSIPCfgCallBack_ind_t::TT2Tlv

### 8.685.2.8 ims_TmrTfInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::TTfTlv

## 8.686 unpack_ims_SLQSSMSCfgCallBack_ind_t Struct Reference

### Data Fields

- ims_SMSFmtInfo SFTlv
- ims_SMSoIPNwInfo SINTlv
- ims_PhCtxtURIInfo PCURTlv
- swi_uint256_t ParamPresenceMask

### 8.686.1 Detailed Description

Structure used to store SMS Config Indication Parameters.

**Parameters**

| | |
|---|---|
| *SFTlv* | • SMS format info<br>• See ims_SMSFmtInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *SINTlv* | • SMS over IP network info<br>• see ims_SMSoIPNwInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *PCURTlv* | • Phone context URI info<br>• see ims_PhCtxtURIInfo for more information<br>• Bit to check in ParamPresenceMask - **18** |

**Note**

: None

### 8.686.2 Field Documentation

#### 8.686.2.1 swi_uint256_t unpack_ims_SLQSSMSCfgCallBack_ind_t::ParamPresenceMask

#### 8.686.2.2 ims_PhCtxtURIInfo unpack_ims_SLQSSMSCfgCallBack_ind_t::PCURTlv

#### 8.686.2.3 ims_SMSFmtInfo unpack_ims_SLQSSMSCfgCallBack_ind_t::SFTlv

#### 8.686.2.4 ims_SMSoIPNwInfo unpack_ims_SLQSSMSCfgCallBack_ind_t::SINTlv

## 8.687 unpack_ims_SLQSUserCfgCallBack_ind_t Struct Reference

### Data Fields

- ims_IMSDomainInfo IDTlv
- swi_uint256_t ParamPresenceMask

### 8.687.1 Detailed Description

Structure used to store SMS Config Indication Parameters.

**Parameters**

| | |
|---|---|
| *IDTlv* | • IMS domain info<br>• See ims_IMSDomainInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |

**Note**

: None

### 8.687.2 Field Documentation

#### 8.687.2.1 ims_IMSDomainInfo unpack_ims_SLQSUserCfgCallBack_ind_t::IDTlv

#### 8.687.2.2 swi_uint256_t unpack_ims_SLQSUserCfgCallBack_ind_t::ParamPresenceMask

## 8.688 unpack_ims_SLQSVoIPCfgCallBack_ind_t Struct Reference

### Data Fields

- ims_SessDurInfo SDTlv
- ims_MinSessExpInfo MSETlv
- ims_EnabAMRWBInfo EAWTlv
- ims_EnabSCRAMRInfo ESATlv
- ims_EnabSCRAMRWBInfo ESAWTlv
- ims_AMRModeInfo AMTlv
- ims_AMRWBModeInfo AWMTlv
- ims_AMROctAlgnInfo AOATlv
- ims_AMRWBOctAlgnInfo AWOATlv
- ims_RngTmrInfo RTTlv
- ims_RngBkTmrInfo RBTTlv
- ims_RTPRTCPInactTmrDurInfo RTIDTlv
- swi_uint256_t ParamPresenceMask

### 8.688.1 Detailed Description

Structure used to store VOIP Config Indication Parameters.

**Parameters**

| | |
|---|---|
| *SDTlv* | • Session Duration tlv<br>• See ims_SessDurInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *MSETlv* | • Minimum Session Timer tlv<br>• see ims_MinSessExpInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *EAWTlv* | • Enable AMR WB tlv<br>• see ims_EnabAMRWBInfo for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *ESATlv* | • Enable SCR AMR NB tlv<br>• see ims_EnabSCRAMRInfo for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *ESAWTlv* | • Enable SCR AMR WB tlv<br>• see ims_EnabSCRAMRWBInfo for more information<br>• Bit to check in ParamPresenceMask - **20** |
| *AMTlv* | • AMR NB Mode tlv<br>• see ims_AMRModeInfo for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *AWMTlv* | • AMR WB Mode Tlv<br>• see ims_AMRWBModeInfo for more information<br>• Bit to check in ParamPresenceMask - **22** |
| *AOATlv* | • AMR NB Octet Aligned tlv<br>• see ims_AMROctAlgnInfo for more information<br>• Bit to check in ParamPresenceMask - **23** |
| *AWOATlv* | • AMR WB Octet Aligned tlv<br>• see ims_AMRWBOctAlgnInfo for more information<br>• Bit to check in ParamPresenceMask - **24** |
| *RTTlv* | • Ringing Timer Duration tlv<br>• see ims_RngTmrInfo for more information<br>• Bit to check in ParamPresenceMask - **25** |
| *RBTTlv* | • Ringback Duration tlv<br>• see ims_RngBkTmrInfo for more information<br>• Bit to check in ParamPresenceMask - **26** |

| | |
|---|---|
| *RTIDTlv* | • RTP/RTCP Inactivity Timer Duration Tlv<br>• see ims_RTPRTCPInactTmrDurInfo for more information<br>• Bit to check in ParamPresenceMask - **27** |

**Note**

: None

### 8.688.2 Field Documentation

#### 8.688.2.1 ims_AMRModeInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::AMTlv

#### 8.688.2.2 ims_AMROctAlgnInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::AOATlv

#### 8.688.2.3 ims_AMRWBModeInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::AWMTlv

#### 8.688.2.4 ims_AMRWBOctAlgnInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::AWOATlv

#### 8.688.2.5 ims_EnabAMRWBInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::EAWTlv

#### 8.688.2.6 ims_EnabSCRAMRInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::ESATlv

#### 8.688.2.7 ims_EnabSCRAMRWBInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::ESAWTlv

#### 8.688.2.8 ims_MinSessExpInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::MSETlv

#### 8.688.2.9 swi_uint256_t unpack_ims_SLQSVoIPCfgCallBack_ind_t::ParamPresenceMask

#### 8.688.2.10 ims_RngBkTmrInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::RBTTlv

#### 8.688.2.11 ims_RTPRTCPInactTmrDurInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::RTIDTlv

#### 8.688.2.12 ims_RngTmrInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::RTTlv

#### 8.688.2.13 ims_SessDurInfo unpack_ims_SLQSVoIPCfgCallBack_ind_t::SDTlv

## 8.689 unpack_imsa_SLQSGetIMSARegStatus_t Struct Reference

### Data Fields

- uint8_t ImsRegStatus
- uint16_t ImsRegErrCode
- uint32_t NewImsRegStatus
- swi_uint256_t ParamPresenceMask

### 8.689.1 Detailed Description

This structure contains response parameters of registration status.

**Parameters**

| | |
|---|---|
| *ImsRegStatus* | • IMS Registration Status (Deprecated).<br>• Values<br>– TRUE - UE is registered on the IMS network<br>– FALSE - UE is not registered on the IMS network<br>• Bit to check in ParamPresenceMask - **16** |
| *ImsRegErrCode* | • IMS Registration Error Code.<br>• An error code is returned when the IMS registration status is IMSA_STATUS_NOT_REGISTERED. -Values<br>– 3xx - Redirection responses<br>– 4xx - Client failure responses<br>– 5xx - Server failure responses<br>– 6xx - Global failure responses<br>• Bit to check in ParamPresenceMask - **17** |
| *NewImsRegStatus* | • New IMS Registration Status<br>• Values<br>– 0 - Not registered for IMS<br>– 1 - Registering for IMS<br>– 2 - Registered for IMS<br>• Bit to check in ParamPresenceMask - **18** |

### 8.689.2 Field Documentation

#### 8.689.2.1 uint16_t unpack_imsa_SLQSGetIMSARegStatus_t::ImsRegErrCode

#### 8.689.2.2 uint8_t unpack_imsa_SLQSGetIMSARegStatus_t::ImsRegStatus

#### 8.689.2.3 uint32_t unpack_imsa_SLQSGetIMSARegStatus_t::NewImsRegStatus

#### 8.689.2.4 swi_uint256_t unpack_imsa_SLQSGetIMSARegStatus_t::ParamPresenceMask

## 8.690 unpack_imsa_SLQSGetIMSAServiceStatus_t Struct Reference

### Data Fields

- uint32_t SmsServiceStatus
- uint32_t VoipServiceStatus
- uint32_t VtServiceStatus
- uint32_t SmsServiceRat
- uint32_t VoipServiceRat
- uint32_t VtServiceRat
- uint32_t UtServiceStatus
- uint32_t UtServiceRat
- uint32_t VsServiceStatus
- uint32_t VsServiceRat
- swi_uint256_t ParamPresenceMask

### 8.690.1 Detailed Description

This structure contains response parameters of service status for various IMS services.

**Parameters**

| | |
|---|---|
| *SmsService-Status* | • SMS Service Status.<br>• Values<br>– 0 - IMS SMS service is not available<br>– 1 - IMS SMS is in limited service<br>– 2 - IMS SMS is in full service<br>• Bit to check in ParamPresenceMask - **16** |
| *VoipSeriveStatus* | • VoIP Service Status.<br>• Values<br>– 0 - IMS VoIP service is not available<br>– 2 - IMS VoIP is in full service<br>• Bit to check in ParamPresenceMask - **17** |
| *VtServiceStatus* | • VT Service Status<br>• Values<br>– 0 - IMS VT service is not available<br>– 2 - IMS VT is in full service<br>• Bit to check in ParamPresenceMask - **18** |
| *SmsServiceRat* | • SMS service RAT<br>• Values<br>– 0 - IMS service is registered on WLAN<br>– 1 - IMS service is registered on WWAN<br>– 2 - IMS service is registered on interworking WLAN<br>• Bit to check in ParamPresenceMask - **19** |
| *VoipServiceRat* | • VoIP service RAT.<br>• Values<br>– 0 - IMS service is registered on WLAN<br>– 1 - IMS service is registered on WWAN<br>– 2 - IMS service is registered on interworking WLAN<br>• Bit to check in ParamPresenceMask - **20** |
| *VtServiceRat* | • VT service RAT.<br>• Values<br>– 0 - IMS service is registered on WLAN<br>– 1 - IMS service is registered on WWAN<br>– 2 - IMS service is registered on interworking WLAN<br>• Bit to check in ParamPresenceMask - **21** |

| | |
|---|---|
| *UtServiceStatus* | • UT service Status.<br>• Values<br>– 0 - IMS UT service is not available<br>– 2 - IMS UT is in full service<br>• Bit to check in ParamPresenceMask - **22** |
| *UtServiceRat* | • UT service RAT.<br>• Values<br>– 0 - IMS service is registered on WLAN<br>– 1 - IMS service is registered on WWAN<br>– 2 - IMS service is registered on interworking WLAN<br>• Bit to check in ParamPresenceMask - **23** |
| *VsServiceStatus* | • VS service Status.<br>• Values<br>– 0 - IMS UT service is not available<br>– 2 - IMS UT is in full service<br>• Bit to check in ParamPresenceMask - **24** |
| *VsServiceRat* | • VS service RAT.<br>• Values<br>– 0 - IMS service is registered on WLAN<br>– 1 - IMS service is registered on WWAN<br>– 2 - IMS service is registered on interworking WLAN<br>• Bit to check in ParamPresenceMask - **25** |

## 8.690.2 Field Documentation

### 8.690.2.1 swi_uint256_t unpack_imsa_SLQSGetIMSAServiceStatus_t::ParamPresenceMask

### 8.690.2.2 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::SmsServiceRat

### 8.690.2.3 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::SmsServiceStatus

### 8.690.2.4 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::UtServiceRat

### 8.690.2.5 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::UtServiceStatus

### 8.690.2.6 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VoipServiceRat

### 8.690.2.7 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VoipServiceStatus

### 8.690.2.8 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VsServiceRat

### 8.690.2.9 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VsServiceStatus

#### 8.690.2.10 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VtServiceRat

#### 8.690.2.11 uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VtServiceStatus

## 8.691 unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t Struct Reference

### Data Fields

- uint8_t PdpConnState
- imsa_IMSFailErrCodeTlv FailErrCode
- swi_uint256_t ParamPresenceMask

### 8.691.1 Detailed Description

Structure used to store IMSA PDP status indication Parameters.

**Parameters**

| | |
|---|---|
| *PdpConnState* | • Values:<br>– 0- not connected<br>– 1- connected<br>• Bit to check in ParamPresenceMask - **1** |
| *FailErrCode* | • Fail error code when PDP is not connected<br>• See imsa_IMSFailErrCodeTlv for more information<br>• Bit to check in ParamPresenceMask - **16** |

**Note**

: None

### 8.691.2 Field Documentation

#### 8.691.2.1 imsa_IMSFailErrCodeTlv unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t::FailErrCode

#### 8.691.2.2 swi_uint256_t unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t::ParamPresenceMask

#### 8.691.2.3 uint8_t unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t::PdpConnState

## 8.692 unpack_imsa_SLQSImsaRatStatusCallBack_ind_t Struct Reference

### Data Fields

- imsa_RatHandoverStatusInfo RatHandover
- swi_uint256_t ParamPresenceMask

### 8.692.1 Detailed Description

Structure used to store IMSA RAT status indication Parameters.

**Parameters**

| | |
|---|---|
| *RatHandover* | • RAT handover status info<br>• See imsa_RatHandoverStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |

**Note**

: None

### 8.692.2 Field Documentation

#### 8.692.2.1 swi_uint256_t unpack_imsa_SLQSImsaRatStatusCallBack_ind_t::ParamPresenceMask

#### 8.692.2.2 imsa_RatHandoverStatusInfo unpack_imsa_SLQSImsaRatStatusCallBack_ind_t::RatHandover

## 8.693 unpack_imsa_SLQSImsaRegStatusCallBack_ind_t Struct Reference

### Data Fields

- imsa_IMSRegStatusInfo IMSRegistration
- imsa_IMSRegStatusErrorCodeInfo IMSRegistrationError
- imsa_NewIMSRegStatusInfo NewIMSRegistration
- swi_uint256_t ParamPresenceMask

### 8.693.1 Detailed Description

Structure used to store IMSA registration status indication Parameters.

**Parameters**

| | |
|---|---|
| *IMSRegistration* | • IMS registration info<br>• See imsa_IMSRegStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **1** |
| *IMSRegistration-Error* | • IMS registration error<br>• see imsa_IMSRegStatusErrorCodeInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *NewIMS-Registration* | • New IMS registration info<br>• see imsa_NewIMSRegStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |

**Note**

: None

### 8.693.2 Field Documentation

#### 8.693.2.1 imsa_IMSRegStatusInfo unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::IMSRegistration

#### 8.693.2.2 imsa_IMSRegStatusErrorCodeInfo unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::IMSRegistrationError

#### 8.693.2.3 imsa_NewIMSRegStatusInfo unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::NewIMSRegistration

#### 8.693.2.4 swi_uint256_t unpack_imsa_SLQSImsaRegStatusCallBack_ind_t::ParamPresenceMask

## 8.694 unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t Struct Reference

### Data Fields

- imsa_SmsSvcStatusInfo SmsService
- imsa_VoipSvcStatusInfo VoipService
- imsa_VtSvcStatusInfo VtService
- imsa_SmsRatInfo SmsRat
- imsa_VoipRatInfo VoipRat
- imsa_VtRatInfo VtRat
- imsa_UtSvcStatusInfo UtService
- imsa_UtRatInfo UtRat
- swi_uint256_t ParamPresenceMask

### 8.694.1 Detailed Description

Structure used to store IMSA service status indication Parameters.

**Parameters**

| | |
|---|---|
| *SmsService* | • SMS service info<br>• See imsa_SmsSvcStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *VoipService* | • VOIP service info<br>• see imsa_VoipSvcStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *VtService* | • VT service info<br>• see imsa_VtSvcStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *SmsRat* | • SMS RAT info<br>• see imsa_SmsRatInfo for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *VoipRat* | • VOIP RAT info<br>• see imsa_VoipRatInfo for more information<br>• Bit to check in ParamPresenceMask - **20** |

| | |
|---|---|
| *VtRat* | • VT RAT info<br>• see imsa_VtRatInfo for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *UtService* | • UT service info<br>• see imsa_UtSvcStatusInfo for more information<br>• Bit to check in ParamPresenceMask - **22** |
| *UtRat* | • UT RAT info<br>• see imsa_UtRatInfo for more information<br>• Bit to check in ParamPresenceMask - **23** |

**Note**

: None

### 8.694.2 Field Documentation

#### 8.694.2.1 swi_uint256_t unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::ParamPresenceMask

#### 8.694.2.2 imsa_SmsRatInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::SmsRat

#### 8.694.2.3 imsa_SmsSvcStatusInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::SmsService

#### 8.694.2.4 imsa_UtRatInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::UtRat

#### 8.694.2.5 imsa_UtSvcStatusInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::UtService

#### 8.694.2.6 imsa_VoipRatInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VoipRat

#### 8.694.2.7 imsa_VoipSvcStatusInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VoipService

#### 8.694.2.8 imsa_VtRatInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VtRat

#### 8.694.2.9 imsa_VtSvcStatusInfo unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VtService

## 8.695 unpack_loc_BestAvailPos_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint32_t * pXid
- uint64_t * pLatitude
- uint64_t * pLongitude
- uint32_t * pHorUncCircular
- uint32_t * pAltitudeWrtEllipsoid
- uint32_t * pVertUnc
- uint64_t * pTimestampUtc
- uint32_t * pTimeUnc

- uint32_t * pHorUncEllipseSemiMinor
- uint32_t * pHorUncEllipseSemiMajor
- uint32_t * pHorUncEllipseOrientAzimuth
- uint8_t * pHorCirConf
- uint8_t * pHorEllpConf
- uint32_t * pHorReliability
- uint32_t * pSpeedHorizontal
- uint32_t * pSpeedUnc
- uint32_t * pAltitudeWrtMeanSeaLevel
- uint8_t * pVertConfidence
- uint32_t * pVertReliability
- uint32_t * pSpeedVertical
- uint32_t * pSpeedVerticalUnc
- uint32_t * pHeading
- uint32_t * pHeadingUnc
- uint32_t * pMagneticDeviation
- uint32_t * pTechnologyMask
- loc_precisionDilution * pPrecisionDilution
- loc_gpsTime * pGpsTime
- uint32_t * pTimeSrc
- loc_sensorDataUsage * pSensorDataUsage
- loc_svUsedforFix * pSvUsedforFix
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.695.1 Detailed Description

This structure contains Best Available Position

**Parameters**

| | |
|---|---|
| *status* | • Valid values:<br>– eQMI_LOC_SUCCESS (0) - Request was completed successfully<br>– eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure<br>– eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported<br>– eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters<br>– eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy<br>– eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline<br>– eQMI_LOC_TIMEOUT (6) - Request failed because it timed out<br>– eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested<br>– eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request<br>– eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed<br>– eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |
| *xid* | Transaction ID that was specified in the Get Best Available Position request.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pLatitude* | • Latitude (specified in WGS84 datum)<br>• Type - Floating point<br>• Units - Degrees<br>• Range - -90.0 to 90.0<br>• Positive values indicate northern latitude<br>• Negative values indicate southern latitude<br>• Bit to check in ParamPresenceMask - **17** |
| *pLongitude* | • Longitude (specified in WGS84 datum)<br>• Type - Floating point<br>• Units - Degrees<br>• Range - -180.0 to 180.0<br>• Positive values indicate eastern latitude<br>• Negative values indicate western latitude<br>• Bit to check in ParamPresenceMask - **18** |
| *pHorUncCircular* | • Horizontal position uncertainty.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **19** |
| *pAltitudeWrt-Ellipsoid* | • Altitude With Respect to WGS84 Ellipsoid.<br>• Units - Meters<br>• Range -500 to 15883<br>• Bit to check in ParamPresenceMask - **20** |
| *pVertUnc* | • Vertical uncertainty.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **21** |
| *pTimestampUtc* | • UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970<br>• Bit to check in ParamPresenceMask - **22** |
| *pTimeUnc* | • Time uncertainty.<br>• Units - Milliseconds<br>• Bit to check in ParamPresenceMask - **23** |
| *pHorUncEllipse-SemiMinor* | • Semi-minor axis of horizontal elliptical uncertainty.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **24** |

| | |
|---|---|
| *pHorUncEllipse-SemiMajor* | • Semi-major axis of horizontal elliptical uncertainty.<br>• Units: Meters<br>• Bit to check in ParamPresenceMask - **25** |
| *pHorUncEllipse-OrientAzimuth* | • Elliptical horizontal uncertainty azimuth of orientation.<br>• Units - Decimal degrees<br>• Range - 0 to 180<br>• Bit to check in ParamPresenceMask - **26** |
| *pHorCirConf* | • Horizontal circular uncertainty confidence<br>• Units: Precent<br>• Range: 0 to 99<br>• Bit to check in ParamPresenceMask - **27** |
| *pHorEllpConf* | • Horizontal elliptical uncertainty confidence<br>• Units: Precent<br>• Range: 0 to 99<br>• Bit to check in ParamPresenceMask - **28** |
| *pHorReliability* | • Values<br>– 0 - Location reliability is not set.<br>– 1 - Location reliability is very low; use it at your own risk<br>– 2 - Location reliability is low; little or no cross-checking is possible.<br>– 3 - Location reliability is medium; limited cross-check passed<br>– 4 - Location reliability is high; strong cross-check passed<br>• Bit to check in ParamPresenceMask - **29** |
| *pSpeed-Horizontal* | • Horizontal speed.<br>• Units - Meters/second<br>• Bit to check in ParamPresenceMask - **30** |
| *pSpeedUnc* | • 3-D Speed uncertainty.<br>• Units - Meters/second.<br>• Bit to check in ParamPresenceMask - **31** |
| *pAltitudeWrt-MeanSeaLevel* | • Altitude With Respect to Sea Level.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **32** |

| | |
|---|---|
| *pVertConfidence* | • Vertical uncertainty confidence.<br>• Units - Percentage<br>• Range 0 to 99<br>• Bit to check in ParamPresenceMask - **33** |
| *pVertReliability* | • Values<br>– 0 - Location reliability is not set.<br>– 1 - Location reliability is very low; use it at your own risk.<br>– 2 - Location reliability is low; little or no cross-checking is possible<br>– 3 - Location reliability is medium; limited cross-check passed<br>– 4 - Location reliability is high; strong cross-check passed<br>• Bit to check in ParamPresenceMask - **34** |
| *pSpeedVertical* | • Vertical speed.<br>• Units - Meters/second<br>• Bit to check in ParamPresenceMask - **35** |
| *pSpeedVertical-Unc* | • Veritical speed<br>• Units: Meters/second<br>• Bit to check in ParamPresenceMask - **36** |
| *pHeading* | • Heading.<br>• Units - Degree<br>• Range 0 to 359.999<br>• Bit to check in ParamPresenceMask - **37** |
| *pHeadingUnc* | • Heading uncertainty.<br>• Units - Degree<br>• Range 0 to 359.999<br>• Bit to check in ParamPresenceMask - **38** |
| *pMagnetic-Deviation* | • 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.<br>• Bit to check in ParamPresenceMask - **39** |

| | |
|---|---|
| *pTechnology-Mask* | • Values<br>– 0x00000001 - Satellites were used to generate the fix<br>– 0x00000002 - Cell towers were used to generate the fix<br>– 0x00000004 - Wi-Fi access points were used to generate the fix<br>– 0x00000008 - Sensors were used to generate the fix<br>– 0x00000010 - Reference Location was used to generate the fix<br>– 0x00000020 - Coarse position injected into the location engine was used to generate the fix<br>– 0x00000040 - AFLT was used to generate the fix<br>– 0x00000080 - GNSS and network-provided measurements were used to generate the fix<br>• Bit to check in ParamPresenceMask - **40** |
| *-pPrecision-Dilution* | • See loc_precisionDilution for more information<br>• Bit to check in ParamPresenceMask - **41** |
| *-pGpsTime* | • See loc_gpsTime for more information<br>• Bit to check in ParamPresenceMask - **42** |
| *pTimeSrc* | • Values<br>– 0 - Invalid time.<br>– 1 - Time is set by the 1X system.<br>– 2 - Time is set by WCDMA/GSM time tagging.<br>– 3 - Time is set by an external injection.<br>– 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite.<br>– 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites.<br>– 6 - Both time of the week and the GPS week number are known.<br>– 7 - Time is set by the position engine after the fix is obtained<br>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.<br>– 9 - Time is set after decoding GLO satellites.<br>– 10- Time is set after transforming the GPS to GLO time<br>– 11- Time is set by the sleep time tag provided by the WCDMA network.<br>– 12- Time is set by the sleep time tag provided by the GSM network<br>– 13- Source of the time is unknown<br>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state<br>– 15- Time is set after decoding QZSS satellites.<br>– 16- Time is set after decoding BDS satellites.<br>• Bit to check in ParamPresenceMask - **43** |
| *-pSensorData-Usage* | • See loc_sensorDataUsage for more information<br>• Bit to check in ParamPresenceMask - **44** |
| *-pSvUsedforFix* | • See loc_svUsedforFix for more information<br>• Bit to check in ParamPresenceMask - **45** |

### 8.695.2 Field Documentation

8.695.2.1 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid

8.695.2.2 uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel

8.695.2.3 swi_uint256_t unpack_loc_BestAvailPos_Ind_t::ParamPresenceMask

8.695.2.4 loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime

8.695.2.5 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading

8.695.2.6 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc

8.695.2.7 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf

8.695.2.8 uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf

8.695.2.9 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability

8.695.2.10 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular

8.695.2.11 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth

8.695.2.12 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor

8.695.2.13 uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor

8.695.2.14 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude

8.695.2.15 uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude

8.695.2.16 uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation

8.695.2.17 loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution

8.695.2.18 loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage

8.695.2.19 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal

8.695.2.20 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc

8.695.2.21 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical

8.695.2.22 uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc

8.695.2.23 loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix

8.695.2.24 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask

8.695.2.25 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc

8.695.2.26 uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc

8.695.2.27 uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc

#### 8.695.2.28 uint8_t* unpack_loc_BestAvailPos_Ind_t::pVertConfidence

#### 8.695.2.29 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertReliability

#### 8.695.2.30 uint32_t* unpack_loc_BestAvailPos_Ind_t::pVertUnc

#### 8.695.2.31 uint32_t* unpack_loc_BestAvailPos_Ind_t::pXid

#### 8.695.2.32 uint32_t unpack_loc_BestAvailPos_Ind_t::status

#### 8.695.2.33 uint16_t unpack_loc_BestAvailPos_Ind_t::Tlvresult

## 8.696 unpack_loc_CradleMountCallback_Ind_t Struct Reference

### Data Fields

- uint32_t cradleMountConfigStatus
- swi_uint256_t ParamPresenceMask

### 8.696.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

**Parameters**

| | |
|---|---|
| *cradleMount-ConfigStatus* | • Values<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure.<br>– 2 - Request failed because it is not supported.<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - engine could not allocate sufficient memory<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 -Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |

### 8.696.2 Field Documentation

#### 8.696.2.1 uint32_t unpack_loc_CradleMountCallback_Ind_t::cradleMountConfigStatus

#### 8.696.2.2 swi_uint256_t unpack_loc_CradleMountCallback_Ind_t::ParamPresenceMask

## 8.697 unpack_loc_Delete_Assist_Data_t Struct Reference

### Data Fields

- uint16_t Tlvresult

- swi_uint256_t ParamPresenceMask

### 8.697.1 Detailed Description

This structure contains LOC delete assist data unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.697.2 Field Documentation

#### 8.697.2.1 swi_uint256_t unpack_loc_Delete_Assist_Data_t::ParamPresenceMask

#### 8.697.2.2 uint16_t unpack_loc_Delete_Assist_Data_t::Tlvresult

## 8.698 unpack_loc_DeleteAssistData_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.698.1 Detailed Description

Contain the parameters passed for SetLocDeleteAssistDataCallback by the device.

**Parameters**

| | |
|---|---|
| *status* | • Status of the Delete Assist Data request<br>• Valid values:<br>– eQMI_LOC_SUCCESS (0) - Request was completed successfully<br>– eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure<br>– eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported<br>– eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters<br>– eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy<br>– eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline<br>– eQMI_LOC_TIMEOUT (6) - Request failed because it timed out<br>– eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested<br>– eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request<br>– eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed<br>– eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack result |

### 8.698.2 Field Documentation

#### 8.698.2.1 swi_uint256_t unpack_loc_DeleteAssistData_Ind_t::ParamPresenceMask

#### 8.698.2.2 uint32_t unpack_loc_DeleteAssistData_Ind_t::status

#### 8.698.2.3 uint16_t unpack_loc_DeleteAssistData_Ind_t::Tlvresult

## 8.699 unpack_loc_EngineState_Ind_t Struct Reference

### Data Fields

- uint32_t engineState
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.699.1 Detailed Description

This structure contains LOC Engine State field.

**Parameters**

| | |
|---|---|
| *engineState* | • Location engine state.<br>• Valid values<br>– 1 - Location engine is on<br>– 2 - Loction engine is off<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack result |

### 8.699.2 Field Documentation

#### 8.699.2.1 uint32_t unpack_loc_EngineState_Ind_t::engineState

#### 8.699.2.2 swi_uint256_t unpack_loc_EngineState_Ind_t::ParamPresenceMask

#### 8.699.2.3 uint16_t unpack_loc_EngineState_Ind_t::Tlvresult

## 8.700 unpack_loc_EventNMEA_Ind_t Struct Reference

### Data Fields

- char NMEAData [201]
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.700.1 Detailed Description

Contain the parameters passed for SetLocEventMaskNMEACallback by the device.

**Parameters**

| | |
|---|---|
| *NMEAData* | • NMEA string<br>• Type - NULL terminated string<br>• Maximum string length (including NULL terminator) - 201<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack result |

### 8.700.2 Field Documentation

#### 8.700.2.1 char unpack_loc_EventNMEA_Ind_t::NMEAData[201]

#### 8.700.2.2 swi_uint256_t unpack_loc_EventNMEA_Ind_t::ParamPresenceMask

#### 8.700.2.3 uint16_t unpack_loc_EventNMEA_Ind_t::Tlvresult

## 8.701 unpack_loc_EventRegister_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.701.1 Detailed Description

This structure contains Event Register unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.701.2 Field Documentation

#### 8.701.2.1 swi_uint256_t unpack_loc_EventRegister_t::ParamPresenceMask

#### 8.701.2.2 uint16_t unpack_loc_EventRegister_t::Tlvresult

## 8.702 unpack_loc_EventTimeSyncCallback_Ind_t Struct Reference

### Data Fields

- uint32_t timeSyncRefCounter
- swi_uint256_t ParamPresenceMask

### 8.702.1 Detailed Description

This structure contains LOC Event Time Sync Reference COunter

**Parameters**

| | |
|---|---|
| *timeSyncRef-Counter* | • Sent by the location engine when it needs to synchronize location engine and control point (sensor processor) times.<br>• Bit to check in ParamPresenceMask - **1** |

### 8.702.2 Field Documentation

#### 8.702.2.1 swi_uint256_t unpack_loc_EventTimeSyncCallback_Ind_t::ParamPresenceMask

#### 8.702.2.2 uint32_t unpack_loc_EventTimeSyncCallback_Ind_t::timeSyncRefCounter

## 8.703 unpack_loc_GetServer_Ind_t Struct Reference

### Data Fields

- uint32_t serverStatus
- uint32_t serverType
- loc_IPv4Info * pIPv4AddrInfo
- loc_IPv6Info * pIPv6AddrInfo
- loc_urlAddr * pURL
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.703.1 Detailed Description

Contain the parameters passed for SetLocGetServerCallback by the device.

**Parameters**

| | |
|---|---|
| *serverStatus* | • Status of the Get Server request.<br>• Valid values<br>– eQMI_LOC_SUCCESS (0) - Request was completed successfully<br>– eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure<br>– eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported<br>– eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters<br>– eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy<br>– eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline<br>– eQMI_LOC_TIMEOUT (6) - Request failed because it timed out<br>– eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested<br>– eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request<br>– eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed<br>– eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |
| *serverType* | • Type of server.<br>• Valid values<br>– eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE<br>– eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC<br>– eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP<br>– eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) - Server type is custom PDE<br>• Bit to check in ParamPresenceMask - **2** |
| *pIPv4AddrInfo* | • See loc_IPv4Info for more information.<br>• Bit to check in ParamPresenceMask - **16** |
| *pIPv6AddrInfo* | • See loc_IPv6Info for more information.<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *pURL* | • See loc_urlAddr for more information.<br>• Bit to check in ParamPresenceMask - **18** |

### 8.703.2 Field Documentation

#### 8.703.2.1 swi_uint256_t unpack_loc_GetServer_Ind_t::ParamPresenceMask

#### 8.703.2.2 loc_IPv4Info* unpack_loc_GetServer_Ind_t::pIPv4AddrInfo

#### 8.703.2.3 loc_IPv6Info* unpack_loc_GetServer_Ind_t::pIPv6AddrInfo

#### 8.703.2.4 loc_urlAddr* unpack_loc_GetServer_Ind_t::pURL

#### 8.703.2.5 uint32_t unpack_loc_GetServer_Ind_t::serverStatus

#### 8.703.2.6 uint32_t unpack_loc_GetServer_Ind_t::serverType

#### 8.703.2.7 uint16_t unpack_loc_GetServer_Ind_t::Tlvresult

## 8.704 unpack_loc_GnssSvInfo_Ind_t Struct Reference

### Data Fields

- uint8_t altitudeAssumed
- loc_satelliteInfo * pSatelliteInfo
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.704.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

**Parameters**

| | |
|---|---|
| *altitudeAssumed* | • Indicates whether altitude is assumed or calculated<br>– 0x00 (FALSE) - Valid altitude is calculated<br>– 0x01 (TRUE) - Valid altitude is assumed; there may not be enough satellites to determine precise altitude<br>• Bit to check in ParamPresenceMask - **1** |
| *pSatelliteInfo* | • See loc_satelliteInfo for more information.<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult* | • unpack result |

### 8.704.2 Field Documentation

#### 8.704.2.1 uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed

#### 8.704.2.2 swi_uint256_t unpack_loc_GnssSvInfo_Ind_t::ParamPresenceMask

#### 8.704.2.3 loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo

#### 8.704.2.4 uint16_t unpack_loc_GnssSvInfo_Ind_t::TlvResult

## 8.705 unpack_loc_InjectPositionCallback_Ind_t Struct Reference

### Data Fields

- uint32_t status
- swi_uint256_t ParamPresenceMask

### 8.705.1 Detailed Description

Contain the parameters passed for LOC Inject Position indication by the device.

**Parameters**

| | |
|---|---|
| *status* | • Position Injection Status<br>• Valid values:<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure.<br>– 2 - Request failed because it is not supported.<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - engine could not allocate sufficient memory<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 -Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |

**Note**

None

### 8.705.2 Field Documentation

#### 8.705.2.1 swi_uint256_t unpack_loc_InjectPositionCallback_Ind_t::ParamPresenceMask

#### 8.705.2.2 uint32_t unpack_loc_InjectPositionCallback_Ind_t::status

## 8.706 unpack_loc_InjectSensorDataCallback_Ind_t Struct Reference

### Data Fields

- uint32_t injectSensorDataStatus
- uint32_t * pOpaqueIdentifier
- uint8_t * pAccelSamplesAccepted
- uint8_t * pGyroSamplesAccepted
- uint8_t * pAccelTempSamplesAccepted
- uint8_t * pGyroTempSamplesAccepted
- swi_uint256_t ParamPresenceMask

## 8.706.1 Detailed Description

This structure contains LOC Inject Sensor Data

**Parameters**

| | |
|---|---|
| *injectSensorDataStatus* | • Values<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure.<br>– 2 - Request failed because it is not supported.<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - engine could not allocate sufficient memory<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 -Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |
| *pOpaqueIdentifier[OPTIONAL]* | • Sent in by the client echoed so the client can relate the indication to the request.<br>• NULL when this TLV not present<br>• Bit to check in ParamPresenceMask - **16** |
| *pAccelSamplesAccepted[OPTIONAL]* | • Lets the client know how many 3-axis accelerometer samples were accepted.<br>• This field is present only if the accelerometer samples were sent in the request.<br>• NULL when this TLV not present<br>• Bit to check in ParamPresenceMask - **17** |
| *pGyroSamplesAccepted[OPTIONAL]* | • Lets the client know how many 3-axis gyroscope samples were accepted.<br>• This field is present only if the gyroscope samples were sent in the request.<br>• NULL when this TLV not present<br>• Bit to check in ParamPresenceMask - **18** |
| *pAccelTempSamplesAccepted[OPTIONAL]* | • Lets the client know how many accelerometer temperature samples were accepted.<br>• This field is present only if the accelerometer temperature samples were sent in the request.<br>• NULL when this TLV not present<br>• Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *pGyroTemp-Samples-Accepted[OPTI-ONAL]* | • Lets the client know how many gyroscope temperature samples were accepted.<br>• This field is present only if the gyroscope temperature samples were sent in the request.<br>• NULL when this TLV not present<br>• Bit to check in ParamPresenceMask - **20** |

### 8.706.2 Field Documentation

#### 8.706.2.1 uint32_t unpack_loc_InjectSensorDataCallback_Ind_t::injectSensorDataStatus

#### 8.706.2.2 uint8_t∗ unpack_loc_InjectSensorDataCallback_Ind_t::pAccelSamplesAccepted

#### 8.706.2.3 uint8_t∗ unpack_loc_InjectSensorDataCallback_Ind_t::pAccelTempSamplesAccepted

#### 8.706.2.4 swi_uint256_t unpack_loc_InjectSensorDataCallback_Ind_t::ParamPresenceMask

#### 8.706.2.5 uint8_t∗ unpack_loc_InjectSensorDataCallback_Ind_t::pGyroSamplesAccepted

#### 8.706.2.6 uint8_t∗ unpack_loc_InjectSensorDataCallback_Ind_t::pGyroTempSamplesAccepted

#### 8.706.2.7 uint32_t∗ unpack_loc_InjectSensorDataCallback_Ind_t::pOpaqueIdentifier

## 8.707 unpack_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference

### Data Fields

- uint32_t injectTimeSyncStatus
- swi_uint256_t ParamPresenceMask

### 8.707.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

**Parameters**

| | |
|---|---|
| *injectTimeSync-Status* | • Values<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure.<br>– 2 - Request failed because it is not supported.<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - engine could not allocate sufficient memory<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 -Location service failed because of an XTRA version-based file format check failure |

- ∗ - Bit to check in ParamPresenceMask - **1**

### 8.707.2 Field Documentation

#### 8.707.2.1 uint32_t unpack_loc_InjectTimeSyncDataCallback_Ind_t::injectTimeSyncStatus

#### 8.707.2.2 swi_uint256_t unpack_loc_InjectTimeSyncDataCallback_Ind_t::ParamPresenceMask

## 8.708 unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference

### Data Fields

- uint32_t status
- swi_uint256_t ParamPresenceMask

### 8.708.1 Detailed Description

Contain the parameters passed for Inject UTC time indication by the device.

**Parameters**

| | |
|---|---|
| *status* | • Status of the UTC Time Injection request<br>• Valid values:<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure.<br>– 2 - Request failed because it is not supported.<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - engine could not allocate sufficient memory<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 -Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |

**Note**

None

### 8.708.2 Field Documentation

#### 8.708.2.1 swi_uint256_t unpack_loc_InjectUTCTimeCallback_Ind_t::ParamPresenceMask

#### 8.708.2.2 uint32_t unpack_loc_InjectUTCTimeCallback_Ind_t::status

## 8.709 unpack_loc_PositionRpt_Ind_t Struct Reference

### Data Fields

- uint32_t sessionStatus
- uint8_t sessionId

- uint64_t * pLatitude
- uint64_t * pLongitude
- uint32_t * pHorUncCircular
- uint32_t * pHorUncEllipseSemiMinor
- uint32_t * pHorUncEllipseSemiMajor
- uint32_t * pHorUncEllipseOrientAzimuth
- uint8_t * pHorConfidence
- uint32_t * pHorReliability
- uint32_t * pSpeedHorizontal
- uint32_t * pSpeedUnc
- uint32_t * pAltitudeWrtEllipsoid
- uint32_t * pAltitudeWrtMeanSeaLevel
- uint32_t * pVertUnc
- uint8_t * pVertConfidence
- uint32_t * pVertReliability
- uint32_t * pSpeedVertical
- uint32_t * pHeading
- uint32_t * pHeadingUnc
- uint32_t * pMagneticDeviation
- uint32_t * pTechnologyMask
- loc_precisionDilution * pPrecisionDilution
- uint64_t * pTimestampUtc
- uint8_t * pLeapSeconds
- loc_gpsTime * pGpsTime
- uint32_t * pTimeUnc
- uint32_t * pTimeSrc
- loc_sensorDataUsage * pSensorDataUsage
- uint32_t * pFixId
- loc_svUsedforFix * pSvUsedforFix
- uint8_t * pAltitudeAssumed
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.709.1 Detailed Description

This structure contains Event Position Report Indication unpack

**Parameters**

| | |
|---|---|
| *sessionStatus* | • Values<br>– 0 - Session was successful<br>– 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.<br>– 2 - Session failed..<br>– 3 - Fix request failed because the session timed out.<br>– 4 - Fix request failed because the session was ended by the user.<br>– 5 - Fix request failed due to bad parameters in the request.<br>– 6 - Fix request failed because the phone is offline.<br>– 7 - Fix request failed because the engine is locked<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *sessionId* | • ID of the session that was specified in the Start request<br>• Range - 0 to 255<br>• Bit to check in ParamPresenceMask - **2** |
| *pLatitude* | • Latitude (specified in WGS84 datum)<br>• Type - Floating point<br>• Units - Degrees<br>• Range - -90.0 to 90.0<br>• Positive values indicate northern latitude<br>• Negative values indicate southern latitude<br>• Bit to check in ParamPresenceMask - **16** |
| *pLongitude* | • Longitude (specified in WGS84 datum)<br>• Type - Floating point<br>• Units - Degrees<br>• Range - -180.0 to 180.0<br>• Positive values indicate eastern latitude<br>• Negative values indicate western latitude<br>• Bit to check in ParamPresenceMask - **17** |
| *pHorUncCircular* | • Horizontal position uncertainty.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **18** |
| *pHorUncEllipse-SemiMinor* | • Semi-minor axis of horizontal elliptical uncertainty.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **19** |
| *pHorUncEllipse-SemiMajor* | • Semi-major axis of horizontal elliptical uncertainty.<br>• Units: Meters<br>• Bit to check in ParamPresenceMask - **20** |
| *pHorUncEllipse-OrientAzimuth* | • Elliptical horizontal uncertainty azimuth of orientation.<br>• Units - Decimal degrees<br>• Range - 0 to 180<br>• Bit to check in ParamPresenceMask - **21** |

| | |
|---|---|
| *pHorConfidence* | • Horizontal uncertainty confidence.<br>• If both elliptical and horizontal uncertainties are specified in this message, the confidence corresponds to the elliptical uncertainty.<br>• Units - Percentage<br>• Range 0-99<br>• Bit to check in ParamPresenceMask - **22** |
| *pHorReliability* | • Values<br>– 0 - Location reliability is not set.<br>– 1 - Location reliability is very low; use it at your own risk<br>– 2 - Location reliability is low; little or no cross-checking is possible.<br>– 3 - Location reliability is medium; limited cross-check passed<br>– 4 - Location reliability is high; strong cross-check passed |

- Bit to check in ParamPresenceMask - **23**

**Parameters**

| | |
|---|---|
| *pSpeed-Horizontal* | • Horizontal speed.<br>• Units - Meters/second<br>• Bit to check in ParamPresenceMask - **24** |
| *pSpeedUnc* | • 3-D Speed uncertainty.<br>• Units - Meters/second.<br>• Bit to check in ParamPresenceMask - **25** |
| *pAltitudeWrt-Ellipsoid* | • Altitude With Respect to WGS84 Ellipsoid.<br>• Units - Meters<br>• Range -500 to 15883<br>• Bit to check in ParamPresenceMask - **26** |
| *pAltitudeWrt-MeanSeaLevel* | • Altitude With Respect to Sea Level.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **27** |
| *pVertUnc* | • Vertical uncertainty.<br>• Units - Meters<br>• Bit to check in ParamPresenceMask - **28** |
| *pVertConfidence* | • Vertical uncertainty confidence.<br>• Units - Percentage<br>• Range 0 to 99<br>• Bit to check in ParamPresenceMask - **29** |

| | |
|---|---|
| *pVertReliability* | • Values<br>– 0 - Location reliability is not set.<br>– 1 - Location reliability is very low; use it at your own risk.<br>– 2 - Location reliability is low; little or no cross-checking is possible<br>– 3 - Location reliability is medium; limited cross-check passed<br>– 4 - Location reliability is high; strong cross-check passed<br>– Bit to check in ParamPresenceMask - **30** |
| *pSpeedVertical* | • Vertical speed.<br>• Units - Meters/second<br>• Bit to check in ParamPresenceMask - **31** |
| *pHeading* | • Heading.<br>• Units - Degree<br>• Range 0 to 359.999<br>• Bit to check in ParamPresenceMask - **32** |
| *pHeadingUnc* | • Heading uncertainty.<br>• Units - Degree<br>• Range 0 to 359.999<br>• Bit to check in ParamPresenceMask - **33** |
| *pMagnetic-Deviation* | • 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.<br>• Bit to check in ParamPresenceMask - **34** |
| *pTechnology-Mask* | • Values<br>– 0x00000001 - Satellites were used to generate the fix<br>– 0x00000002 - Cell towers were used to generate the fix<br>– 0x00000004 - Wi-Fi access points were used to generate the fix<br>– 0x00000008 - Sensors were used to generate the fix<br>– 0x00000010 - Reference Location was used to generate the fix<br>– 0x00000020 - Coarse position injected into the location engine was used to generate the fix<br>– 0x00000040 - AFLT was used to generate the fix<br>– 0x00000080 - GNSS and network-provided measurements were used to generate the fix<br>• Bit to check in ParamPresenceMask - **35** |
| *pPrecision-Dilution* | • See loc_precisionDilution for more information<br>• Bit to check in ParamPresenceMask - **36** |

| | |
|---|---|
| *pTimestampUtc* | • UTC timestamp<br>• Units - Milliseconds since Jan. 1, 1970<br>• Bit to check in ParamPresenceMask - **37** |
| *pLeapSeconds* | • Leap second information. If leapSeconds is not available, timestampUtc is calculated based on a hard-coded value for leap seconds.<br>• Units - Seconds<br>• Bit to check in ParamPresenceMask - **38** |
| *pGpsTime* | • See loc_gpsTime for more information<br>• Bit to check in ParamPresenceMask - **39** |
| *pTimeUnc* | • Time uncertainty.<br>• Units - Milliseconds<br>• Bit to check in ParamPresenceMask - **40** |
| *pTimeSrc* | • Values<br>– 0 - Invalid time.<br>– 1 - Time is set by the 1X system.<br>– 2 - Time is set by WCDMA/GSM time tagging.<br>– 3 - Time is set by an external injection.<br>– 4 - Time is set after decoding over-the-air GPS navigation data from one GPS satellite.<br>– 5 - Time is set after decoding over-the-air GPS navigation data from multiple satellites.<br>– 6 - Both time of the week and the GPS week number are known.<br>– 7 - Time is set by the position engine after the fix is obtained<br>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.<br>– 9 - Time is set after decoding GLO satellites.<br>– 10- Time is set after transforming the GPS to GLO time<br>– 11- Time is set by the sleep time tag provided by the WCDMA network.<br>– 12- Time is set by the sleep time tag provided by the GSM network<br>– 13- Source of the time is unknown<br>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state<br>– 15- Time is set after decoding QZSS satellites.<br>– 16- Time is set after decoding BDS satellites. |

- Bit to check in ParamPresenceMask - **41**

**Parameters**

| | |
|---|---|
| *pSensorData-Usage* | • See loc_sensorDataUsage for more information<br>• Bit to check in ParamPresenceMask - **42** |

| | |
|---|---|
| *pFixId* | • Fix count for the session. Starts with 0 and increments by one for each successive position report for a particular session.<br>• Bit to check in ParamPresenceMask - **43** |
| *pSvUsedforFix* | • See loc_svUsedforFix for more information<br>• Bit to check in ParamPresenceMask - **44** |
| *pAltitude-Assumed* | • Indicates whether altitude is assumed or calculated.<br>• Value<br>– 0x00 - Altitude is calculated<br>– 0x01 - Altitude is assumed<br>• Bit to check in ParamPresenceMask - **45** |

### 8.709.2 Field Documentation

#### 8.709.2.1 uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed

#### 8.709.2.2 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid

#### 8.709.2.3 uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel

#### 8.709.2.4 swi_uint256_t unpack_loc_PositionRpt_Ind_t::ParamPresenceMask

#### 8.709.2.5 uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId

#### 8.709.2.6 loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime

#### 8.709.2.7 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading

#### 8.709.2.8 uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc

#### 8.709.2.9 uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence

#### 8.709.2.10 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability

#### 8.709.2.11 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular

#### 8.709.2.12 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth

#### 8.709.2.13 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor

#### 8.709.2.14 uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor

#### 8.709.2.15 uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude

#### 8.709.2.16 uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds

#### 8.709.2.17 uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude

#### 8.709.2.18 uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation

#### 8.709.2.19 loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution

#### 8.709.2.20 loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage

#### 8.709.2.21 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal

#### 8.709.2.22 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc

#### 8.709.2.23 uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical

#### 8.709.2.24 loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix

#### 8.709.2.25 uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask

#### 8.709.2.26 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc

#### 8.709.2.27 uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc

#### 8.709.2.28 uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeUnc

#### 8.709.2.29 uint8_t* unpack_loc_PositionRpt_Ind_t::pVertConfidence

#### 8.709.2.30 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertReliability

#### 8.709.2.31 uint32_t* unpack_loc_PositionRpt_Ind_t::pVertUnc

#### 8.709.2.32 uint8_t unpack_loc_PositionRpt_Ind_t::sessionId

#### 8.709.2.33 uint32_t unpack_loc_PositionRpt_Ind_t::sessionStatus

#### 8.709.2.34 uint16_t unpack_loc_PositionRpt_Ind_t::Tlvresult

## 8.710 unpack_loc_SensorStreamingCallback_Ind_t Struct Reference

### Data Fields

- loc_accelAcceptReady * pAccelAcceptReady
- loc_gyroAcceptReady * pGyroAcceptReady
- loc_accelTempAcceptReady * pAccelTempAcceptReady
- loc_gyroTempAcceptReady * pGyroTempAcceptReady
- swi_uint256_t ParamPresenceMask

### 8.710.1 Detailed Description

This structure contains LOC Event Sensor Streaming Ready Status

**Parameters**

| | |
|---|---|
| *-pAccelAccept-Ready[OPTION-AL]* | • See loc_accelAcceptReady for more information<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *-pGyroAccept-Ready[OPTION-AL]* | • See loc_gyroAcceptReady for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *-pAccelTemp-AcceptReady[O-PTIONAL]* | • See loc_accelTempAcceptReady for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *-pGyroTemp-AcceptReady[O-PTIONAL]* | • See loc_gyroTempAcceptReady for more information<br>• Bit to check in ParamPresenceMask - **19** |

### 8.710.2 Field Documentation

#### 8.710.2.1 loc_accelAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pAccelAcceptReady

#### 8.710.2.2 loc_accelTempAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pAccelTempAcceptReady

#### 8.710.2.3 swi_uint256_t unpack_loc_SensorStreamingCallback_Ind_t::ParamPresenceMask

#### 8.710.2.4 loc_gyroAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pGyroAcceptReady

#### 8.710.2.5 loc_gyroTempAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pGyroTempAcceptReady

## 8.711 unpack_loc_SetExtPowerConfig_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.711.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

**Parameters**

| | |
|---|---|
| *status* | • Valid values<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure.<br>– 2 - Request failed because it is not supported.<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - engine could not allocate sufficient memory<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 -Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack result |

### 8.711.2 Field Documentation

#### 8.711.2.1 swi_uint256_t unpack_loc_SetExtPowerConfig_Ind_t::ParamPresenceMask

#### 8.711.2.2 uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status

#### 8.711.2.3 uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult

## 8.712 unpack_loc_SetExtPowerState_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.712.1 Detailed Description

This structure contains Set Ext Power State unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.712.2 Field Documentation

#### 8.712.2.1 swi_uint256_t unpack_loc_SetExtPowerState_t::ParamPresenceMask

#### 8.712.2.2 uint16_t unpack_loc_SetExtPowerState_t::Tlvresult

## 8.713 unpack_loc_SetOperationMode_Ind_t Struct Reference

### Data Fields

- uint32_t status
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.713.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

**Parameters**

| | |
|---|---|
| *status* | • Status of the Set Operation Mode request.<br>• Valid values:<br>– 0 - Request was completed successfully<br>– 1 - Request failed because of a general failure<br>– 2 - Request failed because it is not supported<br>– 3 - Request failed because it contained invalid parameters<br>– 4 - Request failed because the engine is busy<br>– 5 - Request failed because the phone is offline<br>– 6 - Request failed because it timed out<br>– 7 - Request failed because an undefined configuration was requested<br>– 8 - Request failed because the engine could not allocate sufficient memory for the request<br>– 9 - Request failed because the maximum number of Geofences are already programmed<br>– 10 - Location service failed because of an XTRA version-based file format check failure<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack result |

### 8.713.2 Field Documentation

#### 8.713.2.1 swi_uint256_t unpack_loc_SetOperationMode_Ind_t::ParamPresenceMask

#### 8.713.2.2 uint32_t unpack_loc_SetOperationMode_Ind_t::status

#### 8.713.2.3 uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult

## 8.714 unpack_loc_SetOperationMode_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.714.1 Detailed Description

This structure contains Set Operation Mode unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.714.2 Field Documentation

#### 8.714.2.1 swi_uint256_t unpack_loc_SetOperationMode_t::ParamPresenceMask

#### 8.714.2.2 uint16_t unpack_loc_SetOperationMode_t::Tlvresult

## 8.715 unpack_loc_SetServer_Ind_t Struct Reference

### Data Fields

- uint32_t serverStatus
- uint16_t Tlvresult

### 8.715.1 Detailed Description

Contain the parameters passed for SetLocSetServerCallback by the device.

**Parameters**

| | |
|---|---|
| *serverStatus* | • Status of the Get Server request.<br>• Valid values<br>– eQMI_LOC_SUCCESS (0) - Request was completed successfully<br>– eQMI_LOC_GENERAL_FAILURE (1) - Request failed because of a general failure<br>– eQMI_LOC_UNSUPPORTED (2) - Request failed because it is not supported<br>– eQMI_LOC_INVALID_PARAMETER (3) - Request failed because it contained invalid parameters<br>– eQMI_LOC_ENGINE_BUSY (4) - Request failed because the engine is busy<br>– eQMI_LOC_PHONE_OFFLINE (5) - Request failed because the phone is offline<br>– eQMI_LOC_TIMEOUT (6) - Request failed because it timed out<br>– eQMI_LOC_CONFIG_NOT_SUPPORTED (7) - Request failed because an undefined configuration was requested<br>– eQMI_LOC_INSUFFICIENT_MEMORY (8) - Request failed because the engine could not allocate sufficient memory for the request<br>– eQMI_LOC_MAX_GEOFENCE_PROGRAMMED (9) - Request failed because the maximum number of Geofences are already programmed<br>– eQMI_LOC_XTRA_VERSION_CHECK_FAILURE (10) - Location service failed because of an XTRA version-based file format check failure<br>– 0xffffffff - Invalid data. |

### 8.715.2 Field Documentation

#### 8.715.2.1 uint32_t unpack_loc_SetServer_Ind_t::serverStatus

#### 8.715.2.2 uint16_t unpack_loc_SetServer_Ind_t::Tlvresult

## 8.716 unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.716.1 Detailed Description

This structure contains Set Operation Mode unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.716.2 Field Documentation

#### 8.716.2.1 swi_uint256_t unpack_loc_SLQSLOCGetBestAvailPos_t::ParamPresenceMask

#### 8.716.2.2 uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult

## 8.717 unpack_loc_Start_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.717.1 Detailed Description

This structure contains Start LOC unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.717.2 Field Documentation

#### 8.717.2.1 swi_uint256_t unpack_loc_Start_t::ParamPresenceMask

#### 8.717.2.2 uint16_t unpack_loc_Start_t::Tlvresult

## 8.718 unpack_loc_Stop_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.718.1 Detailed Description

This structure contains Stop LOC unpack

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack result.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.718.2 Field Documentation

#### 8.718.2.1 swi_uint256_t unpack_loc_Stop_t::ParamPresenceMask

#### 8.718.2.2 uint16_t unpack_loc_Stop_t::Tlvresult

## 8.719 unpack_nas_GetACCOLC_t Struct Reference

### Data Fields

- uint8_t * pAccolc
- swi_uint256_t ParamPresenceMask

### 8.719.1 Detailed Description

Retrieves information about the access overload class unpack structure

**Parameters**

| | |
|---|---|
| *pAccolc* | • An 8-bit integer representation of the ACCOLC.<br>  – Range: 0 to 15 (0x00 to 0x0F).<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.719.2 Field Documentation

#### 8.719.2.1 uint8_t* unpack_nas_GetACCOLC_t::pAccolc

#### 8.719.2.2 swi_uint256_t unpack_nas_GetACCOLC_t::ParamPresenceMask

## 8.720 unpack_nas_GetANAAAAuthenticationStatus_t Struct Reference

### Data Fields

- uint32_t * pAuthStatus
- swi_uint256_t ParamPresenceMask

### 8.720.1 Detailed Description

AN-AAA authentication status of the device

**Parameters**

| | |
|---|---|
| *AN-AAA* | Authentication Status.<br>• Status of the last AN-AAA authentication request, if any, for the current 1xEV-DO session.<br>– 0 - AAA_STATUS_FAILED - Authentication failed<br>– 1 - AAA_STATUS_SUCCESS - Authentication success<br>– 2 - AAA_STATUS_NO_REQUEST - No authentication requested<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.720.2 Field Documentation

#### 8.720.2.1 swi_uint256_t unpack_nas_GetANAAAAuthenticationStatus_t::ParamPresenceMask

#### 8.720.2.2 uint32_t* unpack_nas_GetANAAAAuthenticationStatus_t::pAuthStatus

## 8.721 unpack_nas_GetCDMANetworkParameters_t Struct Reference

### Data Fields

- uint8_t SCI
- uint8_t SCM
- uint8_t RegHomeSID
- uint8_t RegForeignSID
- uint8_t RegForeignNID
- uint8_t ForceRev0
- uint8_t CustomSCP
- uint32_t Protocol
- uint32_t Broadcast
- uint32_t Application
- uint32_t Roaming
- swi_uint256_t ParamPresenceMask

### 8.721.1 Detailed Description

Gets the current CDMA network parameters

**Parameters**

| | |
|---|---|
| *SCI* | • Slot cycle index<br>• Bit to check in ParamPresenceMask - **17** |
| *SCM* | • Station class mark<br>• Bit to check in ParamPresenceMask - **18** |
| *RegHomeSID* | • Register on home SID<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **19** |
| *RegForeignSID* | • Register on foreign SID<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **19** |
| *RegForeignNID* | • Register on foreign NID<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **19** |
| *ForceRev0* | • Force CDMA 1x-EV-DO Rev. 0 mode<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **20** |
| *CustomSCP* | • Use a custom config for CDMA 1x-EV-DO SCP<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **21** |
| *Protocol* | • Protocol mask for custom SCP config<br>– 0x00000001 - Subtype 2 Physical Layer<br>– 0x00000002 - Enhanced CCMAC<br>– 0x00000004 - Enhanced ACMAC<br>– 0x00000008 - Enhanced FTCMAC<br>– 0x00000010 - Subtype 3 RTCMAC<br>– 0x00000020 - Subsystem 1 RTCMAC<br>– 0x00000040 - Enhanced Idle<br>– 0x00000080 - Generic Multimode Capable Disc Port<br>• Bit to check in ParamPresenceMask - **21** |

| | |
|---|---|
| *Broadcast* | • Broadcast mask for custom SCP config<br>– 0x00000001 - Generic broadcast enabled<br>• Bit to check in ParamPresenceMask - **21** |
| *Application* | • Application mask for custom SCP config<br>– 0x00000001 - SN Multiflow Packet Application<br>– 0x00000002 - SN Enhanced Multiflow Packet Application<br>• Bit to check in ParamPresenceMask - **21** |
| *Roaming* | • Roaming preference<br>– 0 - Automatic<br>– 1 - Home Only<br>– 2 - Affiliated Roaming Only<br>– 3 - Home and Affiliated Roaming<br>• Bit to check in ParamPresenceMask - **22** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.721.2 Field Documentation

#### 8.721.2.1 uint32_t unpack_nas_GetCDMANetworkParameters_t::Application

#### 8.721.2.2 uint32_t unpack_nas_GetCDMANetworkParameters_t::Broadcast

#### 8.721.2.3 uint8_t unpack_nas_GetCDMANetworkParameters_t::CustomSCP

#### 8.721.2.4 uint8_t unpack_nas_GetCDMANetworkParameters_t::ForceRev0

#### 8.721.2.5 swi_uint256_t unpack_nas_GetCDMANetworkParameters_t::ParamPresenceMask

#### 8.721.2.6 uint32_t unpack_nas_GetCDMANetworkParameters_t::Protocol

#### 8.721.2.7 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignNID

#### 8.721.2.8 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegForeignSID

#### 8.721.2.9 uint8_t unpack_nas_GetCDMANetworkParameters_t::RegHomeSID

#### 8.721.2.10 uint32_t unpack_nas_GetCDMANetworkParameters_t::Roaming

#### 8.721.2.11 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCI

#### 8.721.2.12 uint8_t unpack_nas_GetCDMANetworkParameters_t::SCM

## 8.722 unpack_nas_GetHomeNetwork3GPP2_t Struct Reference

## Data Fields

- uint16_t * pMCC
- uint16_t * pMNC
- uint8_t nameSize
- uint8_t * pName
- uint16_t * pSID
- uint16_t * pNID
- uint16_t * pNw2MCC
- uint16_t * pNw2MNC
- uint8_t * pNw2DescDisp
- uint8_t * pNw2DescEnc
- uint8_t * pNw2DescLen
- uint8_t * pNw2Name
- swi_uint256_t ParamPresenceMask

### 8.722.1 Detailed Description

Structure for HomeNetwork3GPP2 unpack.

**Parameters**

| | |
|---|---|
| *pMCC* | • Mobile country code (UMTS only).<br>• Bit to check in ParamPresenceMask - **1** |
| *pMNC* | • Mobile network code (UMTS only).<br>• Bit to check in ParamPresenceMask - **1** |
| *nameSize* | • Maximum number of characters (including NULL terminator) that network name array can contain (UMTS only).<br>• Bit to check in ParamPresenceMask - **1** |
| *pName* | • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).<br>• Bit to check in ParamPresenceMask - **1** |
| *pSID* | • Home network system ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000<br>• Bit to check in ParamPresenceMask - **16** |
| *pNID* | • Home network ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pNw2MCC* | • Mobile country code (3GPP2 only).<br>• Range : 0 to 999<br>• Bit to check in ParamPresenceMask - **17** |
| *pNw2MNC* | • Mobile network code (3GPP2 only).<br>• Range : 0 to 999<br>• Bit to check in ParamPresenceMask - **17** |
| *pNw2DescDisp* | • Network Name Display (3GPP2 only). -Valid Value<br>– 0x00 - Do not display<br>– 0x01 - Display<br>– 0xFF - Unknown<br>• Bit to check in ParamPresenceMask - **17** |
| *pNw2DescEnc* | • Encoding of the network description (3GPP2 only).<br>• Valid Value<br>– 0x00 - Octet, unspecified<br>– 0x02 - 7-bit ASCII (liteqmi_helper_decode7bitAsciiEncString)<br>– 0x04 - Unicode<br>– 0x09 - GSM 7-bit default<br>• Bit to check in ParamPresenceMask - **17** |
| *pNw2DescLen* | • Network Description Length (3GPP2 only).<br>• Bit to check in ParamPresenceMask - **17** |
| *pNw2Name* | • Network Name (3GPP2 only).<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.722.2 Field Documentation

#### 8.722.2.1 uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize

#### 8.722.2.2 swi_uint256_t unpack_nas_GetHomeNetwork3GPP2_t::ParamPresenceMask

#### 8.722.2.3 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC

#### 8.722.2.4 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC

#### 8.722.2.5 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName

#### 8.722.2.6 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID

#### 8.722.2.7 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp

#### 8.722.2.8 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc

#### 8.722.2.9 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen

#### 8.722.2.10 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC

#### 8.722.2.11 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC

#### 8.722.2.12 uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name

#### 8.722.2.13 uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID

## 8.723 unpack_nas_GetHomeNetwork_t Struct Reference

### Data Fields

- uint16_t mcc
- uint16_t mnc
- char name [255]
- uint16_t sid
- uint16_t nid
- swi_uint256_t ParamPresenceMask

### 8.723.1 Detailed Description

This structure contains unpack get get home network parameters.

**Parameters**

| | |
|---|---|
| *mcc* | • Mobile country code (UMTS only).<br>• A 16-bit integer representation of MCC. Range: 0 to 999.<br>• Bit to check in ParamPresenceMask - **1** |
| *mnc* | • Mobile network code (UMTS only).<br>• A 16-bit integer representation of MNC. Range: 0 to 999.<br>• Bit to check in ParamPresenceMask - **1** |
| *name* | • Network name or description represented as a NULL terminated string (empty string returned when unknown) (UMTS only).<br>• Bit to check in ParamPresenceMask - **1** |
| *sid* | • Home network system ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *nid* | • Home network ID<br>– 0xFFFF - Unknown.<br>– Only applies to cdma2000<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.723.2 Field Documentation

#### 8.723.2.1 uint16_t unpack_nas_GetHomeNetwork_t::mcc

#### 8.723.2.2 uint16_t unpack_nas_GetHomeNetwork_t::mnc

#### 8.723.2.3 char unpack_nas_GetHomeNetwork_t::name[255]

#### 8.723.2.4 uint16_t unpack_nas_GetHomeNetwork_t::nid

#### 8.723.2.5 swi_uint256_t unpack_nas_GetHomeNetwork_t::ParamPresenceMask

#### 8.723.2.6 uint16_t unpack_nas_GetHomeNetwork_t::sid

## 8.724 unpack_nas_GetNetworkPreference_t Struct Reference

### Data Fields

- uint32_t ActiveTechPref
- uint32_t Duration
- uint32_t PersistentTechPref
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.724.1 Detailed Description

Returns the network registration preference. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use unpack_nas_SLQSGetSys-SelectionPref() for new firmware versions and new modules

**Parameters**

| | |
|---|---|
| *TechnologyPref[-OUT]* | • Bitmask representing the radio technology preference set.<br>• No bits set indicates to the device to automatically determine the technology to use<br>• Values:<br>– Bit 0 - Technology is 3GPP2<br>– Bit 1 - Technology is 3GPP<br>• Any combination of the following may be returned:<br>– Bit 2 - Analog - AMPS if 3GPP2, GSM if 3GPP<br>– Bit 3 - Digital - CDMA if 3GPP2, WCDMA if 3GPP<br>– Bit 4 - HDR<br>– Bit 5 - LTE<br>– Bits 6 to 15 - Reserved<br>• Bit to check in ParamPresenceMask - **1** |
| *Duration[OUT]* | • Duration of active preference<br>– 0 - Permanent<br>– 1 - Power cycle<br>– 2 - Until the end of the next call or a power cycle<br>– 3 - Until the end of the next call, a specified time, or a power cycle<br>– 4 to 6 - Until the end of the next call<br>• Bit to check in ParamPresenceMask - **1** |
| *Persistent-TechnologyPref[-OUT]* | • Bit field representing persistent radio technology preference<br>– Same representation as the pTechnologyPref parameter<br>• Bit to check in ParamPresenceMask - **16** |
| *Tlvresult* | • unpack result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.724.2 Field Documentation

#### 8.724.2.1 uint32_t unpack_nas_GetNetworkPreference_t::ActiveTechPref

#### 8.724.2.2 uint32_t unpack_nas_GetNetworkPreference_t::Duration

#### 8.724.2.3 swi_uint256_t unpack_nas_GetNetworkPreference_t::ParamPresenceMask

#### 8.724.2.4 uint32_t unpack_nas_GetNetworkPreference_t::PersistentTechPref

#### 8.724.2.5 uint16_t unpack_nas_GetNetworkPreference_t::Tlvresult

## 8.725 unpack_nas_GetRFInfo_t Struct Reference

### Data Fields

- uint8_t instancesSize
- RFBandInfoElements RFBandInfoElements [255]
- swi_uint256_t ParamPresenceMask

### 8.725.1 Detailed Description

Structur to strore the RFInfoList

**Parameters**

| | |
|---|---|
| *instanceSize* | • Upon input, maximum number of elements that the RF info instances array can contain.<br>• Upon successful output, actual number of elements in RF info instances array. |
| *RFBandInfo-Elements* | • RF info instances array<br>  – See RFBandInfoElements for more information<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.725.2 Field Documentation

#### 8.725.2.1 uint8_t unpack_nas_GetRFInfo_t::instancesSize

#### 8.725.2.2 swi_uint256_t unpack_nas_GetRFInfo_t::ParamPresenceMask

#### 8.725.2.3 RFBandInfoElements unpack_nas_GetRFInfo_t::RFBandInfoElements[255]

## 8.726 unpack_nas_GetServingNetwork_t Struct Reference

### Data Fields

- uint32_t RegistrationState
- uint32_t CSDomain
- uint32_t PSDomain
- uint32_t RAN
- uint8_t RadioIfacesSize
- uint8_t RadioIfaces [255]
- uint32_t Roaming
- uint16_t MCC
- uint16_t MNC
- uint8_t nameSize
- uint8_t Name [255]
- uint8_t DataCapsLen
- uint8_t DataCaps [255]
- swi_uint256_t ParamPresenceMask

### 8.726.1 Detailed Description

This structure contains unpack get get serving network parameters.

**Parameters**

| | |
|---|---|
| *Registration-State* | • Registration state:<br>– 0 - Not registered<br>– 1 - Registered<br>– 2 - Searching/Not Registered<br>– 3 - Registration Denied<br>– 4 - Unknown<br>• Bit to check in ParamPresenceMask - **1** |
| *CSDomain* | • Circuit switch domain status:<br>– 0 - Unknown/Not Applicable<br>– 1 - Attached<br>– 2 - Detached<br>• Bit to check in ParamPresenceMask - **1** |
| *PSDomain* | • Packet switch domain status<br>– 0 - Unknown/Not Applicable<br>– 1 - Attached<br>– 2 - Detached<br>• Bit to check in ParamPresenceMask - **1** |
| *RAN* | • Type of radio access network on which mobile is registered:<br>– 0 - Unknown<br>– 1 - cdma2000 network<br>– 2 - UMTS network<br>• Bit to check in ParamPresenceMask - **1** |
| *RadioIfacesSize* | • Upon input, maximum number of elements that the radio interface array contain.<br>• Upon successful output, actual number of elements in the radio interface array.<br>• Bit to check in ParamPresenceMask - **1** |
| *RadioIfaces* | • An array of Radio Interface Technology<br>– See qaGobiApiTableRadioInterfaces.h for the Radio Interface Technologies<br>• Bit to check in ParamPresenceMask - **1** |
| *Roaming* | • Roaming indicator<br>• Bit to check in ParamPresenceMask - **16** |
| *MCC* | • Mobile country code<br>• Bit to check in ParamPresenceMask - **18** |

| | |
|---|---|
| *MNC* | • Mobile network code<br>• Bit to check in ParamPresenceMask - **18** |
| *nameSize* | • Maximum number of characters (including NULL terminator) that network name array can contain; applicable only for UMTS networks<br>• Bit to check in ParamPresenceMask - **18** |
| *Name* | • Network name or description represented as a NULL terminated string; empty string is returned when unknown; applicable only for UMTS networks<br>• Bit to check in ParamPresenceMask - **18** |
| *DataCapsLen[IN/OUT]* | • Upon input, the maximum number of elements the data capabilities array can contain.<br>• Upon output, the actual number of elements in the data capabilities array.<br>• Bit to check in ParamPresenceMask - **17** |
| *DataCaps[OUT]* | • Data capabilities array of unsigned long type<br>– 1 - GPRS<br>– 2 - EDGE<br>– 3 - HSDPA<br>– 4 - HSUPA<br>– 5 - WCDMA<br>– 6 - CDMA 1xRTT<br>– 7 - CDMA 1xEV-DO Rev 0<br>– 8 - CDMA 1xEV-DO Rev. A<br>– 9 - GSM<br>– 10 - EVDO Rev. B<br>– 11 - LTE<br>– 12 - HSDPA Plus<br>– 13 - Dual Carrier HSDPA Plus<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.726.2 Field Documentation

#### 8.726.2.1 uint32_t unpack_nas_GetServingNetwork_t::CSDomain

#### 8.726.2.2 uint8_t unpack_nas_GetServingNetwork_t::DataCaps[255]

#### 8.726.2.3 uint8_t unpack_nas_GetServingNetwork_t::DataCapsLen

#### 8.726.2.4 uint16_t unpack_nas_GetServingNetwork_t::MCC

#### 8.726.2.5 uint16_t unpack_nas_GetServingNetwork_t::MNC

#### 8.726.2.6 uint8_t unpack_nas_GetServingNetwork_t::Name[255]

#### 8.726.2.7 uint8_t unpack_nas_GetServingNetwork_t::nameSize

#### 8.726.2.8 swi_uint256_t unpack_nas_GetServingNetwork_t::ParamPresenceMask

#### 8.726.2.9 uint32_t unpack_nas_GetServingNetwork_t::PSDomain

#### 8.726.2.10 uint8_t unpack_nas_GetServingNetwork_t::RadioIfaces[255]

#### 8.726.2.11 uint8_t unpack_nas_GetServingNetwork_t::RadioIfacesSize

#### 8.726.2.12 uint32_t unpack_nas_GetServingNetwork_t::RAN

#### 8.726.2.13 uint32_t unpack_nas_GetServingNetwork_t::RegistrationState

#### 8.726.2.14 uint32_t unpack_nas_GetServingNetwork_t::Roaming

## 8.727 unpack_nas_GetServingNetworkCapabilities_t Struct Reference

### Data Fields

- uint8_t DataCapsLen
- uint8_t DataCaps [255]
- swi_uint256_t ParamPresenceMask

### 8.727.1 Detailed Description

This structure contains unpack get serving network capabilities parameters.

**Parameters**

| | |
|---|---|
| *DataCapsLen* | • Upon input, the maximum number of elements the data capabilities array can contain.<br>• Upon output, the actual number of elements in the data capabilities array.<br>• Bit to check in ParamPresenceMask - **17** |
| *DataCaps[OUT]* | • Data capabilities array of unsigned long type<br>– 1 - GPRS<br>– 2 - EDGE<br>– 3 - HSDPA<br>– 4 - HSUPA<br>– 5 - WCDMA<br>– 6 - CDMA 1xRTT<br>– 7 - CDMA 1xEV-DO Rev 0<br>– 8 - CDMA 1xEV-DO Rev. A<br>– 9 - GSM<br>– 10 - EVDO Rev. B<br>– 11 - LTE<br>– 12 - HSDPA Plus<br>– 13 - Dual Carrier HSDPA Plus<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.727.2 Field Documentation

#### 8.727.2.1 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]

#### 8.727.2.2 uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen

#### 8.727.2.3 swi_uint256_t unpack_nas_GetServingNetworkCapabilities_t::ParamPresenceMask

## 8.728 unpack_nas_GetSignalStrengths_t Struct Reference

### Data Fields

- uint32_t len
- signed char rssi [8]
- uint32_t radio [8]
- swi_uint256_t ParamPresenceMask

### 8.728.1 Detailed Description

This structure contains unpack get signal strengths parameters.

**Parameters**

| | |
|---|---|
| *len* | • number of rssi & radio items following |
| *rssi* | • signal strength array<br>• Bit to check in ParamPresenceMask - **16** |
| *radio* | • radio interface array<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.728.2 Field Documentation

#### 8.728.2.1 uint32_t unpack_nas_GetSignalStrengths_t::len

#### 8.728.2.2 swi_uint256_t unpack_nas_GetSignalStrengths_t::ParamPresenceMask

#### 8.728.2.3 uint32_t unpack_nas_GetSignalStrengths_t::radio[8]

#### 8.728.2.4 signed char unpack_nas_GetSignalStrengths_t::rssi[8]

## 8.729 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
- swi_uint256_t ParamPresenceMask

### 8.729.1 Detailed Description

Contain the network scan information.

**Parameters**

| | |
|---|---|
| *p3GppNetwork-InstanceSize* | • Upon input, maximum number of elements that the network info instance array can contain.<br>• Upon successful output, the actual number of elements in the network info instance array.<br>• Bit to check in ParamPresenceMask - **16** |
| *p3GppNetwork-InfoInstances* | • Network info instance array<br>– See nas_QmiNas3GppNetworkInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pRATInstance-Size* | • Upon input, maximum number of elements that the RAT info instance array can contain.<br>• Upon successful output, the actual number of elements in the RAT info instance array.<br>• Bit to check in ParamPresenceMask - **17** |
| *pRATINstance* | • RAT info instance array<br>– See nas_QmiNas3GppNetworkRAT for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pPCSInstance-Size* | • Upon input, maximum number of elements that the PCS Digit info instance array can contain.<br>• Upon successful output, the actual number of elements in the PCS Digit info instance array.<br>• Bit to check in ParamPresenceMask - **18** |
| *pPCSInstance* | • PCS Digit info instance array<br>– See nas_QmisNasPcsDigit for more information<br>• Bit to check in ParamPresenceMask - **18** |

| | |
|---|---|
| *pScanResult* | • status of network scan<br>• 0x00 - scan successful<br>• 0x01 - scan was aborted<br>• 0x02 - scan did not complete due to a radio link failure recovery in progress<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.729.2 Field Documentation

#### 8.729.2.1 nas_QmiNas3GppNetworkInfo* unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances

#### 8.729.2.2 uint8_t* unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize

#### 8.729.2.3 swi_uint256_t unpack_nas_PerformNetworkScan_t::ParamPresenceMask

#### 8.729.2.4 nas_QmisNasPcsDigit* unpack_nas_PerformNetworkScan_t::pPCSInstance

#### 8.729.2.5 uint8_t* unpack_nas_PerformNetworkScan_t::pPCSInstanceSize

#### 8.729.2.6 nas_QmiNas3GppNetworkRAT* unpack_nas_PerformNetworkScan_t::pRATINstance

#### 8.729.2.7 uint8_t* unpack_nas_PerformNetworkScan_t::pRATInstanceSize

#### 8.729.2.8 uint32_t* unpack_nas_PerformNetworkScan_t::pScanResult

## 8.730 unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference

### Data Fields

- uint8_t dataCapsSize
- uint8_t dataCaps [255]
- swi_uint256_t ParamPresenceMask

### 8.730.1 Detailed Description

This structure contains unpack set data capabilities callback indication parameters.

**Parameters**

| | |
|---|---|
| *dataCapsSize* | • Number of Data Capabilities<br>• Bit to check in ParamPresenceMask - **17** |
| *dataCaps* | • Data Capabilities<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.730.2 Field Documentation

#### 8.730.2.1 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]

#### 8.730.2.2 uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize

#### 8.730.2.3 swi_uint256_t unpack_nas_SetDataCapabilitiesCallback_ind_t::ParamPresenceMask

## 8.731 unpack_nas_SetEventReportInd_t Struct Reference

### Data Fields

- nas_SignalStrengthTlv SSTlv
- nas_RFInfoTlv RFTlv
- nas_RejectReasonTlv RRTlv
- nas_SLQSSignalStrengthsTlv SLQSSSTlv
- swi_uint256_t ParamPresenceMask

### 8.731.1 Detailed Description

This structure contains unpack set event report indication parameters.

**Parameters**

| | |
|---|---|
| *SSTlv* | • signal strength tlv<br>• Bit to check in ParamPresenceMask - **16** |
| *RFTlv* | • RF tlv<br>• Bit to check in ParamPresenceMask - **17** |
| *RRTlv* | • RR tlv<br>• Bit to check in ParamPresenceMask - **18** |
| *SLQSSSTlv* | • signal strength complete info tlv<br>• Bit to check in ParamPresenceMask - **19**<br>• Bit to check in ParamPresenceMask - **20**<br>• Bit to check in ParamPresenceMask - **21**<br>• Bit to check in ParamPresenceMask - **22**<br>• Bit to check in ParamPresenceMask - **23**<br>• Bit to check in ParamPresenceMask - **24**<br>• Bit to check in ParamPresenceMask - **25**<br>• Bit to check in ParamPresenceMask - **26** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.731.2 Field Documentation

#### 8.731.2.1 swi_uint256_t unpack_nas_SetEventReportInd_t::ParamPresenceMask

#### 8.731.2.2 nas_RFInfoTlv unpack_nas_SetEventReportInd_t::RFTlv

#### 8.731.2.3 nas_RejectReasonTlv unpack_nas_SetEventReportInd_t::RRTlv

#### 8.731.2.4 nas_SLQSSignalStrengthsTlv unpack_nas_SetEventReportInd_t::SLQSSSTlv

#### 8.731.2.5 nas_SignalStrengthTlv unpack_nas_SetEventReportInd_t::SSTlv

## 8.732 unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference

### Data Fields

- nas_PhyCaAggScellIndType sPhyCaAggScellIndType
- nas_PhyCaAggScellDlBw sPhyCaAggScellDlBw
- nas_PhyCaAggScellInfo sPhyCaAggScellInfo
- nas_PhyCaAggPcellInfo sPhyCaAggPcellInfo
- nas_PhyCaAggScellIndex sPhyCaAggScellIndex
- swi_uint256_t ParamPresenceMask

### 8.732.1 Detailed Description

Structure for storing the LTE PHY CA indication parameters.

**Parameters**

| | |
|---|---|
| *pPhyCaAgg-ScellIndType* | • See nas_PhyCaAggScellIndType for more information.<br>• Bit to check in ParamPresenceMask - **1** |
| *sPhyCaAgg-ScellDlBw* | • See nas_PhyCaAggScellDlBw for more information.<br>• Bit to check in ParamPresenceMask - **16** |
| *sPhyCaAgg-ScellInfo* | • See nas_PhyCaAggScellInfo for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *sPhyCaAgg-PcellInfo* | • See nas_PhyCaAggPcellInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *sPhyCaAgg-ScellIndex* | • See nas_PhyCaAggScellIndex for more information.<br>• Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.732.2 Field Documentation

#### 8.732.2.1 swi_uint256_t unpack_nas_SetNasLTECphyCaIndCallback_ind_t::ParamPresenceMask

#### 8.732.2.2 nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggPcellInfo

#### 8.732.2.3 nas_PhyCaAggScellDlBw unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellDlBw

#### 8.732.2.4 nas_PhyCaAggScellIndex unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellIndex

#### 8.732.2.5 nas_PhyCaAggScellIndType unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellIndType

#### 8.732.2.6 nas_PhyCaAggScellInfo unpack_nas_SetNasLTECphyCaIndCallback_ind_t::sPhyCaAggScellInfo

## 8.733 unpack_nas_SetNetworkPreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.733.1 Detailed Description

This structure contains unpack set network preference parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack result |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.733.2 Field Documentation

#### 8.733.2.1 swi_uint256_t unpack_nas_SetNetworkPreference_t::ParamPresenceMask

#### 8.733.2.2 uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult

## 8.734 unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference

### Data Fields

- uint8_t roaming
- swi_uint256_t ParamPresenceMask

### 8.734.1 Detailed Description

This structure contains unpack set roaming indicator callback indication parameters.

**Parameters**

| | |
|---|---|
| *roaming* | • Roaming Indication<br>– 0 - Roaming<br>– 1 - Home<br>– 2 - Roaming partner<br>– >2 - Operator defined values<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.734.2 Field Documentation

#### 8.734.2.1 swi_uint256_t unpack_nas_SetRoamingIndicatorCallback_ind_t::ParamPresenceMask

#### 8.734.2.2 uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming

## 8.735 unpack_nas_SetServingSystemCallback_ind_t Struct Reference

### Data Fields

- NASServingSystemInfo SSInfo
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.735.1 Detailed Description

This structure contains unpack set serving system callback indication parameters.

**Parameters**

| | |
|---|---|
| *SSInfo* | • Serving system parameters information<br>– See NASServingSystemInfo for more details<br>• Bit to check in ParamPresenceMask - **1** |
| *Tlvresult* | • unpack result<br>• Bit to check in ParamPresenceMask - **2** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.735.2 Field Documentation

#### 8.735.2.1 swi_uint256_t unpack_nas_SetServingSystemCallback_ind_t::ParamPresenceMask

#### 8.735.2.2 NASServingSystemInfo unpack_nas_SetServingSystemCallback_ind_t::SSInfo

#### 8.735.2.3 uint16_t unpack_nas_SetServingSystemCallback_ind_t::Tlvresult

## 8.736 unpack_nas_SLQSGetErrorRate_t Struct Reference

### Data Fields

- uint16_t * pCDMAFrameErrRate
- uint16_t * pHDRPackErrRate
- uint8_t * pGSMBER
- uint8_t * pWCDMABER
- swi_uint256_t ParamPresenceMask

### 8.736.1 Detailed Description

Structure for Error Rate unpack parameters.

**Parameters**

| | |
|---|---|
| *pCDMAFrame-ErrRate* | • CDMA Frame Error Rate<br>• Valid error rate values between 1 and 10000 are returned to indicate the percentage, e.g., a value of 300 means the error rate is 3%.<br>• A value of 0xFFFF indicates that the error rate is unknown/unavailable.<br>• Bit to check in ParamPresenceMask - **16** |
| *pHDRPackErr-Rate* | • HDR Packet Error Rate<br>• Valid error rate values between 1 and 10000 are returned to indicate the percentage, e.g., a value of 300 means the error rate is 3%.<br>• A value of 0xFFFF indicates that the error rate is unknown/unavailable.<br>• Bit to check in ParamPresenceMask - **17** |
| *pGSMBER* | • GSM Bit Error Rate<br>• Valid error rate values between 1 and 100 are returned to indicate the percentage value.<br>• A 0% block error rate (BLER) indicates No Data.<br>• Bit to check in ParamPresenceMask - **18** |
| *pWCDMABER* | • WCDMA Block Error Rate<br>• Valid error rate values between 1 and 100 are returned to indicate the percentage value.<br>• A value of 0xFF indicates that the error rate is unknown/unavailable.<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.736.2 Field Documentation

#### 8.736.2.1 swi_uint256_t unpack_nas_SLQSGetErrorRate_t::ParamPresenceMask

#### 8.736.2.2 uint16_t* unpack_nas_SLQSGetErrorRate_t::pCDMAFrameErrRate

#### 8.736.2.3 uint8_t* unpack_nas_SLQSGetErrorRate_t::pGSMBER

#### 8.736.2.4 uint16_t* unpack_nas_SLQSGetErrorRate_t::pHDRPackErrRate

#### 8.736.2.5 uint8_t* unpack_nas_SLQSGetErrorRate_t::pWCDMABER

## 8.737 unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference

### Data Fields

- NasGetLTECphyCaInfo LTECphyCaInfo
- uint16_t TlvResult
- swi_uint256_t ParamPresenceMask

### 8.737.1 Detailed Description

This structure contains unpack get carrier aggregation event information parameters.

**Parameters**

| | |
|---|---|
| *LTECphyCa* | • Carrier aggregation event information<br>  – See NasGetLTECphyCaInfo for more details |
| *TlvResult* | • unpack result<br>• Bit to check in ParamPresenceMask - **2** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.737.2 Field Documentation

#### 8.737.2.1 NasGetLTECphyCaInfo unpack_nas_SlqsGetLTECphyCAInfo_t::LTECphyCaInfo

#### 8.737.2.2 swi_uint256_t unpack_nas_SlqsGetLTECphyCAInfo_t::ParamPresenceMask

#### 8.737.2.3 uint16_t unpack_nas_SlqsGetLTECphyCAInfo_t::TlvResult

## 8.738 unpack_nas_SLQSGetNetworkTime_t Struct Reference

### Data Fields

- nas_timeInfo * p3GPP2TimeInfo
- nas_timeInfo * p3GPPTimeInfo
- swi_uint256_t ParamPresenceMask

### 8.738.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

**Parameters**

| | |
|---|---|
| *p3GPP2TimeInfo* | [Optional]<br>• See nas_timeInfo for more information |

- ∗ - Bit to check in ParamPresenceMask - **16**

**Parameters**

| | |
|---|---|
| *p3GPPTimeInfo* | [Optional]<br>– See nas_timeInfo for more information |

- ∗ - Bit to check in ParamPresenceMask - **17**

**Parameters**

| | |
|---|---|
| *ParamPresenceMask* | – bitmask representation to indicate valid parameters. |

### 8.738.2 Field Documentation

#### 8.738.2.1 nas_timeInfo∗ unpack_nas_SLQSGetNetworkTime_t::p3GPP2TimeInfo

#### 8.738.2.2 nas_timeInfo∗ unpack_nas_SLQSGetNetworkTime_t::p3GPPTimeInfo

#### 8.738.2.3 swi_uint256_t unpack_nas_SLQSGetNetworkTime_t::ParamPresenceMask

## 8.739 unpack_nas_SLQSGetOperatorNameData_t Struct Reference

### Data Fields

- nas_serviceProviderName ∗ pSrvcProviderName
- nas_operatorPLMNList ∗ pOperatorPLMNList
- nas_PLMNNetworkName ∗ pPLMNNetworkName
- nas_operatorNameString ∗ pOperatorNameString
- nas_PLMNNetworkNameData ∗ pNITZInformation
- swi_uint256_t ParamPresenceMask

### 8.739.1 Detailed Description

Structure for Operator Name unpack.

**Parameters**

| | |
|---|---|
| *pSrvcProviderName* | • Refer nas_serviceProviderName for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pOperatorPLMNList* | • Refer nas_operatorPLMNList for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• Bit to check in ParamPresenceMask - **17** |
| *pPLMNNetworkName* | • Refer nas_PLMNNetworkName for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• Bit to check in ParamPresenceMask - **18** |
| *pOperatorNameString* | • Refer nas_operatorNameString for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• Bit to check in ParamPresenceMask - **19** |
| *pNITZInformation* | • Refer nas_PLMNNetworkNameData for details (Optional).<br>• Can provide NULL if this parameter is not required.<br>• Bit to check in ParamPresenceMask - **20** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.739.2 Field Documentation

#### 8.739.2.1 swi_uint256_t unpack_nas_SLQSGetOperatorNameData_t::ParamPresenceMask

#### 8.739.2.2 nas_PLMNNetworkNameData* unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation

#### 8.739.2.3 nas_operatorNameString* unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString

#### 8.739.2.4 nas_operatorPLMNList* unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList

#### 8.739.2.5 nas_PLMNNetworkName* unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName

#### 8.739.2.6 nas_serviceProviderName* unpack_nas_SLQSGetOperatorNameData_t::pSrvcProviderName

## 8.740 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]
- swi_uint256_t ParamPresenceMask

### 8.740.1 Detailed Description

This structure contains unpack get operator name for specificed network parameters.

**Parameters**

| | |
|---|---|
| *spnEncoding* | • Coding scheme used for service provider name. This value is ignored if spn_len is zero Values:<br>– 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0<br>– 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038<br>– Note: This value is ignored if spnLength is zero.<br>• Bit to check in ParamPresenceMask - **16** |
| *spnLength* | • Length of SPN which follows<br>• Note: This SPN value is ignored if spnLength is zero and spnEncoding is 0xff.<br>• Bit to check in ParamPresenceMask - **16** |
| *spn* | • Service Provider name string<br>• Bit to check in ParamPresenceMask - **16** |
| *shortNameEn* | • Coding scheme used for PLMN short name. This value is ignored if PLMN short name length is zero Values:<br>– 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0<br>– 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038<br>– Note: This value is ignored if shortNameLen is zero.<br>• Bit to check in ParamPresenceMask - **16** |
| *shortNameCI* | • Indicates whether the country initials are to be added to the shortName. This value is ignored if shortNameLen is zero. Values:<br>– 0x00 - Do not add the letters for the countrys initials to the name<br>– 0x01 - Add the countrys initials and a text string to the name<br>– 0xFF - Not specified<br>– Note: This value is ignored if shortNameLen is zero.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *shortNameSB* | • PLMN short name spare bits. This value is ignored if shortNameLen is zero. Values:<br>– 0x01 - Bit 8 is spare and set to 0 in octet<br>– 0x02 - Bits 7 and 8 are spare and set to 0 in octet n<br>– 0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x00 - Carries no information about the number of spare bits in octet n<br>– Note: This value is ignored if shortNameLen is zero.<br>• Bit to check in ParamPresenceMask - **16** |
| *shortNameLen* | • Length of shortName which follows<br>• Note: This shortName value is ignored if shortNameLen is zero and shortNameEn is 0xff.<br>• Bit to check in ParamPresenceMask - **16** |
| *shortName* | • PLMN short name<br>• Bit to check in ParamPresenceMask - **16** |
| *longNameEn* | • Coding scheme used for PLMN long name. This value is ignored if PLMN long name length is zero Values:<br>– 0x00 - SMS default 7-bit coded alphabet as defined in 3GPP TS 23.038 with bit 8 set to 0<br>– 0x01 - UCS2 (16 bit, little-endian) 3GPP TS 23.038<br>– Note: This value is ignored if shortNameLen is zero.<br>• Bit to check in ParamPresenceMask - **16** |
| *longNameCI* | • Indicates whether the country initials are to be added to the longName. This value is ignored if longNameLen is zero. Values:<br>– 0x00 - Do not add the letters for the countrys initials to the name<br>– 0x01 - Add the countrys initials and a text string to the name<br>– 0xFF - Not specified<br>– Note: This value is ignored if shortNameLen is zero.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *longNameSB* | • PLMN long name spare bits. This value is ignored if longNameLen is zero. Values:<br>– 0x01 - Bit 8 is spare and set to 0 in octet<br>– 0x02 - Bits 7 and 8 are spare and set to 0 in octet n<br>– 0x03 - Bits 6 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x04 - Bits 5 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x05 - Bits 4 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x06 - Bits 3 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x07 - Bits 2 to 8 (inclusive) are spare and set to 0 in octet n<br>– 0x00 - Carries no information about the number of spare bits in octet n<br>– Note: This value is ignored if shortNameLen is zero.<br>• Bit to check in ParamPresenceMask - **16** |
| *longNameLen* | • Length of longName which follows<br>• Note: This longName value is ignored if longNameLen is zero and longNameEn is 0xff.<br>• Bit to check in ParamPresenceMask - **16** |
| *longName* | • PLMN long name<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

## 8.740.2 Field Documentation

### 8.740.2.1 char unpack_nas_SLQSGetPLMNName_t::longName[255]

### 8.740.2.2 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameCI

### 8.740.2.3 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameEn

### 8.740.2.4 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameLen

### 8.740.2.5 uint8_t unpack_nas_SLQSGetPLMNName_t::longNameSB

### 8.740.2.6 swi_uint256_t unpack_nas_SLQSGetPLMNName_t::ParamPresenceMask

### 8.740.2.7 uint8_t unpack_nas_SLQSGetPLMNName_t::shortName[255]

### 8.740.2.8 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameCI

### 8.740.2.9 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameEn

### 8.740.2.10 char unpack_nas_SLQSGetPLMNName_t::shortNameLen

### 8.740.2.11 uint8_t unpack_nas_SLQSGetPLMNName_t::shortNameSB

### 8.740.2.12 char unpack_nas_SLQSGetPLMNName_t::spn[255]

#### 8.740.2.13 uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding

#### 8.740.2.14 uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength

## 8.741 unpack_nas_SLQSGetServingSystem_t Struct Reference

### Data Fields

- nas_servSystem ServingSystem
- uint8_t RoamIndicatorVal
- nas_dataSrvCapabilities DataSrvCapabilities
- nas_currentPLMN CurrentPLMN
- uint16_t SystemID
- uint16_t NetworkID
- uint16_t BasestationID
- uint32_t BasestationLatitude
- uint32_t BasestationLongitude
- nas_roamIndList RoamingIndicatorList
- uint8_t DefaultRoamInd
- nas_qaQmi3Gpp2TimeZone Gpp2TimeZone
- uint8_t CDMA_P_Rev
- uint8_t GppTimeZone
- uint8_t GppNetworkDSTAdjustment
- uint16_t Lac
- uint32_t CellID
- uint8_t ConcSvcInfo
- uint8_t PRLInd
- uint8_t DTMInd
- nas_detailSvcInfo DetailedSvcInfo
- nas_CDMASysInfoExt CDMASystemInfoExt
- uint8_t HdrPersonality
- uint16_t TrackAreaCode
- nas_callBarStatus CallBarStatus
- swi_uint256_t ParamPresenceMask

### 8.741.1 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**

| | |
|---|---|
| *ServingSystem* | • Serving System<br>• See nas_servSystem for more information<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *RoamIndicator-Val* | • Optional parameter indicating Roaming Indicator value<br>• Values:<br>– 0x00 - Roaming<br>– 0x01 - Home<br>– 0x02 - Flashing<br>– 0x03 and above - Operator defined values<br>• Bit to check in ParamPresenceMask - **16** |
| *DataSrv-Capabilities* | • Optional parameter indicating Data services capability<br>• See nas_dataSrvCapabilities for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *CurrentPLMN* | • Optional parameter indicating Current PLMN<br>• See nas_currentPLMN for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *SystemID* | • Optional parameter indicating System ID<br>• Bit to check in ParamPresenceMask - **19** |
| *NetworkID* | • Optional parameter indicating Network ID<br>• Bit to check in ParamPresenceMask - **19** |
| *BaseStationID* | • Optional parameter indicating Base Station Identification Number<br>• Bit to check in ParamPresenceMask - **20** |
| *BaseStation-Latitude* | • Optional parameter indicating Base station latitude in units of 0.25 sec,expressed as a two's complement signed number with positive numbers signifying North latitude<br>• Bit to check in ParamPresenceMask - **20** |
| *Basestation-Longitude* | • Optional parameter indicating Base station longitude in units of 0.25 sec, expressed as a Two's complement signed number with positive numbers signifying East longitude<br>• Bit to check in ParamPresenceMask - **20** |
| *Roaming-IndicatorList* | • Optional parameter indicating Roaming Indicator List<br>• See nas_roamIndList for more information<br>• Bit to check in ParamPresenceMask - **21** |

| | |
|---|---|
| *DefaultRoamInd* | • Optional parameter indicating Default Roaming Indicator<br>• Values:<br>– 0x00 - Roaming<br>– 0x01 - Home<br>• Bit to check in ParamPresenceMask - **22** |
| *Gpp2TimeZone* | • Optional parameter indicating 3GPP2 Time Zone<br>• See nas_qaQmi3Gpp2TimeZone for more information<br>• Bit to check in ParamPresenceMask - **23** |
| *CDMA_P_Rev* | • Optional parameter indicating CDMA P_Rev in use<br>• Bit to check in ParamPresenceMask - **24** |
| *GppTimeZone* | • Optional parameter indicating Offset from Universal time, i.e., difference between local time and Universal time, in increments of 15 min. (signed value).<br>• Bit to check in ParamPresenceMask - **26** |
| *GppNetworkDS-TAdjustment* | • Optional parameter indicating 3GPP network daylight saving adjustment<br>• Values:<br>– 0x00 - No adjustment for Daylight Saving Time<br>– 0x01 - 1 hr adjustment for Daylight Saving Time<br>– 0x02 - 2 hr adjustment for Daylight Saving Time<br>• Bit to check in ParamPresenceMask - **27** |
| *Lac* | • Optional parameter indicating 3GPP Location Area Code<br>• Bit to check in ParamPresenceMask - **28** |
| *CellID* | • Optional parameter indicating 3GPP Cell ID<br>• Bit to check in ParamPresenceMask - **29** |
| *ConcSvcInfo* | • Optional parameter indicating 3GPP2 concurrent service Info<br>• Values:<br>– 0x00 - Concurrent service not available<br>– 0x01 - Concurrent service available<br>• Bit to check in ParamPresenceMask - **30** |
| *PRLInd* | • Optional parameter indicating 3GPP2 PRL Indicator<br>• Values:<br>– 0x00 - System not in PRL<br>– 0x01 - System is in PRL<br>• Bit to check in ParamPresenceMask - **31** |

| | |
|---|---|
| *DTMInd* | • Optional parameter indicating Dual Transfer Mode Indication(GSM Only)<br>• Values:<br>– 0x00 - DTM not supported<br>– 0x01 - DTM supported<br>• Bit to check in ParamPresenceMask - **32** |
| *DetailedSvcInfo* | • Optional parameter indicating Detailed service information<br>• See nas_detailSvcInfo for more information<br>• Bit to check in ParamPresenceMask - **33** |
| *CDMASystem-InfoExt* | • Optional parameter indicating CDMA System Info Ext<br>• See nas_CDMASysInfoExt for more information<br>• Bit to check in ParamPresenceMask - **34** |
| *HdrPersonality* | • Optional parameter indicating HDR Personality Information<br>• Values:<br>– 0x00 - Unknown<br>– 0x01 - HRPD<br>– 0x02 - eHRPD<br>• Bit to check in ParamPresenceMask - **35** |
| *TrackAreaCode* | • Optional parameter indicating Tracking area code information for LTE<br>• Bit to check in ParamPresenceMask - **36** |
| *CallBarStatus* | • Optional parameter indicating Call Barring Status<br>• See nas_callBarStatus for more information<br>• Bit to check in ParamPresenceMask - **37** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

## 8.741.2 Field Documentation

### 8.741.2.1 uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID

### 8.741.2.2 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude

### 8.741.2.3 uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude

### 8.741.2.4 nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus

### 8.741.2.5 uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev

### 8.741.2.6 nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt

#### 8.741.2.7 uint32_t unpack_nas_SLQSGetServingSystem_t::CellID

#### 8.741.2.8 uint8_t unpack_nas_SLQSGetServingSystem_t::ConcSvcInfo

#### 8.741.2.9 nas_currentPLMN unpack_nas_SLQSGetServingSystem_t::CurrentPLMN

#### 8.741.2.10 nas_dataSrvCapabilities unpack_nas_SLQSGetServingSystem_t::DataSrvCapabilities

#### 8.741.2.11 uint8_t unpack_nas_SLQSGetServingSystem_t::DefaultRoamInd

#### 8.741.2.12 nas_detailSvcInfo unpack_nas_SLQSGetServingSystem_t::DetailedSvcInfo

#### 8.741.2.13 uint8_t unpack_nas_SLQSGetServingSystem_t::DTMInd

#### 8.741.2.14 nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetServingSystem_t::Gpp2TimeZone

#### 8.741.2.15 uint8_t unpack_nas_SLQSGetServingSystem_t::GppNetworkDSTAdjustment

#### 8.741.2.16 uint8_t unpack_nas_SLQSGetServingSystem_t::GppTimeZone

#### 8.741.2.17 uint8_t unpack_nas_SLQSGetServingSystem_t::HdrPersonality

#### 8.741.2.18 uint16_t unpack_nas_SLQSGetServingSystem_t::Lac

#### 8.741.2.19 uint16_t unpack_nas_SLQSGetServingSystem_t::NetworkID

#### 8.741.2.20 swi_uint256_t unpack_nas_SLQSGetServingSystem_t::ParamPresenceMask

#### 8.741.2.21 uint8_t unpack_nas_SLQSGetServingSystem_t::PRLInd

#### 8.741.2.22 uint8_t unpack_nas_SLQSGetServingSystem_t::RoamIndicatorVal

#### 8.741.2.23 nas_roamIndList unpack_nas_SLQSGetServingSystem_t::RoamingIndicatorList

#### 8.741.2.24 nas_servSystem unpack_nas_SLQSGetServingSystem_t::ServingSystem

#### 8.741.2.25 uint16_t unpack_nas_SLQSGetServingSystem_t::SystemID

#### 8.741.2.26 uint16_t unpack_nas_SLQSGetServingSystem_t::TrackAreaCode

## 8.742 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 Io
- uint8_t sinr
- uint16_t errorRateListLen
- nas_errorRateListElement errorRateList [18]
- nas_rsrqInformation rsrqInfo
- int16_t ltesnr

- int16_t ltersrp
- swi_uint256_t ParamPresenceMask

### 8.742.1 Detailed Description

This structure contains the Signal Strength Information

**Parameters**

| | |
|---|---|
| *signalStrength-ReqMask* | • Request Mask<br>– Request additional signal information for: Bit 0 - RSSI Information bit<br>Valid values are:<br>0 - Do Not Request Additional Info for RSSI<br>1 - Request Additional Info for RSSI<br>Bit 1 - ECIO Information bit<br>Valid values are:<br>0 - Do Not Request Additional Info for ECIO<br>1 - Request Additional Info for ECIO<br>Bit 2 - IO Information bit<br>Valid values are:<br>0 - Do Not Request Additional Info for IO<br>1 - Request Additional Info for IO<br>Bit 3 - SINR Information bit<br>Valid values are:<br>0 - Do Not Request Additional Info for SINR<br>1 - Request Additional Info for SINR<br>Bit 4 - ERROR RATE Information bit<br>Valid values are:<br>0 - Do Not Request Additional Info for Error Rate<br>1 - Request Additional Info for Error Rate<br>Bit 5 - RSRQ Information bit<br>Valid values are:<br>0 - Do Not Request Additional Info for RSRQ<br>1 - Request Additional Info for RSRQ<br>Bit 6 - LTE SNR information bit<br>Valid values are:<br>0 - Do not request additional information for LTE SNR<br>1 - Request additional information for LTE SNR Bit 7 - LTE RSRP Information bit<br>Valid values are:<br>0 - Do not request additional information for LTE RSRP<br>1 - Request additional information for LTE RSRP<br>• Bit to check in ParamPresenceMask - **1** |
| *rxSignal-StrengthListLen* | • Number of elements in Receive Signal Strength List<br>• Bit to check in ParamPresenceMask - **17** |
| *rxSignal-StrengthList* | • See nas_rxSignalStrengthListElement for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *ecioListLen* | • Number of elements in ECIO List<br>• Bit to check in ParamPresenceMask - **18** |

| | |
|---|---|
| *ecioList* | • See nas_ecioListElement for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *Io* | • Received Io in dBm; IO is only applicable for 1xEV-DO<br>• Bit to check in ParamPresenceMask - **19** |
| *sinr* | • SINR level<br>– SINR is only applicable for 1xEV-DO; valid levels are 0 to 8 where maximum value for 0 - SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3 - SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 - SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB<br>• Bit to check in ParamPresenceMask - **20** |
| *errorRateListLen* | • Number of elements in Error Rate List<br>• Bit to check in ParamPresenceMask - **21** |
| *errorRateList* | • See nas_errorRateListElement for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *rsrqInfo* | • See nas_rsrqInformation for more information<br>• Bit to check in ParamPresenceMask - **22** |
| *ltesnr* | • LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE<br>• Bit to check in ParamPresenceMask - **23** |
| *ltersrp* | • LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE<br>• Bit to check in ParamPresenceMask - **24** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.742.2 Field Documentation

#### 8.742.2.1 **nas_ecioListElement** unpack_nas_SLQSGetSignalStrength_t::ecioList[18]

#### 8.742.2.2 uint16_t unpack_nas_SLQSGetSignalStrength_t::ecioListLen

#### 8.742.2.3 **nas_errorRateListElement** unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]

#### 8.742.2.4 uint16_t unpack_nas_SLQSGetSignalStrength_t::errorRateListLen

#### 8.742.2.5 int32_t unpack_nas_SLQSGetSignalStrength_t::lo

#### 8.742.2.6 int16_t unpack_nas_SLQSGetSignalStrength_t::ltersrp

#### 8.742.2.7 int16_t unpack_nas_SLQSGetSignalStrength_t::ltesnr

#### 8.742.2.8 swi_uint256_t unpack_nas_SLQSGetSignalStrength_t::ParamPresenceMask

#### 8.742.2.9 nas_rsrqInformation unpack_nas_SLQSGetSignalStrength_t::rsrqInfo

#### 8.742.2.10 nas_rxSignalStrengthListElement unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]

#### 8.742.2.11 uint16_t unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen

#### 8.742.2.12 uint16_t unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask

#### 8.742.2.13 uint8_t unpack_nas_SLQSGetSignalStrength_t::sinr

## 8.743 unpack_nas_SLQSGetSysInfo_t Struct Reference

### Data Fields

- nas_SrvStatusInfo * pCDMASrvStatusInfo
- nas_SrvStatusInfo * pHDRSrvStatusInfo
- nas_GSMSrvStatusInfo * pGSMSrvStatusInfo
- nas_GSMSrvStatusInfo * pWCDMASrvStatusInfo
- nas_GSMSrvStatusInfo * pLTESrvStatusInfo
- nas_CDMASysInfo * pCDMASysInfo
- nas_HDRSysInfo * pHDRSysInfo
- nas_GSMSysInfo * pGSMSysInfo
- nas_WCDMASysInfo * pWCDMASysInfo
- nas_LTESysInfo * pLTESysInfo
- nas_AddCDMASysInfo * pAddCDMASysInfo
- uint16_t * pAddHDRSysInfo
- nas_AddSysInfo * pAddGSMSysInfo
- nas_AddSysInfo * pAddWCDMASysInfo
- uint16_t * pAddLTESysInfo
- nas_CallBarringSysInfo * pGSMCallBarringSysInfo
- nas_CallBarringSysInfo * pWCDMACallBarringSysInfo
- uint8_t * pLTEVoiceSupportSysInfo
- uint8_t * pGSMCipherDomainSysInfo
- uint8_t * pWCDMACipherDomainSysInfo
- swi_uint256_t ParamPresenceMask

### 8.743.1 Detailed Description

This structure contains unpack get system information parameters.

**Parameters**

| | |
|---|---|
| *pCDMASrv-StatusInfo* | • See nas_SrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pHDRSrvStatus-Info* | • See nas_SrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pGSMSrvStatus-Info* | • See nas_GSMSrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *pWCDMASrv-StatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **19** |
| *pLTESrvStatus-Info* | • See nas_GSMSrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *pCDMASysInfo* | • See nas_CDMASysInfo for more information.<br>• Bit to check in ParamPresenceMask - **21** |
| *pHDRSysInfo* | • See nas_HDRSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **22** |
| *pGSMSysInfo* | • See nas_GSMSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **23** |
| *pWCDMASys-Info* | • See nas_WCDMASysInfo for more information.<br>• Bit to check in ParamPresenceMask - **24** |
| *pLTESysInfo* | • See nas_LTESysInfo for more information.<br>• Bit to check in ParamPresenceMask - **25** |
| *pAddCDMASys-Info* | • See nas_AddCDMASysInfo for more information.<br>• Bit to check in ParamPresenceMask - **26** |
| *pAddHDRSys-Info* | • System table index referencing the beginning of the geo in which the current serving system is present.<br>• When the system index is not known, 0xFFFF is used.<br>• Bit to check in ParamPresenceMask - **27** |
| *pAddGSMSys-Info* | • See nas_AddSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **28** |

| | |
|---|---|
| *pAddWCDMA-SysInfo* | • See nas_AddSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **29** |
| *pAddLTESysInfo* | • System table index referencing the beginning of the geo in which the current serving system is present.<br>• When the system index is not known, 0xFFFF is used.<br>• Bit to check in ParamPresenceMask - **30** |
| *pGSMCall-BarringSysInfo* | • See nas_CallBarringSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **31** |
| *pWCDMACall-BarringSysInfo* | • See nas_CallBarringSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **32** |
| *pLTEVoice-SupportSysInfo* | • Indicates voice support status on LTE.<br>– 0x00 - Voice is not supported<br>– 0x01 - Voice is supported<br>• Bit to check in ParamPresenceMask - **33** |
| *pGSMCipher-DomainSysInfo* | • Ciphering on the service domain.<br>– 0x00 - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>• Bit to check in ParamPresenceMask - **34** |
| *pWCDMA-CipherDomain-SysInfo* | • Ciphering on the service domain.<br>– 0x00 - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>• Bit to check in ParamPresenceMask - **35** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

## 8.743.2 Field Documentation

### 8.743.2.1 nas_AddCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo

### 8.743.2.2 nas_AddSysInfo* unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo

### 8.743.2.3 uint16_t* unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo

#### 8.743.2.4 uint16_t* unpack_nas_SLQSGetSysInfo_t::pAddLTESysInfo

#### 8.743.2.5 nas_AddSysInfo* unpack_nas_SLQSGetSysInfo_t::pAddWCDMASysInfo

#### 8.743.2.6 swi_uint256_t unpack_nas_SLQSGetSysInfo_t::ParamPresenceMask

#### 8.743.2.7 nas_SrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pCDMASrvStatusInfo

#### 8.743.2.8 nas_CDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pCDMASysInfo

#### 8.743.2.9 nas_CallBarringSysInfo* unpack_nas_SLQSGetSysInfo_t::pGSMCallBarringSysInfo

#### 8.743.2.10 uint8_t* unpack_nas_SLQSGetSysInfo_t::pGSMCipherDomainSysInfo

#### 8.743.2.11 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pGSMSrvStatusInfo

#### 8.743.2.12 nas_GSMSysInfo* unpack_nas_SLQSGetSysInfo_t::pGSMSysInfo

#### 8.743.2.13 nas_SrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pHDRSrvStatusInfo

#### 8.743.2.14 nas_HDRSysInfo* unpack_nas_SLQSGetSysInfo_t::pHDRSysInfo

#### 8.743.2.15 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pLTESrvStatusInfo

#### 8.743.2.16 nas_LTESysInfo* unpack_nas_SLQSGetSysInfo_t::pLTESysInfo

#### 8.743.2.17 uint8_t* unpack_nas_SLQSGetSysInfo_t::pLTEVoiceSupportSysInfo

#### 8.743.2.18 nas_CallBarringSysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMACallBarringSysInfo

#### 8.743.2.19 uint8_t* unpack_nas_SLQSGetSysInfo_t::pWCDMACipherDomainSysInfo

#### 8.743.2.20 nas_GSMSrvStatusInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASrvStatusInfo

#### 8.743.2.21 nas_WCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pWCDMASysInfo

## 8.744 unpack_nas_SLQSGetSysSelectionPref_t Struct Reference

### Data Fields

- uint8_t * pEmerMode
- uint16_t * pModePref
- uint64_t * pBandPref
- uint16_t * pPRLPref
- uint16_t * pRoamPref
- uint64_t * pLTEBandPref
- uint8_t * pNetSelPref
- uint32_t * pSrvDomainPref
- uint32_t * pGWAcqOrderPref
- swi_uint256_t ParamPresenceMask

### 8.744.1 Detailed Description

This structure contains unpack get get system selection preferences parameters.

**Parameters**

| | |
|---|---|
| *pEmerMode* | • Optional parameter specifying the emergency Mode<br>• Values:<br>– 0 - OFF (normal)<br>– 1 - ON (Emergency) |

- ∗ - Bit to check in ParamPresenceMask - **16**

**Parameters**

| | |
|---|---|
| *pModePref* | – Optional parameter<br>– Bit Mask indicating the radio technology mode preference<br>– Bit values:<br>∗ Bit 0 - cdma2000 1x<br>∗ Bit 1 - cdma2000 HRPD(1xEV-DO)<br>∗ Bit 2 - GSM<br>∗ Bit 3 - UMTS<br>∗ Bit 4 - LTE |

- ∗ - Bit to check in ParamPresenceMask - **17**

**Parameters**

| | |
|---|---|
| *pBandPref* | – Optional parameter<br>– Bit mask representing the band preference<br>– Bit values:<br>* Bit 0 - Band Class 0, A-System<br>* Bit 1 - Band Class 0, B-System, Band Class 0 AB, GSM 850 Band<br>* Bit 2 - Band Class 1, all blocks<br>* Bit 3 - Band Class 2 place holder<br>* Bit 4 - Band Class 3, A-System<br>* Bit 5 - Band Class 4, all blocks<br>* Bit 6 - Band Class 5, all blocks<br>* Bit 7 - GSM_DCS_1800 band<br>* Bit 8 - GSM Extended GSM (E-GSM) 900 band<br>* Bit 9 - GSM Primary GSM (P-GSM) 900 band<br>* Bit 10 - Band Class 6<br>* Bit 11 - Band Class 7<br>* Bit 12 - Band Class 8<br>* Bit 13 - Band Class 9<br>* Bit 14 - Band Class 10<br>* Bit 15 - Band Class 11<br>* Bit 16 - GSM 450 band<br>* Bit 17 - GSM 480 band<br>* Bit 18 - GSM 750 band<br>* Bit 19 - GSM 850 band<br>* Bit 20 - GSM Railways GSM 900 Band<br>* Bit 21 - GSM PCS 1900 band<br>* Bit 22 - WCDMA Europe, Japan, and China IMT 2100 band<br>* Bit 23 - WCDMA U.S. PCS 1900 band<br>* Bit 24 - WCDMA Europe and China DCS 1800 band<br>* Bit 25 - WCDMA U.S. 1700 band<br>* Bit 26 - WCDMA U.S. 850 band<br>* Bit 27 - WCDMA Japan 800 band<br>* Bit 28 - Band Class 12<br>* Bit 29 - Band Class 14<br>* Bit 30 - Reserved<br>* Bit 31 - Band Class 15<br>* Bit 32 to 47 - Reserved<br>* Bit 48 - WCDMA Europe 2600 band<br>* Bit 49 - WCDMA Europe and Japan 900 band<br>* Bit 50 - WCDMA Japan 1700 band<br>* Bit 51 to 55 - Reserved<br>* Bit 56 - Band Class 16<br>* Bit 57 - Band Class 17<br>* Bit 58 - Band Class 18<br>* Bit 59 - Band Class 19<br>* Bit 60 to 64 - Reserved |

- $*$ - Bit to check in ParamPresenceMask - **18**

**Parameters**

| | |
|---|---|
| *pPRLPref* | – Optional parameter indicating the CDMA PRL Preference<br>– Values:<br>* 0x0001 - Acquire available system only on the A side<br>* 0x0002 - Acquire available system only on the B side<br>* 0x3FFF - Acquire any available systems |

- ∗ - Bit to check in ParamPresenceMask - **19**

**Parameters**

| | |
|---|---|
| *pRoamPref* | – Optional parameter indicating the roaming Preference<br>– Values:<br>* 0x01 - Acquire only systems for which the roaming indicator is off<br>* 0x02 - Acquire a system as long as its roaming indicator is not off<br>* 0x03 - Acquire only systems for which the roaming indicator is off or solid on, i.e. not flashing; CDMA only<br>* 0xFF - Acquire systems, regardless of their roaming indicator |

- ∗ - Bit to check in ParamPresenceMask - **20**

**Parameters**

| | |
|---|---|
| *pLTEBandPref* | – Optional parameter<br>– Bit mask representing the LTE band preference<br>– Bit Values<br>* Bit 0 - E-UTRA Operating Band 1<br>* Bit 1 - E-UTRA Operating Band 2<br>* Bit 2 - E-UTRA Operating Band 3<br>* Bit 3 - E-UTRA Operating Band 4<br>* Bit 4 - E-UTRA Operating Band 5<br>* Bit 5 - E-UTRA Operating Band 6<br>* Bit 6 - E-UTRA Operating Band 7<br>* Bit 7 - E-UTRA Operating Band 8<br>* Bit 8 - E-UTRA Operating Band 9<br>* Bit 9 - E-UTRA Operating Band 10<br>* Bit 10 - E-UTRA Operating Band 11<br>* Bit 11 - E-UTRA Operating Band 12<br>* Bit 12 - E-UTRA Operating Band 13<br>* Bit 13 - E-UTRA Operating Band 14<br>* Bit 16 - E-UTRA Operating Band 17<br>* Bit 17 - E-UTRA Operating Band 18<br>* Bit 18 - E-UTRA Operating Band 19<br>* Bit 19 - E-UTRA Operating Band 20<br>* Bit 20 - E-UTRA Operating Band 21<br>* Bit 32 - E-UTRA Operating Band 33<br>* Bit 33 - E-UTRA Operating Band 34<br>* Bit 34 - E-UTRA Operating Band 35<br>* Bit 35 - E-UTRA Operating Band 36<br>* Bit 36 - E-UTRA Operating Band 37<br>* Bit 37 - E-UTRA Operating Band 38<br>* Bit 38 - E-UTRA Operating Band 39<br>* Bit 39 - E-UTRA Operating Band 40<br>* All other bits are reserved |

- ∗ - Bit to check in ParamPresenceMask - **21**

**Parameters**

| | |
|---|---|
| *pNetSelPref* | – Optional parameter indicating network selection preference<br>– Values:<br>* 0x00 - Automatic network selection<br>* 0x01 - Manual network selection. |

- ∗ - Bit to check in ParamPresenceMask - **22**

**Parameters**

| | |
|---|---|
| *pSrvDomainPref* | – Optional parameter indicating Service domain preference<br>– Values:<br>* 0x00 - Circuit switched only<br>* 0x01 - Packet switched only<br>* 0x02 - Circuit switched and packet switched<br>* 0x03 - Packet switched attach<br>* 0x04 - Packet switched detach |

- ∗ - Bit to check in ParamPresenceMask - **24**

**Parameters**

| | |
|---|---|
| *pGWAcqOrder-Pref* | – Optional parameter indicating GSM/WCDMA Acquisition order Preference<br>– Values:<br>* 0x00 - Automatic<br>* 0x01 - GSM then WCDMA<br>* 0x02 - WCDMA then GSM |

- ∗ - Bit to check in ParamPresenceMask - **25**

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | – bitmask representation to indicate valid parameters. |

### 8.744.2 Field Documentation

#### 8.744.2.1 swi_uint256_t unpack_nas_SLQSGetSysSelectionPref_t::ParamPresenceMask

#### 8.744.2.2 uint64_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pBandPref

#### 8.744.2.3 uint8_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pEmerMode

#### 8.744.2.4 uint32_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pGWAcqOrderPref

#### 8.744.2.5 uint64_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pLTEBandPref

#### 8.744.2.6 uint16_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pModePref

#### 8.744.2.7 uint8_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pNetSelPref

#### 8.744.2.8 uint16_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pPRLPref

#### 8.744.2.9 uint16_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pRoamPref

#### 8.744.2.10 uint32_t∗ unpack_nas_SLQSGetSysSelectionPref_t::pSrvDomainPref

## 8.745 unpack_nas_SLQSGetSysSelectionPrefExt_t Struct Reference

### Data Fields

- nas_EmerModeTlv * pEmerMode
- nas_ModePrefTlv * pModePref
- nas_BandPrefTlv * pBandPref
- nas_PRLPrefTlv * pPRLPref
- nas_RoamPrefTlv * pRoamPref
- nas_LTEBandPrefTlv * pLTEBandPref
- nas_NetSelPrefTlv * pNetSelPref
- nas_SrvDomainPrefTlv * pSrvDomainPref
- nas_GWAcqOrderPrefTlv * pGWAcqOrderPref
- nas_AcqOrderPrefTlv * pAcqOrderPref
- nas_RatDisabledMaskTlv * pRatDisabledMask
- nas_CiotLteOpModePrefTlv * pCiotLteOpModePref
- nas_LteM1BandPrefTlv * pLteM1BandPref
- nas_LteNb1BandPrefTlv * pLteNb1BandPref
- nas_CiotAcqOrderPrefTlv * pCiotAcqOrderPref
- nas_NR5gBandPrefTlv * pNr5gBandPref

### 8.745.1 Detailed Description

Structure for storing the current preferred system selection settings for the device.

**Parameters**

| | |
|---|---|
| *pEmerMode* | • Optional parameter specifying the emergency Mode<br>• See nas_EmerModeTlv for more information |
| *pModePref* | • Optional parameterindicating the radio technology mode preference<br>• See nas_ModePrefTlv for more information |
| *pBandPref* | • Optional parameter representing the band preference<br>• See nas_BandPrefTlv for more information |
| *pPRLPref* | • Optional parameter indicating the CDMA PRL Preference<br>• See nas_PRLPrefTlv for more information |
| *pRoamPref* | • Optional parameter indicating the roaming Preference<br>• See nas_RoamPrefTlv for more information |
| *pLTEBandPref* | • Optional parameter representing the LTE band preference<br>• See nas_LTEBandPrefTlv for more information |
| *pNetSelPref* | • Optional parameter indicating network selection preference<br>• See nas_NetSelPrefTlv for more information |

| | |
|---|---|
| *pSrvDomainPref* | • Optional parameter indicating Service domain preference<br>• See nas_SrvDomainPrefTlv for more information |
| *pGWAcqOrder-Pref* | • Optional parameter indicating GSM/WCDMA Acquisition order Preference<br>• See nas_GWAcqOrderPrefTlv for more information |
| *pAcqOrderPref* | • Optional parameter<br>• Acquisition Order Preference<br>• See nas_AcqOrderPrefTlv for more information |
| *pRatDisabled-Mask* | • Optional parameter representing the radio technologies that are disabled.<br>• See nas_RatDisabledMaskTlv for more information |
| *pCiotLteOp-ModePref* | • Optional parameter<br>• CIOT LTE Operational Mode Preference<br>• See nas_CiotLteOpModePrefTlv for more information |
| *pLteM1BandPref* | • Optional parameter representing the LTE M1 band preference<br>• See nas_LteM1BandPrefTlv for more information |
| *pLteNb1Band-Pref* | • Optional parameter representing the LTE NB1 band preference.<br>• See nas_LteNb1BandPrefTlv for more information |
| *pCiotAcqOrder-Pref* | • Optional parameter indicating CIOT Acquisition Order Preference<br>• See nas_CiotAcqOrderPrefTlv for more information |
| *pNr5gBandPref* | • Optional parameter indicating NR5G Band Preference<br>• See nas_NR5gBandPrefTlv for more information |

## 8.745.2 Field Documentation

### 8.745.2.1 nas_AcqOrderPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pAcqOrderPref

### 8.745.2.2 nas_BandPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pBandPref

### 8.745.2.3 nas_CiotAcqOrderPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pCiotAcqOrderPref

### 8.745.2.4 nas_CiotLteOpModePrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pCiotLteOpModePref

### 8.745.2.5 nas_EmerModeTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pEmerMode

### 8.745.2.6 nas_GWAcqOrderPrefTlv* unpack_nas_SLQSGetSysSelectionPrefExt_t::pGWAcqOrderPref

#### 8.745.2.7 nas_LTEBandPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pLTEBandPref

#### 8.745.2.8 nas_LteM1BandPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pLteM1BandPref

#### 8.745.2.9 nas_LteNb1BandPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pLteNb1BandPref

#### 8.745.2.10 nas_ModePrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pModePref

#### 8.745.2.11 nas_NetSelPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pNetSelPref

#### 8.745.2.12 nas_NR5gBandPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pNr5gBandPref

#### 8.745.2.13 nas_PRLPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pPRLPref

#### 8.745.2.14 nas_RatDisabledMaskTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pRatDisabledMask

#### 8.745.2.15 nas_RoamPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pRoamPref

#### 8.745.2.16 nas_SrvDomainPrefTlv∗ unpack_nas_SLQSGetSysSelectionPrefExt_t::pSrvDomainPref

## 8.746 unpack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference

### Data Fields

- nas_namName ∗ pNAMNameInfo
- nas_dirNum ∗ pDirNum
- nas_homeSIDNID ∗ pHomeSIDNID
- nas_minBasedIMSI ∗ pMinBasedIMSI
- nas_trueIMSI ∗ pTrueIMSI
- nas_CDMAChannel ∗ pCDMAChannel
- swi_uint256_t ParamPresenceMask

### 8.746.1 Detailed Description

Structure for 3GPP2Subscription unpack.

**Parameters**

| | |
|---|---|
| *pNAMNameInfo* | [Optional]<br>• See nas_namName for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pDirNum* | [Optional]<br>• See nas_dirNum for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pHomeSIDNID* | [Optional]<br>• See nas_homeSIDNID for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *pMinBasedIMSI* | [Optional]<br>• See nas_minBasedIMSI for more information<br>• Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *pTrueIMSI* | [Optional]<br>• See nas_trueIMSI for more information<br>• Bit to check in ParamPresenceMask - **20** |
| *pCDMAChannel* | [Optional]<br>• See nas_CDMAChannel for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.746.2 Field Documentation

#### 8.746.2.1 swi_uint256_t unpack_nas_SLQSNasGet3GPP2Subscription_t::ParamPresenceMask

#### 8.746.2.2 nas_CDMAChannel∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel

#### 8.746.2.3 nas_dirNum∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum

#### 8.746.2.4 nas_homeSIDNID∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID

#### 8.746.2.5 nas_minBasedIMSI∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI

#### 8.746.2.6 nas_namName∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo

#### 8.746.2.7 nas_trueIMSI∗ unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI

## 8.747 unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference

### Data Fields

- nas_GERANInfo ∗ pGERANInfo
- nas_UMTSInfo ∗ pUMTSInfo
- nas_CDMAInfo ∗ pCDMAInfo
- nas_LTEInfoIntrafreq ∗ pLTEInfoIntrafreq
- nas_LTEInfoInterfreq ∗ pLTEInfoInterfreq
- nas_LTEInfoNeighboringGSM ∗ pLTEInfoNeighboringGSM
- nas_LTEInfoNeighboringWCDMA ∗ pLTEInfoNeighboringWCDMA
- uint32_t ∗ pUMTSCellID
- nas_WCDMAInfoLTENeighborCell ∗ pWCDMAInfoLTENeighborCell
- swi_uint256_t ParamPresenceMask

### 8.747.1 Detailed Description

This structure contains information about the Get Cell Location response parameters.

**Parameters**

| | |
|---|---|
| *pGERANInfo* | • See nas_GERANInfo for more information.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pUMTSInfo* | • See nas_UMTSInfo for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pCDMAInfo* | • See nas_CDMAInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *pLTEInfo-Intrafreq* | • See nas_LTEInfoIntrafreq for more information.<br>• Bit to check in ParamPresenceMask - **19** |
| *pLTEInfo-Interfreq* | • See nas_LTEInfoInterfreq for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *pLTEInfo-NeighboringGSM* | • See nas_LTEInfoNeighboringGSM for more information.<br>• Bit to check in ParamPresenceMask - **21** |
| *pLTEInfo-NeighboringWCDMA* | • See nas_LTEInfoNeighboringWCDMA for more information.<br>• Bit to check in ParamPresenceMask - **22** |
| *pUMTSCellID* | • Cell ID.<br>• 0xFFFFFFFF indicates cell ID information is not present.<br>• Bit to check in ParamPresenceMask - **23** |
| *pWCDMAInfoLTENeighborCell* | • See nas_WCDMAInfoLTENeighborCell for more information.<br>• Bit to check in ParamPresenceMask - **24** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.747.2 Field Documentation

#### 8.747.2.1 swi_uint256_t unpack_nas_SLQSNasGetCellLocationInfo_t::ParamPresenceMask

#### 8.747.2.2 nas_CDMAInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo

#### 8.747.2.3 nas_GERANInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo

#### 8.747.2.4 nas_LTEInfoInterfreq* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq

#### 8.747.2.5 nas_LTEInfoIntrafreq* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq

#### 8.747.2.6 nas_LTEInfoNeighboringGSM* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM

#### 8.747.2.7 nas_LTEInfoNeighboringWCDMA* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA

#### 8.747.2.8 uint32_t* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID

#### 8.747.2.9 nas_UMTSInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo

#### 8.747.2.10 nas_WCDMAInfoLTENeighborCell* unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell

## 8.748 unpack_nas_SLQSNASGeteDRXParams_t Struct Reference

### Data Fields

- uint8_t * pCycleLen
- uint8_t * pPagingTimeWindow
- uint8_t * pEdrxEnable
- swi_uint256_t ParamPresenceMask

### 8.748.1 Detailed Description

This structure contains the SLQSNASGeteDRXParams unpack parameters.

**Parameters**

| | |
|---|---|
| *pCycleLen* | [Optional]<br>• eDRX cycle length<br>• Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32<br>• Bit to check in ParamPresenceMask - **16** |
| *pPagingTime-Window* | [Optional]<br>• eDRX paging time window<br>• Value is per 3GPP TS 24.008 Rel-13 Section10.5.5.32<br>• Bit to check in ParamPresenceMask - **17** |
| *pEdrxEnable* | [Optional]<br>• eDRX Enable<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable<br>• Bit to check in ParamPresenceMask - **18** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.748.2 Field Documentation

#### 8.748.2.1 swi_uint256_t unpack_nas_SLQSNASGeteDRXParams_t::ParamPresenceMask

#### 8.748.2.2 uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pCycleLen

#### 8.748.2.3 uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pEdrxEnable

#### 8.748.2.4 uint8_t* unpack_nas_SLQSNASGeteDRXParams_t::pPagingTimeWindow

## 8.749 unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference

### Data Fields

- uint8_t * pColorCode
- swi_uint256_t ParamPresenceMask

### 8.749.1 Detailed Description

Structure for storing the current preferred system selection settings for the device.

**Parameters**

| | |
|---|---|
| *pColorCode* | [Optional]<br>• Color code value<br>• Color code corresponding to the sector to which the AT is sending the access probe<br>• See 3GPP2 C.S0024-B V3.0, Section 7.11.6.2.1 for more information.<br>– 0xFF - Not Available<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.749.2 Field Documentation

#### 8.749.2.1 swi_uint256_t unpack_nas_SLQSNasGetHDRColorCode_t::ParamPresenceMask

#### 8.749.2.2 uint8_t* unpack_nas_SLQSNasGetHDRColorCode_t::pColorCode

## 8.750 unpack_nas_SLQSNasGetRFInfo_t Struct Reference

### Data Fields

- nas_RfBandInfoList rfbandInfoList
- nas_RfDedicatedBandInfo * pRfDedicatedBandInfo
- nas_RfBandInfoExtFormat * pRfBandInfoExtFormat
- nas_RfBandwidthInfo * pRfBandwidthInfo
- nas_LTEOperationMode * pLTEOperationMode
- uint16_t Tlvresult

### 8.750.1 Detailed Description

This structure contains the SLQSNasGetRFInfo response parameters.

**Parameters**

| | |
|---|---|
| *rfbandInfoList* | • See nas_RfBandInfoList for more information |
| *pRfDedicated-BandInfo* | [Optional]<br>• See nas_RfDedicatedBandInfo for more information |

| | |
|---|---|
| *pRfBandInfoExt-Format* | [Optional]<br>• See nas_RfBandInfoExtFormat for more information |
| *pRfBandwidth-Info* | [Optional]<br>• See nas_RfBandwidthInfo for more information |
| *pLTEOpration-Mode* | [Optional]<br>• See nas_LTEOperationMode for more information |

### 8.750.2 Field Documentation

#### 8.750.2.1 nas_LTEOperationMode∗ unpack_nas_SLQSNasGetRFInfo_t::pLTEOperationMode

#### 8.750.2.2 nas_RfBandInfoExtFormat∗ unpack_nas_SLQSNasGetRFInfo_t::pRfBandInfoExtFormat

#### 8.750.2.3 nas_RfBandwidthInfo∗ unpack_nas_SLQSNasGetRFInfo_t::pRfBandwidthInfo

#### 8.750.2.4 nas_RfDedicatedBandInfo∗ unpack_nas_SLQSNasGetRFInfo_t::pRfDedicatedBandInfo

#### 8.750.2.5 nas_RfBandInfoList unpack_nas_SLQSNasGetRFInfo_t::rfbandInfoList

#### 8.750.2.6 uint16_t unpack_nas_SLQSNasGetRFInfo_t::TlvresultI

## 8.751 unpack_nas_SLQSNasGetSigInfo_t Struct Reference

### Data Fields

- cdmaSSInfo CDMASSInfo
- hdrSSInfo HDRSSInfo
- int8_t GSMSSInfo
- cdmaSSInfo WCDMASSInfo
- lteSSInfo LTESSInfo
- swi_uint256_t ParamPresenceMask

### 8.751.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

**Parameters**

| | |
|---|---|
| *CDMASSInfo* | • See cdmaSSInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *HDRSSInfo* | • See hdrSSInfo for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *GSMSSInfo* | • GSM signal strength is the RSSI in dBm (signed value).<br>• A value of -125 dBm or lower is used to indicate No Signal.<br>• Bit to check in ParamPresenceMask - **18** |

| | |
|---|---|
| *WCDMASSInfo* | • See cdmaSSInfo for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *LTESSInfo* | • See lteSSInfo for more information<br>• Bit to check in ParamPresenceMask - **20** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.751.2 Field Documentation

#### 8.751.2.1 **cdmaSSInfo** unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo

#### 8.751.2.2 int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo

#### 8.751.2.3 **hdrSSInfo** unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo

#### 8.751.2.4 **lteSSInfo** unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo

#### 8.751.2.5 **swi_uint256_t** unpack_nas_SLQSNasGetSigInfo_t::ParamPresenceMask

#### 8.751.2.6 **cdmaSSInfo** unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo

## 8.752 unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference

### Data Fields

- nas_rxInfo * pRXChain0Info
- nas_rxInfo * pRXChain1Info
- nas_txInfo * pTXInfo
- swi_uint256_t ParamPresenceMask

### 8.752.1 Detailed Description

Structure for TxRx Info unpack

**Parameters**

| | |
|---|---|
| *pRXChain0Info* | [Optional]<br>• See nas_rxInfo for more information.<br>• Bit to check in ParamPresenceMask - **16** |
| *pRXChain1Info* | [Optional]<br>• See nas_rxInfo for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pTXInfo* | [Optional]<br>• See nas_txInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.752.2 Field Documentation

#### 8.752.2.1 swi_uint256_t unpack_nas_SLQSNasGetTxRxInfo_t::ParamPresenceMask

#### 8.752.2.2 nas_rxInfo∗ unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info

#### 8.752.2.3 nas_rxInfo∗ unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info

#### 8.752.2.4 nas_txInfo∗ unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo

## 8.753 unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference

### Data Fields

- nas_UniversalTime universalTime
- uint8_t ∗ pTimeZone
- uint8_t ∗ pDayltSavAdj
- uint8_t ∗ pRadioInterface
- swi_uint256_t ParamPresenceMask

### 8.753.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

**Parameters**

| | |
|---|---|
| *universalTime* | • See nas_UniversalTime for more information.<br>• Bit to check in ParamPresenceMask - **1** |
| *pTimeZone* | • Time Zone.<br>• Offset from Universal time, i.e., the difference between local time and Universal time, in increments of 15 min (signed value).<br>• Bit to check in ParamPresenceMask - **16** |
| *pDayltSavAdj* | • Daylight Saving Adjustment.<br>• Daylight saving adjustment in hr.<br>  – Possible values: 0, 1, and 2.<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *pRadioInterface* | • Radio interface from which the information comes<br>• Values<br>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X<br>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - cdma2000 HRPD (1xEV-DO)<br>– 0x04 - NAS_RADIO_IF_GSM - GSM<br>– 0x05 - NAS_RADIO_IF_UMTS - UMTS<br>– 0x08 - NAS_RADIO_IF_LTE - LTE<br>– 0x09 - NAS_RADIO_IF_TDSCDMA -TD-SCDMA<br>• Bit to check in ParamPresenceMask - **18** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.753.2 Field Documentation

#### 8.753.2.1 swi_uint256_t unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::ParamPresenceMask

#### 8.753.2.2 uint8_t∗ unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj

#### 8.753.2.3 uint8_t∗ unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface

#### 8.753.2.4 uint8_t∗ unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone

#### 8.753.2.5 nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime

## 8.754 unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference

### Data Fields

- cdmaSSInfo ∗ pCDMASigInfo
- hdrSSInfo ∗ pHDRSigInfo
- int8_t ∗ pGSMSigInfo
- cdmaSSInfo ∗ pWCDMASigInfo
- lteSSInfo ∗ pLTESigInfo
- int8_t ∗ pRscp
- tdscdmaSigInfoExt ∗ pTDSCDMASigInfoExt
- swi_uint256_t ParamPresenceMask

### 8.754.1 Detailed Description

Structure for storing the nasSigInfo indication parameters.

**Parameters**

| | |
|---|---|
| *pCDMASigInfo* | • See cdmaSSInfo for more information.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pHDRSigInfo* | • See hdrSSInfo for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pGSMSigInfo* | • one byte value, GSM signal strength is the RSSI in dBm (signed value). A value of -125 dBm or lower is used to indicate No Signal<br>• Bit to check in ParamPresenceMask - **18** |
| *pWCDMASigInfo* | • See cdmaSSInfo for more information.<br>• Bit to check in ParamPresenceMask - **19** |
| *pLTESigInfo* | • See lteSSInfo for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *pRscp* | • RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. Measurement range: -120 dBm to -25 dBm.<br>• Bit to check in ParamPresenceMask - **21** |
| *pTDSCDMASig-InfoExt* | • See tdscdmaSigInfoExt for more information.<br>• Bit to check in ParamPresenceMask - **22** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.754.2 Field Documentation

#### 8.754.2.1 swi_uint256_t unpack_nas_SLQSNasSigInfoCallback_ind_t::ParamPresenceMask

#### 8.754.2.2 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pCDMASigInfo

#### 8.754.2.3 int8_t∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pGSMSigInfo

#### 8.754.2.4 hdrSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo

#### 8.754.2.5 lteSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo

#### 8.754.2.6 int8_t∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp

#### 8.754.2.7 tdscdmaSigInfoExt∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt

#### 8.754.2.8 cdmaSSInfo∗ unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo

## 8.755 unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference

### Data Fields

- nas_wcdmaUARFCN ∗ pWcdmaUARFCN

- nas_lteEARFCN * pLteEARFCN
- nas_ltePCI * pLtePCI
- swi_uint256_t ParamPresenceMask

### 8.755.1 Detailed Description

Structure for Get Channel Lock unpack.

**Parameters**

| | |
|---|---|
| *pWcdmaUARFCN* | [Optional]<br>• See nas_wcdmaUARFCN for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pLteEARFCN* | [Optional]<br>• See nas_lteEARFCN for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pLtePCI* | [Optional]<br>• See nas_ltePCI for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.755.2 Field Documentation

#### 8.755.2.1 swi_uint256_t unpack_nas_SLQSNASSwiGetChannelLock_t::ParamPresenceMask

#### 8.755.2.2 nas_lteEARFCN* unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN

#### 8.755.2.3 nas_ltePCI* unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI

#### 8.755.2.4 nas_wcdmaUARFCN* unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN

## 8.756 unpack_nas_SLQSNasSwiModemStatus_t Struct Reference

### Data Fields

- nas_CommInfo commonInfo
- nas_LTEInfo * pLTEInfo
- swi_uint256_t ParamPresenceMask

### 8.756.1 Detailed Description

Structure for storing the SLQS Nas Swi Modem Status response parameters.

**Parameters**

| | |
|---|---|
| *commonInfo* | (mandatory)<br>• See nas_CommInfo for more information<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *pLTEInfo* | (optional)<br>• See nas_LTEInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.756.2 Field Documentation

#### 8.756.2.1 **nas_CommInfo** unpack_nas_SLQSNasSwiModemStatus_t::commonInfo

#### 8.756.2.2 **swi_uint256_t** unpack_nas_SLQSNasSwiModemStatus_t::ParamPresenceMask

#### 8.756.2.3 **nas_LTEInfo*** unpack_nas_SLQSNasSwiModemStatus_t::pLTEInfo

## 8.757 unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference

### Data Fields

- NASQmiCbkNasSwiOTAMessageInd Info
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.757.1 Detailed Description

This structure contains unpack OTA message indication parameters.

**Parameters**

| | |
|---|---|
| *Info* | • Structure used to store all QMI Notification Info.<br>– See NASQmiCbkNasSwiOTAMessageInd for more details |
| *Tlvresult* | • unpack result<br>• Bit to check in ParamPresenceMask - **2** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.757.2 Field Documentation

#### 8.757.2.1 **NASQmiCbkNasSwiOTAMessageInd** unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info

#### 8.757.2.2 **swi_uint256_t** unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::ParamPresenceMask

#### 8.757.2.3 uint16_t unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult

## 8.758 unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference

### Data Fields

- char t3396_apn [101]
- uint8_t t3396_plmn_id [3]
- uint32_t t3396_val
- swi_uint256_t ParamPresenceMask

### 8.758.1 Detailed Description

Structure for Network Timer indication parameters.

**Parameters**

| | |
|---|---|
| *t3396_apn* | • apn<br>• Bit to check in ParamPresenceMask - **16** |
| *t3396_plmn_id* | • plmn id<br>• Bit to check in ParamPresenceMask - **16** |
| *t3396_val* | • timer value<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.758.2 Field Documentation

#### 8.758.2.1 swi_uint256_t unpack_nas_SLQSNasTimerCallback_ind_t::ParamPresenceMask

#### 8.758.2.2 char unpack_nas_SLQSNasTimerCallback_ind_t::t3396_apn[101]

#### 8.758.2.3 uint8_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_plmn_id[3]

#### 8.758.2.4 uint32_t unpack_nas_SLQSNasTimerCallback_ind_t::t3396_val

## 8.759 unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference

### Data Fields

- NASQmiCbkNasSystemSelPrefInd Info
- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.759.1 Detailed Description

This structure contains unpack set system selection preference callback indication parameters.

**Parameters**

| | |
|---|---|
| *Info* | • Structure used to store all QMI Notification Info.<br>– See NASQmiCbkNasSystemSelPrefInd for more details |
| *Tlvresult* | • unpack result<br>• Bit to check in ParamPresenceMask - **2** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.759.2 Field Documentation

#### 8.759.2.1 NASQmiCbkNasSystemSelPrefInd unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info

#### 8.759.2.2 swi_uint256_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::ParamPresenceMask

#### 8.759.2.3 uint16_t unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult

## 8.760 unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference

### Data Fields

- uint16_t ∗ pCurrentPersonality
- uint8_t ∗ pPersonalityListLength
- nas_protocolSubtypeElement ∗ pProtocolSubtypeElement
- swi_uint256_t ParamPresenceMask

### 8.760.1 Detailed Description

Structure for HDRPersonality unpack.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality[Out]* | • Current active personality index.<br>• Bit to check in ParamPresenceMask - **16** |
| *pPersonalityList-Length[In/Out]* | • Number of Personality Protocol Subtype contains in this response.<br>• maximum input value is 3<br>• Bit to check in ParamPresenceMask - **17** |
| *pProtocol-Subtype-Element[Out]* | • See nas_protocolSubtypeElement for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.760.2 Field Documentation

#### 8.760.2.1 swi_uint256_t unpack_nas_SLQSSwiGetHDRPersonality_t::ParamPresenceMask

#### 8.760.2.2 uint16_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pCurrentPersonality

#### 8.760.2.3 uint8_t* unpack_nas_SLQSSwiGetHDRPersonality_t::pPersonalityListLength

#### 8.760.2.4 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRPersonality_t::pProtocolSubtypeElement

## 8.761 unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference

### Data Fields

- uint16_t * pCurrentPrsnlty
- uint8_t * pPersonalityListLength
- nas_protocolSubtypeElement * pProtoSubTypElmnt
- uint64_t * pAppSubType
- swi_uint256_t ParamPresenceMask

### 8.761.1 Detailed Description

Structure for HDRProtSubtype unpack.

**Parameters**

| | |
|---|---|
| *pCurrent-Personality* | • Current active personality index.<br>• Bit to check in ParamPresenceMask - **16** |
| *pPersonalityList-Length* | • Number of Personality Protocol Subtype contains in this response.<br>• maximum input value is 4<br>• Bit to check in ParamPresenceMask - **17** |
| *pProtocol-SubtypeElement* | • See nas_protocolSubtypeElement for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pAppSubType* | • Stream application subtype<br>• Application subtype for each stream<br>• Bit to check in ParamPresenceMask - **18** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.761.2 Field Documentation

#### 8.761.2.1 uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType

#### 8.761.2.2 swi_uint256_t unpack_nas_SLQSSwiGetHDRProtSubtype_t::ParamPresenceMask

#### 8.761.2.3 uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty

#### 8.761.2.4 uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength

#### 8.761.2.5 nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypElmnt

## 8.762 unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference

### Data Fields

- nas_DRCParams * pDRCParams
- uint8_t * pUATI
- nas_PilotSetData * pPilotSetData
- swi_uint256_t ParamPresenceMask

### 8.762.1 Detailed Description

Structure for HRPD Stats unpack.

**Parameters**

| | |
|---|---|
| *pDRCParams* | • See nas_DRCParams for more information.<br>• Bit to check in ParamPresenceMask - **16** |
| *pUATI* | • A 128-bit address that includes the access terminal identifier and subnet ID<br>• Size must be 16 bytes<br>• Bit to check in ParamPresenceMask - **17** |
| *pPilotSetData* | • See nas_PilotSetData for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.762.2 Field Documentation

#### 8.762.2.1 swi_uint256_t unpack_nas_SLQSSwiGetHRPDStats_t::ParamPresenceMask

#### 8.762.2.2 nas_DRCParams* unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams

#### 8.762.2.3 nas_PilotSetData* unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData

#### 8.762.2.4 uint8_t* unpack_nas_SLQSSwiGetHRPDStats_t::pUATI

## 8.763 unpack_nas_SLQSSwiGetLteCQI_t Struct Reference

### Data Fields

- uint8_t ValidityCW0
- uint8_t CQIValueCW0
- uint8_t ValidityCW1
- uint8_t CQIValueCW1
- swi_uint256_t ParamPresenceMask

## 8.763.1 Detailed Description

This structure contains paramaters unpack fetch CQI parameters for LTE data session.

**Parameters**

| | |
|---|---|
| *ValidityCW0[OUT]* | • Values<br>– 0- Invalid.<br>– 1- Valid.<br>• Bit to check in ParamPresenceMask - **1** |
| *CQIValueCW0[OUT]* | • Values<br>– Range 0∼15<br>• Bit to check in ParamPresenceMask - **1** |
| *ValidityCW1[OUT]* | • Values<br>– 0- Invalid.<br>– 1- Valid.<br>• Bit to check in ParamPresenceMask - **1** |
| *CQIValueCW1[OUT]* | • Values<br>– Range 0∼15<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

## 8.763.2 Field Documentation

### 8.763.2.1 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0

### 8.763.2.2 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1

### 8.763.2.3 swi_uint256_t unpack_nas_SLQSSwiGetLteCQI_t::ParamPresenceMask

### 8.763.2.4 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0

### 8.763.2.5 uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1

## 8.764 unpack_nas_SLQSSwiGetLteSccRxInfo_t Struct Reference

### Data Fields

- nas_SccRxInfo * pSccRxInfo
- swi_uint256_t ParamPresenceMask

### 8.764.1 Detailed Description

This structure contains unpack get LTE Secondary carrier Rx signal level information parameters.

**Parameters**

| | |
|---|---|
| *pSccRxInfo* | Secondary carrier Rx signal level info<br>• See nas_SccRxInfo<br>– Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.764.2 Field Documentation

#### 8.764.2.1 swi_uint256_t unpack_nas_SLQSSwiGetLteSccRxInfo_t::ParamPresenceMask

#### 8.764.2.2 nas_SccRxInfo* unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo

## 8.765 unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference

### Data Fields

- nas_HDRPersonality_Ind_Data * pHDRPersInd
- swi_uint256_t ParamPresenceMask

### 8.765.1 Detailed Description

Structure for HDR Personality indication unpack.

**Parameters**

| | |
|---|---|
| *pHDRPersInd* | • See nas_HDRPersonality_Ind_Data for more information. |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.765.2 Field Documentation

#### 8.765.2.1 swi_uint256_t unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::ParamPresenceMask

#### 8.765.2.2 nas_HDRPersonality_Ind_Data* unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::pHDRPersInd

## 8.766 unpack_nas_SLQSSwiNetworkDebug_t Struct Reference

### Data Fields

- uint8_t * pObjectVer
- nas_NetworkStat1x * pNetworkStat1x
- nas_NetworkStatEVDO * pNetworkStatEVDO
- nas_DeviceConfigDetail * pDeviceConfigDetail
- nas_DataStatusDetail * pDataStatusDetail
- swi_uint256_t ParamPresenceMask

### 8.766.1 Detailed Description

Structure for SwiNetworkDebug unpack.

**Parameters**

| | |
|---|---|
| *pObjectVer* | • Object's version number for the host to handle<br>  – 0xFF - NA<br>  – Others - shows in decimal<br>• Bit to check in ParamPresenceMask - **16** |
| *pNetworkStat1x* | • See nas_NetworkStat1x for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pNetworkStatEVDO* | • See nas_NetworkStatEVDO for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *pDeviceConfigDetail* | • See nas_DeviceConfigDetail for more information.<br>• Bit to check in ParamPresenceMask - **19** |
| *pDataStatusDetail* | • See nas_DataStatusDetail for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.766.2 Field Documentation

#### 8.766.2.1 swi_uint256_t unpack_nas_SLQSSwiNetworkDebug_t::ParamPresenceMask

#### 8.766.2.2 nas_DataStatusDetail* unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail

#### 8.766.2.3 nas_DeviceConfigDetail* unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail

#### 8.766.2.4 nas_NetworkStat1x* unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x

#### 8.766.2.5 nas_NetworkStatEVDO* unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO

#### 8.766.2.6 uint8_t* unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer

## 8.767 unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t Struct Reference

### Data Fields

- nas_RankIndicatorTlv rankIndicatorTlv
- swi_uint256_t ParamPresenceMask

### 8.767.1 Detailed Description

Structure for Rank Indicator indication unpack.

**Parameters**

| | |
|---|---|
| *rankIndicatorTlv* | • See nas_RankIndicatorTlv for more information.<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.767.2 Field Documentation

#### 8.767.2.1 swi_uint256_t unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::ParamPresenceMask

#### 8.767.2.2 nas_RankIndicatorTlv unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::rankIndicatorTlv

## 8.768 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
- swi_uint256_t ParamPresenceMask

### 8.768.1 Detailed Description

This structure contains unpack system information callback indication parameters.

**Parameters**

| | |
|---|---|
| *pCDMASrvStatusInfo* | • See nas_SrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **16** |
| *pHDRSrvStatusInfo* | • See nas_SrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pGSMSrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *pWCDMASrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **19** |
| *pLTESrvStatusInfo* | • See nas_GSMSrvStatusInfo for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *pCDMASysInfo* | • See nas_CDMASysInfo for more information.<br>• Bit to check in ParamPresenceMask - **21** |
| *pHDRSysInfo* | • See nas_HDRSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **22** |
| *pGSMSysInfo* | • See nas_GSMSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **23** |
| *pWCDMASysInfo* | • See nas_WCDMASysInfo for more information.<br>• Bit to check in ParamPresenceMask - **24** |
| *pLTESysInfo* | • See nas_LTESysInfo for more information.<br>• Bit to check in ParamPresenceMask - **25** |
| *pAddCDMASysInfo* | • See nas_AddCDMASysInfo for more information.<br>• Bit to check in ParamPresenceMask - **26** |

| | |
|---|---|
| *pAddHDRSysInfo* | • System table index referencing the beginning of the geo in which the current serving system is present.<br>• When the system index is not known, 0xFFFF is used.<br>• Bit to check in ParamPresenceMask - **27** |
| *pAddGSMSysInfo* | • See nas_AddSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **28** |
| *pAddWCDMASysInfo* | • See nas_AddSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **29** |
| *pAddLTESysInfo* | • System table index referencing the beginning of the geo in which the current serving system is present.<br>• When the system index is not known, 0xFFFF is used.<br>• Bit to check in ParamPresenceMask - **30** |
| *pGSMCallBarringSysInfo* | • See nas_CallBarringSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **31** |
| *pWCDMACallBarringSysInfo* | • See nas_CallBarringSysInfo for more information.<br>• Bit to check in ParamPresenceMask - **32** |
| *pLTEVoiceSupportSysInfo* | • Indicates voice support status on LTE.<br>– 0x00 - Voice is not supported<br>– 0x01 - Voice is supported<br>• Bit to check in ParamPresenceMask - **33** |
| *pGSMCipherDomainSysInfo* | • Ciphering on the service domain.<br>– 0x00 - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>• Bit to check in ParamPresenceMask - **34** |
| *pWCDMACipherDomainSysInfo* | • Ciphering on the service domain.<br>– 0x00 - No service<br>– 0x01 - Circuit-switched only<br>– 0x02 - Packet-switched only<br>– 0x03 - Circuit-switched and packet-switched<br>• Bit to check in ParamPresenceMask - **35** |

| | |
|---|---|
| *pSysInfoNo-Change* | • System Info No Change.<br>• Flag used to notify clients that a request to select a network ended with no change in the PLMN.<br>  – 0x01 - No change in system information<br>• Bit to check in ParamPresenceMask - **36** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.768.2 Field Documentation

#### 8.768.2.1 nas_AddCDMASysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddCDMASysInfo

#### 8.768.2.2 nas_AddSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddGSMSysInfo

#### 8.768.2.3 uint16_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddHDRSysInfo

#### 8.768.2.4 uint16_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddLTESysInfo

#### 8.768.2.5 nas_AddSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo

#### 8.768.2.6 swi_uint256_t unpack_nas_SLQSSysInfoCallback_ind_t::ParamPresenceMask

#### 8.768.2.7 nas_SrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo

#### 8.768.2.8 nas_CDMASysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo

#### 8.768.2.9 nas_CallBarringSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo

#### 8.768.2.10 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo

#### 8.768.2.11 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo

#### 8.768.2.12 nas_GSMSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo

#### 8.768.2.13 nas_SrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSrvStatusInfo

#### 8.768.2.14 nas_HDRSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pHDRSysInfo

#### 8.768.2.15 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo

#### 8.768.2.16 nas_LTESysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo

#### 8.768.2.17 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo

#### 8.768.2.18 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange

#### 8.768.2.19 nas_CallBarringSysInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo

#### 8.768.2.20 uint8_t∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo

#### 8.768.2.21 nas_GSMSrvStatusInfo∗ unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo

#### 8.768.2.22 nas_WCDMASysInfo* unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo

## 8.769 unpack_omaDmConfigTlv_t Struct Reference

### Data Fields

- uint8_t state
- uint8_t userInputReq
- uint16_t userInputTimeout
- uint16_t alertmsglength
- uint8_t alertmsg [256]

### 8.769.1 Detailed Description

This structure will hold the SwiOmaDmConfig session parameters information.

**Parameters**

| | |
|---|---|
| *state* | • 0x01 - OMA-DM Read Request<br>• 0x02 - OMA-DM Change Request<br>• 0x03 - OMA-DM Config Complete |
| *userInputReq* | - Bit mask of available user inputs<br>• 0x00 - No user input required.Informational indication<br>• 0x01 - Accept<br>• 0x02 - Reject |
| *userInput-Timeout* | • Timeout for user input in minutes. A value of 0 means no time-out |
| *alertmsglength* | • Length of Alert message string in bytes |
| *alertmsg* | • Alert message in UCS2 (Max 256 characters) |

### 8.769.2 Field Documentation

#### 8.769.2.1 uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]

#### 8.769.2.2 uint16_t unpack_omaDmConfigTlv_t::alertmsglength

#### 8.769.2.3 uint8_t unpack_omaDmConfigTlv_t::state

#### 8.769.2.4 uint8_t unpack_omaDmConfigTlv_t::userInputReq

#### 8.769.2.5 uint16_t unpack_omaDmConfigTlv_t::userInputTimeout

## 8.770 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

### 8.770.1 Detailed Description

This structure will hold the SwiOmaDmFota session parameters information.

**Parameters**

| | |
|---|---|
| *state* | • 0x01 - No Firmware available<br>• 0x02 - Query Firmware Download<br>• 0x03 - Firmware Downloading<br>• 0x04 - Firmware downloaded<br>• 0x05 - Query Firmware Update<br>• 0x06 - Firmware updating<br>• 0x07 - Firmware updated |
| *userInputReq* | - Bit mask of available user inputs<br>• 0x00 - No user input required.Informational indication<br>• 0x01 - Accept<br>• 0x02 - Reject |
| *userInput-Timeout* | • Timeout for user input in minutes. A value of 0 means no time-out |
| *fwdloadsize* | • The size (in bytes) of the firmware update package |
| *fwloadComplete* | • The number of bytes downloaded. Need to determine how often to send this message for progress bar notification. Every 500ms or 5% increment. |
| *update-CompleteStatus* | • This field should be looked at only when the OMADM session is complete.<br>• See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h for update complete status. |
| *severity* | • 0x01 - Mandatory<br>• 0x02 - Optional |

| | |
|---|---|
| *versionlength* | • Length of FW Version string in bytes |
| *version* | • FW Version string in ASCII (Max 256 characters) |
| *namelength* | • Length Package Name string in bytes |
| *package_name* | • Package Name in UCS2 (Max 256 characters) |
| *descriptionlength* | • Length of description in bytes |
| *description* | • Description of Update Package in USC2 (Max 256 characters) |
| *sessionType* | • 0x00 - Client initiated<br>• 0x01 - Network initiated |

## 8.770.2 Field Documentation

### 8.770.2.1 uint8_t unpack_omaDmFotaTlv_t::description[256]

### 8.770.2.2 uint16_t unpack_omaDmFotaTlv_t::descriptionlength

### 8.770.2.3 uint32_t unpack_omaDmFotaTlv_t::fwdloadsize

### 8.770.2.4 uint32_t unpack_omaDmFotaTlv_t::fwloadComplete

### 8.770.2.5 uint16_t unpack_omaDmFotaTlv_t::namelength

### 8.770.2.6 uint8_t unpack_omaDmFotaTlv_t::package_name[256]

### 8.770.2.7 uint8_t unpack_omaDmFotaTlv_t::sessionType

### 8.770.2.8 uint8_t unpack_omaDmFotaTlv_t::severity

### 8.770.2.9 uint8_t unpack_omaDmFotaTlv_t::state

### 8.770.2.10 uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus

### 8.770.2.11 uint8_t unpack_omaDmFotaTlv_t::userInputReq

### 8.770.2.12 uint16_t unpack_omaDmFotaTlv_t::userInputTimeout

### 8.770.2.13 uint8_t unpack_omaDmFotaTlv_t::version[256]

### 8.770.2.14 uint16_t unpack_omaDmFotaTlv_t::versionlength

## 8.771 unpack_omaDmNotificationsTlv_t Struct Reference

### Data Fields

- uint8_t notification
- uint16_t sessionStatus

### 8.771.1 Detailed Description

This structure will hold the SwiOmaDmConfig session notification parameters information.

**Parameters**

| | |
|---|---|
| *notification* | • 0x01 - GPS settings change<br>• 0x02 - Device reset<br>• 0x03 - Device factory reset<br>• 0x04 - CI-DC Session start<br>• 0x05 - CI-DC Session end<br>• 0x06 - CI-PRL Session start<br>• 0x07 - CI-PRL Session end<br>• 0x08 - CI-FUMO Session start<br>• 0x09 - CI-FUMO session end<br>• 0x0A - HFA-DC Session start<br>• 0x0B - HFA-DC Session end<br>• 0x0C - HFA-PRL Session start<br>• 0x0D - HFA-PRL Session end<br>• 0x0E - HFA-FUMO Session start<br>• 0x0F - HFA-FUMO session end<br>• 0x10 - NI Session start<br>• 0x11 - NI session end |
| *sessionStatus* | • This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications<br>• See qaGobiApiTableSwiOMADMSessionStatus.h for session completion code. |

### 8.771.2 Field Documentation

#### 8.771.2.1 uint8_t unpack_omaDmNotificationsTlv_t::notification

#### 8.771.2.2 uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus

## 8.772 unpack_pds_GetPDSDefaults_t Struct Reference

### Data Fields

- uint32_t * pOperation
- uint8_t * pTimeout
- uint32_t * pInterval
- uint32_t * pAccuracy

- swi_uint256_t ParamPresenceMask

### 8.772.1 Detailed Description

Structure contain session configuration parameters.

**Parameters**

| | |
|---|---|
| *pOperation* | • Current session operating mode<br>– 0 - Standalone<br>– 1 - MS based<br>– 2 - MS assisted<br>• Bit to check in ParamPresenceMask - **1** |
| *pTimeout* | • Maximum amount of time (seconds) to work on each fix, maximum is 255<br>• Bit to check in ParamPresenceMask - **1** |
| *pInterval* | • Interval (seconds) between fix requests<br>• Bit to check in ParamPresenceMask - **1** |
| *pAccuracy* | • Preferred accuracy threshold (meters)<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.772.2 Field Documentation

#### 8.772.2.1 uint32_t∗ unpack_pds_GetPDSDefaults_t::pAccuracy

#### 8.772.2.2 swi_uint256_t unpack_pds_GetPDSDefaults_t::ParamPresenceMask

#### 8.772.2.3 uint32_t∗ unpack_pds_GetPDSDefaults_t::pInterval

#### 8.772.2.4 uint32_t∗ unpack_pds_GetPDSDefaults_t::pOperation

#### 8.772.2.5 uint8_t∗ unpack_pds_GetPDSDefaults_t::pTimeout

## 8.773 unpack_pds_GetPDSState_t Struct Reference

### Data Fields

- uint32_t ∗ pEnabledStatus
- uint32_t ∗ pTrackingStatus
- swi_uint256_t ParamPresenceMask

### 8.773.1 Detailed Description

Structure contain parameters of current PDS state.

**Parameters**

| | |
|---|---|
| *pEnabledStatus* | • Current PDS state<br>– 0 - disable<br>– 1 - enable<br>• Bit to check in ParamPresenceMask - **1** |
| *pTrackingStatus* | • Current PDS tracking session state<br>• Values:<br>– 0x00 - Unknown<br>– 0x01 - Inactive<br>– 0x02 - Active<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.773.2 Field Documentation

#### 8.773.2.1 swi_uint256_t unpack_pds_GetPDSState_t::ParamPresenceMask

#### 8.773.2.2 uint32_t* unpack_pds_GetPDSState_t::pEnabledStatus

#### 8.773.2.3 uint32_t* unpack_pds_GetPDSState_t::pTrackingStatus

## 8.774 unpack_pds_GetPortAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t * pbAuto
- swi_uint256_t ParamPresenceMask

### 8.774.1 Detailed Description

Structure contain the parameter of automatic tracking configuration for the NMEA COM port.

**Parameters**

| | |
|---|---|
| *pbAuto* | • Automatic tracking enabled for NMEA COM port<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>– Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.774.2 Field Documentation

#### 8.774.2.1 swi_uint256_t unpack_pds_GetPortAutomaticTracking_t::ParamPresenceMask

#### 8.774.2.2 uint32_t* unpack_pds_GetPortAutomaticTracking_t::pbAuto

## 8.775 unpack_pds_GetServiceAutomaticTracking_t Struct Reference

### Data Fields

- uint32_t * pbAuto
- swi_uint256_t ParamPresenceMask

### 8.775.1 Detailed Description

Structure contain the parameter for automatic tracking state.

**Parameters**

| | |
|---|---|
| *pbAuto* | • Automatic tracking session started for service<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.775.2 Field Documentation

#### 8.775.2.1 swi_uint256_t unpack_pds_GetServiceAutomaticTracking_t::ParamPresenceMask

#### 8.775.2.2 uint32_t* unpack_pds_GetServiceAutomaticTracking_t::pbAuto

## 8.776 unpack_pds_GetXTRAAutomaticDownload_t Struct Reference

### Data Fields

- uint32_t * pbEnabled
- uint16_t * pInterval
- swi_uint256_t ParamPresenceMask

### 8.776.1 Detailed Description

Structure contain the parameter of automatic tracking configuration for the NMEA COM port.

**Parameters**

| | |
|---|---|
| *pbEnabled* | • Automatic XTRA download status<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **16** |
| *pInterval* | • Interval (hours) between XTRA downloads<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.776.2 Field Documentation

#### 8.776.2.1 swi_uint256_t unpack_pds_GetXTRAAutomaticDownload_t::ParamPresenceMask

#### 8.776.2.2 uint32_t* unpack_pds_GetXTRAAutomaticDownload_t::pbEnabled

#### 8.776.2.3 uint16_t* unpack_pds_GetXTRAAutomaticDownload_t::pInterval

## 8.777 unpack_pds_GetXTRANetwork_t Struct Reference

### Data Fields

- uint32_t * pPreference
- swi_uint256_t ParamPresenceMask

### 8.777.1 Detailed Description

Structure contain the parameter for XTRA WWAN network preference

**Parameters**

| | |
|---|---|
| *pPreference* | • XTRA WWAN network preference<br>– 0x00 - None (any available network)<br>– 0x01 - Home-only, only when on home systems<br>– 0x02 - Roam-only, only when on non-home systems<br>• Bit to check in ParamPresenceMask - **18** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.777.2 Field Documentation

#### 8.777.2.1 swi_uint256_t unpack_pds_GetXTRANetwork_t::ParamPresenceMask

#### 8.777.2.2 uint32_t* unpack_pds_GetXTRANetwork_t::pPreference

## 8.778 unpack_pds_GetXTRAValidity_t Struct Reference

### Data Fields

- uint16_t * pGPSWeek
- uint16_t * pGPSWeekOffset
- uint16_t * pDuration
- swi_uint256_t ParamPresenceMask

### 8.778.1 Detailed Description

Structure contain the parameter for XTRA database validity period

**Parameters**

| | |
|---|---|
| *pGPSWeek* | • Starting GPS week of validity period<br>• Bit to check in ParamPresenceMask - **19** |
| *pGPSWeek-Offset* | • Starting GPS week offset (minutes) of validity period<br>• Bit to check in ParamPresenceMask - **19** |
| *pDuration* | • Length of validity period (hours)<br>• NULL pointer - Invalid data.<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.778.2 Field Documentation

#### 8.778.2.1 swi_uint256_t unpack_pds_GetXTRAValidity_t::ParamPresenceMask

#### 8.778.2.2 uint16_t* unpack_pds_GetXTRAValidity_t::pDuration

#### 8.778.2.3 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeek

#### 8.778.2.4 uint16_t* unpack_pds_GetXTRAValidity_t::pGPSWeekOffset

## 8.779 unpack_pds_SetEventReport_Ind_t Struct Reference

### Data Fields

- uint8_t has_PositionDataNMEA
- char PositionDataNMEA [200]
- uint8_t has_dLongitude
- double dLongitude
- uint8_t has_dLatitude

- double dLatitude
- uint8_t has_SessionStatus
- uint8_t SessionStatus
- uint8_t has_posSrc
- uint32_t posSrc
- swi_uint256_t ParamPresenceMask

### 8.779.1 Detailed Description

Structure contain the parameter for the Set event report Indication. Please check has_<Param_Name> field for presence of optional parameters

**Parameters**

| | |
|---|---|
| *PositionDataNMEA* | • Position Data NMEA.<br>• String containing the position data in NMEA sentence format (maximum 200 bytes).<br>• Bit to check in ParamPresenceMask - **16** |
| *dLongitude* | • Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of the Greenwich Meridian.<br>• Units - Decimal degrees<br>• Range - -180 to +180<br>• Note - Value is in double float format<br>• Bit to check in ParamPresenceMask - **19** |
| *dLatitude* | • Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator.<br>• Units: Decimal degrees<br>• Range: -90 to +90<br>• Note - Value is in double float format<br>• Bit to check in ParamPresenceMask - **19** |
| *session_status* | • Position Session Status<br>• Status of current session.<br>• Values<br>– 0x00 - Success<br>– 0x01 - In progress<br>– 0x02 - General failure<br>– 0x03 - Timeout<br>– 0x04 - User ended the session<br>– 0x05 - Bad parameter<br>– 0x06 - Phone is offline<br>– 0x07 - Engine is locked<br>– 0x08 - E911 session in progress<br>• Bit to check in ParamPresenceMask - **18**<br>• Note - This TLV is always sent while a fix is being generated or if it halts due to an error. The Parsed Position Data TLV (0x13) is included if the session status is Success (0x00) or In Progress (0x01); otherwise this TLV is passed alone. |

| | |
|---|---|
| *posSrc* | • Position Source<br>• This TLV is always sent with the Parsed Position Data TLV (0x13) when the latitude/longitude is marked as valid. Source of the position (bitmask).<br>• Values<br>– 0x00000001 - GPS<br>– 0x00000002 – Cell ID<br>– 0x00000004 – GLONASS<br>– 0x00000008 – Network<br>– 0x00000010 – External position injection<br>• Bit to check in ParamPresenceMask - **28** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.779.2 Field Documentation

#### 8.779.2.1 double unpack_pds_SetEventReport_Ind_t::dLatitude

#### 8.779.2.2 double unpack_pds_SetEventReport_Ind_t::dLongitude

#### 8.779.2.3 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLatitude

#### 8.779.2.4 uint8_t unpack_pds_SetEventReport_Ind_t::has_dLongitude

#### 8.779.2.5 uint8_t unpack_pds_SetEventReport_Ind_t::has_PositionDataNMEA

#### 8.779.2.6 uint8_t unpack_pds_SetEventReport_Ind_t::has_posSrc

#### 8.779.2.7 uint8_t unpack_pds_SetEventReport_Ind_t::has_SessionStatus

#### 8.779.2.8 swi_uint256_t unpack_pds_SetEventReport_Ind_t::ParamPresenceMask

#### 8.779.2.9 char unpack_pds_SetEventReport_Ind_t::PositionDataNMEA[200]

#### 8.779.2.10 uint32_t unpack_pds_SetEventReport_Ind_t::posSrc

#### 8.779.2.11 uint8_t unpack_pds_SetEventReport_Ind_t::SessionStatus

## 8.780 unpack_pds_SetPdsState_Ind_t Struct Reference

### Data Fields

- uint16_t TlvPresent
- uint32_t EnabledStatus
- uint32_t TrackingStatus
- swi_uint256_t ParamPresenceMask

### 8.780.1 Detailed Description

Structure contain the parameter for the Set PDS State Indication.

**Parameters**

| | |
|---|---|
| *EnabledStatus* | • GPS service state.<br>• Values<br>– 0x00 - Disable<br>– 0x01 - Enable<br>• Bit to check in ParamPresenceMask - **1** |
| *TrackingStatus* | • Tracking session state.<br>• Values<br>– 0x00 - Unknown<br>– 0x01 - Inactive<br>– 0x02 - Active<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.780.2 Field Documentation

#### 8.780.2.1 uint32_t unpack_pds_SetPdsState_Ind_t::EnabledStatus

#### 8.780.2.2 swi_uint256_t unpack_pds_SetPdsState_Ind_t::ParamPresenceMask

#### 8.780.2.3 uint16_t unpack_pds_SetPdsState_Ind_t::TlvPresent

#### 8.780.2.4 uint32_t unpack_pds_SetPdsState_Ind_t::TrackingStatus

## 8.781 unpack_pds_SLQSGetAGPSConfig_t Struct Reference

### Data Fields

- uint32_t * pServerAddress
- uint32_t * pServerPort
- uint8_t * pServerURL
- uint8_t * pServerURLLength
- swi_uint256_t ParamPresenceMask

### 8.781.1 Detailed Description

Structure contain the parameter for PDS AGPS (MS-based) configuration unpack.

**Parameters**

| | |
|---|---|
| *pServerAddress* | • IPv4 address of AGPS server. "0" if not set<br>• Bit to check in ParamPresenceMask - **16** |
| *pServerPort* | • Port number of AGPS server. "0" if not set<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *pServerURL* | • URL of the AGPS server. "0" if not set<br>• Bit to check in ParamPresenceMask - **17** |
| *pServerURL-Length* | • URL length of AGPS server. "0" if not set<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.781.2 Field Documentation

#### 8.781.2.1 swi_uint256_t unpack_pds_SLQSGetAGPSConfig_t::ParamPresenceMask

#### 8.781.2.2 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerAddress

#### 8.781.2.3 uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerPort

#### 8.781.2.4 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURL

#### 8.781.2.5 uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURLLength

## 8.782 unpack_pds_SLQSGetGPSStateInfo_t Struct Reference

### Data Fields

- uint8_t EngineState
- uint32_t ValidMask
- uint64_t Latitude
- uint64_t Longitude
- uint32_t HorizontalUncertainty
- uint32_t Altitude
- uint32_t VerticalUncertainty
- uint32_t TimeStmp_tow_ms
- uint16_t TimeStmp_gps_week
- uint32_t Time_uncert_ms
- uint8_t Iono_valid
- uint32_t gps_ephemeris_sv_msk
- uint32_t gps_almanac_sv_msk
- uint32_t gps_health_sv_msk
- uint32_t gps_visible_sv_msk
- uint32_t glo_ephemeris_sv_msk
- uint32_t glo_almanac_sv_msk
- uint32_t glo_health_sv_msk
- uint32_t glo_visible_sv_msk
- uint32_t sbas_ephemeris_sv_msk
- uint32_t sbas_almanac_sv_msk
- uint32_t sbas_health_sv_msk
- uint32_t sbas_visible_sv_msk
- uint16_t xtra_start_gps_week

- uint16_t xtra_start_gps_minutes
- uint16_t xtra_valid_duration_hours
- swi_uint256_t ParamPresenceMask

### 8.782.1 Detailed Description

Structure contain the parameter for GPS state Info.

**Parameters**

| | |
|---|---|
| *EngineState* | • Values:<br>– 0 - OFF<br>– 1 - ON<br>• This field is always valid<br>• Bit to check in ParamPresenceMask - **16** |
| *ValidMask* | • Mask of valid state information data.<br>• Values:<br>– 0x00000001 - Position(latitude/longitude/horizontal uncertainty)<br>– 0x00000002 - Altitude and vertical uncertainty<br>– 0x00000004 - Time ms<br>– 0x00000008 - Time week number<br>– 0x00000010 - Time uncertainty<br>– 0x00000020 - Iono validity<br>– 0x00000040 - GPS ephemeris<br>– 0x00000080 - GPS almanac<br>– 0x00000100 - GPS health<br>– 0x00000200 - GPS visible SVs<br>– 0x00000400 - GLONASS ephemeris<br>– 0x00000800 - GLONASS almanac<br>– 0x00001000 - GLONASS health<br>– 0x00002000 - GLONASS visible SVs<br>– 0x00004000 - SBAS ephemeris<br>– 0x00008000 - SBAS almanac<br>– 0x00010000 - SBAS health<br>– 0x00020000 - SBAS visible SVs<br>– 0x00040000 - XTRA information<br>• Bit to check in ParamPresenceMask - **16** |
| *Latitude* | • Latitude position referenced to the WGS-84 reference ellipsoid, counting positive angles north of the equator and negative angles south of the equator.<br>• Units: Decimal degrees<br>• Range: -90 to +90 degrees.<br>• Value is in double float format (refer to IEEE Std 754-1985)<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *Longitude* | • Longitude position referenced to the WGS-84 reference ellipsoid, counting positive angles east of the Greenwich Meridian and negative angles west of Greenwich meridian.<br>• Units: Decimal degrees<br>• Range: -180 to +180 degrees<br>• Value is in double float format (refer to IEEE Std 754-1985)<br>• Bit to check in ParamPresenceMask - **16** |
| *Horizontal-Uncertainty* | • Circular horizontal uncertainty (in meters). The uncertainty is provided at 63 percent confidence.<br>• Value is in single float format (refer to IEEE Std 754-1985)<br>• Bit to check in ParamPresenceMask - **16** |
| *Altitude* | • Height above the WGS-84 reference ellipsoid. Value conveys height (in meters) plus 500 m<br>• Range -500 to 15883<br>• Value in single float format (refer to IEEE Std 754-1985)<br>• Bit to check in ParamPresenceMask - **16** |
| *Vertical-Uncertainty* | • Vertical uncertainty (in meters). The uncertainty is provided at 68 percent confidence.<br>• Value in single float format (refer to IEEE Std 754-1985)<br>• Bit to check in ParamPresenceMask - **16** |
| *TimeStmp_tow_-ms* | • Time stamp in GPS time of week( in milliseconds)<br>• Bit to check in ParamPresenceMask - **16** |
| *TimeStmp_gps-_week* | • GPS week number<br>• Bit to check in ParamPresenceMask - **16** |
| *Time_uncert_ms* | • Time uncertainty (in milliseconds). The uncertainty is provided at 99 percent confidence.<br>• Bit to check in ParamPresenceMask - **16** |
| *Iono_valid* | • Iono validity.<br>• Values:<br>  – 0 - Invalid<br>  – 1 - Valid<br>• Bit to check in ParamPresenceMask - **16** |
| *gps_ephemeris-_sv_msk* | • GPS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *gps_almanac_-sv_msk* | • GPS SV mask for almanac; if the bit is set, almanac for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *gps_health_sv_-msk* | • GPS SV mask for health; if the bit is set, health for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *gps_visible_sv_-msk* | • GPS SV mask for visible Svs; if the bit is set, the SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *glo_ephemeris_-sv_msk* | • GLONASS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *glo_almanac_sv-_msk* | • GLONASS SV mask for almanac; if the bit is set, almanac for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *glo_health_sv_-msk* | • GLONASS SV mask for health; if the bit is set, health for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *glo_visible_sv_-msk* | • GLONASS SV mask for visible SVs; if the bit is set, the SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *sbas_-ephemeris_sv_-msk* | • SBAS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *sbas_almanac_-sv_msk* | • SBAS SV mask for almanac; if the bit is set, almanac for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *sbas_health_sv-_msk* | • SBAS SV mask for health; if the bit is set, health for that SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *sbas_visible_sv-_msk* | • SBAS SV mask for visible SVs; if the bit is set, the SV is available.<br>• Bit to check in ParamPresenceMask - **16** |
| *xtra_start_gps_-week* | • Current XTRA information is valid starting from this GPS week number<br>• Bit to check in ParamPresenceMask - **16** |
| *xtra_start_gps_-minutes* | • Current XTRA information is valid starting from the GPS minutes with the GPS week<br>• Bit to check in ParamPresenceMask - **16** |
| *xtra_valid_-duration_hours* | • XTRA information is valid for this many hours starting from the specified GPS week/minutes<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

## 8.782.2 Field Documentation

### 8.782.2.1 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Altitude

### 8.782.2.2 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::EngineState

### 8.782.2.3 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_almanac_sv_msk

### 8.782.2.4 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_ephemeris_sv_msk

### 8.782.2.5 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_health_sv_msk

### 8.782.2.6 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::glo_visible_sv_msk

### 8.782.2.7 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_almanac_sv_msk

### 8.782.2.8 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_ephemeris_sv_msk

### 8.782.2.9 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_health_sv_msk

### 8.782.2.10 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::gps_visible_sv_msk

### 8.782.2.11 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::HorizontalUncertainty

### 8.782.2.12 uint8_t unpack_pds_SLQSGetGPSStateInfo_t::Iono_valid

### 8.782.2.13 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Latitude

### 8.782.2.14 uint64_t unpack_pds_SLQSGetGPSStateInfo_t::Longitude

### 8.782.2.15 swi_uint256_t unpack_pds_SLQSGetGPSStateInfo_t::ParamPresenceMask

### 8.782.2.16 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_almanac_sv_msk

### 8.782.2.17 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_ephemeris_sv_msk

### 8.782.2.18 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_health_sv_msk

### 8.782.2.19 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::sbas_visible_sv_msk

### 8.782.2.20 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::Time_uncert_ms

### 8.782.2.21 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_gps_week

### 8.782.2.22 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::TimeStmp_tow_ms

### 8.782.2.23 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::ValidMask

### 8.782.2.24 uint32_t unpack_pds_SLQSGetGPSStateInfo_t::VerticalUncertainty

#### 8.782.2.25 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_minutes

#### 8.782.2.26 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_start_gps_week

#### 8.782.2.27 uint16_t unpack_pds_SLQSGetGPSStateInfo_t::xtra_valid_duration_hours

## 8.783 unpack_qmi_t Struct Reference

### Data Fields

- enum msgtype type
- uint16_t msgid
- uint16_t xid

### 8.783.1 Detailed Description

qmi response context

**Parameters**

| | | |
|---|---|---|
| out | *type* | message type |
| out | *msgid* | message id |
| out | *xid* | transaction id |

### 8.783.2 Field Documentation

#### 8.783.2.1 uint16_t unpack_qmi_t::msgid

#### 8.783.2.2 enum msgtype unpack_qmi_t::type

#### 8.783.2.3 uint16_t unpack_qmi_t::xid

## 8.784 unpack_qos_BindDataPort_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.784.1 Detailed Description

This structure contains unpack Qos BindDataPort.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.784.2 Field Documentation

#### 8.784.2.1 swi_uint256_t unpack_qos_BindDataPort_t::ParamPresenceMask

## 8.785 unpack_qos_dataRate_t Struct Reference

### Data Fields

- uint32_t dataRateMax
- uint32_t guaranteedRate

### 8.785.1 Detailed Description

This structure contains the IP flow data rate min max

**Parameters**

| | |
|---|---|
| *dataRateMax* | Maximum required data rate (bits per second) |
| *guaranteedRate* | Minimum guaranteed data rate (bits per second) |

### 8.785.2 Field Documentation

#### 8.785.2.1 uint32_t unpack_qos_dataRate_t::dataRateMax

#### 8.785.2.2 uint32_t unpack_qos_dataRate_t::guaranteedRate

## 8.786 unpack_qos_IPv4Addr_t Struct Reference

### Data Fields

- uint32_t addr
- uint32_t subnetMask

### 8.786.1 Detailed Description

This structure contains the IPv4 filter address

**Parameters**

| | |
|---|---|
| *addr* | IPv4 address |
| *subnetMask* | A packet matches if:<br>• (addr and subnetMask) == (IP pkt addr & subnetMask) Callers to set up a filter with a range of source addresses, if needed; subnet mask of all 1s (255.255.255.255) specifies a single address value |

### 8.786.2 Field Documentation

#### 8.786.2.1 uint32_t unpack_qos_IPv4Addr_t::addr

#### 8.786.2.2 uint32_t unpack_qos_IPv4Addr_t::subnetMask

## 8.787 unpack_qos_IPv6Addr_t Struct Reference

### Data Fields

- uint8_t addr [16]

- uint8_t prefixLen

### 8.787.1 Detailed Description

This structure contains the IPv6 filter address

**Parameters**

| | |
|---|---|
| *addr* | IPv6 address (in network byte order); this is a 16-byte byte array (in Big-endian format) |
| *prefixLen* | 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. |

### 8.787.2 Field Documentation

#### 8.787.2.1 uint8_t unpack_qos_IPv6Addr_t::addr[16]

#### 8.787.2.2 uint8_t unpack_qos_IPv6Addr_t::prefixLen

## 8.788 unpack_qos_IPv6TrafCls_t Struct Reference

### Data Fields

- uint8_t val
- uint8_t mask

### 8.788.1 Detailed Description

This structure contains the IPv6 filter traffic class

**Parameters**

| | |
|---|---|
| *val* | The traffic class value |
| *mask* | The packet matches the traffic class filter if: (IPv6_filter_traffic_class_val and IPv6_filter_traffic_class_mask) == (Traffic class value in the IP packet & IPv6_filter_traffic_class_mask) Example:<br>• IPv6_filter_tc_val = 00101000<br>• IPv6_filter_tc_mask = 11111100 Filter will compare only the first 6 bits in IPv6_filter_traffic_class with the first 6 bits in the traffic class field of the IP packet; first 6 bits in the traffic class field of the IP packet must be 001010 to match filter; last 2 bits can be anything, since they are ignored by filtering |

### 8.788.2 Field Documentation

#### 8.788.2.1 uint8_t unpack_qos_IPv6TrafCls_t::mask

#### 8.788.2.2 uint8_t unpack_qos_IPv6TrafCls_t::val

## 8.789 unpack_qos_pktErrRate_t Struct Reference

### Data Fields

- uint16_t multiplier

- uint16_t exponent

### 8.789.1 Detailed Description

This structure contains the IP flow packet error rate

**Parameters**

| | |
|---|---|
| *multiplier* | Factor m in calculating packet error rate: E = m∗10∗∗(-p) |
| *exponent* | Factor p in calculating packet error rate (see above) |

### 8.789.2 Field Documentation

#### 8.789.2.1 uint16_t unpack_qos_pktErrRate_t::exponent

#### 8.789.2.2 uint16_t unpack_qos_pktErrRate_t::multiplier

## 8.790 unpack_qos_Port_t Struct Reference

### Data Fields

- uint16_t port
- uint16_t range

### 8.790.1 Detailed Description

This structure contains the Port Filter

**Parameters**

| | |
|---|---|
| *port* | port value of the filter |
| *range* | 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 |

### 8.790.2 Field Documentation

#### 8.790.2.1 uint16_t unpack_qos_Port_t::port

#### 8.790.2.2 uint16_t unpack_qos_Port_t::range

## 8.791 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

### 8.791.1 Detailed Description

Structure with QoS flow details.

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

**Parameters**

| | |
|---|---|
| *QFlowState* | • QoS flow state information, please check unpack_qos_QosFlowInfoState_t for more information |
| *is_TxQFlowGranted_Available* | • TRUE if optional TxQFlowGranted is available |
| *TxQFlowGranted* | • The Tx Qos flow granted, please check unpack_qos_swiQosFlow_t for more information |
| *is_RxQFlowGranted_Available* | • TRUE if optional RxQFlowGranted is available |
| *RxQFlowGranted* | • The Rx Qos flow granted, please check unpack_qos_swiQosFlow_t for more information |
| *NumTxFilters* | • Number of Tx filters available |
| *TxQFilter* | • The Tx Qos filter, please check unpack_qos_swiQosFilter_t for more information<br>• See LITEQMI_MAX_QOS_FILTERS for more information |
| *NumRxFilters* | • Number of Tx filters available |
| *RxQFilter* | • The Rx Qos filter, please check unpack_qos_swiQosFilter_t for more information<br>• See LITEQMI_MAX_QOS_FILTERS for more information |
| *BearerID* | • The bearer ID<br>• Bearer ID or Radio Link Protocol (RLP) ID of the activated flow.<br>• Valid Values - 0 to 16<br>• 0xFF - Invalid value. |

### 8.791.2 Field Documentation

#### 8.791.2.1 uint8_t unpack_qos_QosFlowInfo_t::BearerID

#### 8.791.2.2 uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available

#### 8.791.2.3 uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available

#### 8.791.2.4 uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters

#### 8.791.2.5 uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters

#### 8.791.2.6 unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState

#### 8.791.2.7 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]

#### 8.791.2.8 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted

#### 8.791.2.9 unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]

#### 8.791.2.10 unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted

## 8.792 unpack_qos_QosFlowInfoState_t Struct Reference

### Data Fields

- uint32_t id
- uint8_t isNewFlow
- uint8_t state

### 8.792.1 Detailed Description

This structure contains QoS flow state

**Parameters**

| | |
|---|---|
| *id* | QoS identifier |
| *isNewFlow* | • 1 - Newly added flow<br>• 0 - Existing flow |
| *state* | This indicates that the flow that was added/modified/deleted:<br>• 0x01 - Flow activated<br>• 0x02 - Flow modified<br>• 0x03 - Flow deleted<br>• 0x04 - Flow suspended<br>• 0x05 - Flow enabled<br>• 0x06 - Flow disabled |

### 8.792.2 Field Documentation

#### 8.792.2.1 uint32_t unpack_qos_QosFlowInfoState_t::id

#### 8.792.2.2 uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow

#### 8.792.2.3 uint8_t unpack_qos_QosFlowInfoState_t::state

## 8.793 unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference

### Data Fields

- uint8_t NWQoSStatus
- swi_uint256_t ParamPresenceMask

### 8.793.1 Detailed Description

Structure that contains the response to get NW QoS status command

**Parameters**

| | |
|---|---|
| *NWQoSStatus* | Network QoS support status<br>• 0 - No QoS support in network<br>• 1 - Network supports QoS<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.793.2 Field Documentation

#### 8.793.2.1 uint8_t unpack_qos_SLQSQosGetNetworkStatus_t::NWQoSStatus

#### 8.793.2.2 swi_uint256_t unpack_qos_SLQSQosGetNetworkStatus_t::ParamPresenceMask

## 8.794 unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference

### Data Fields

- uint32_t apnId
- uint8_t ambr_ul
- uint8_t ambr_dl
- uint8_t ambr_ul_ext
- uint8_t ambr_dl_ext
- uint8_t ambr_ul_ext2
- uint8_t ambr_dl_ext2
- swi_uint256_t ParamPresenceMask

### 8.794.1 Detailed Description

Structure that contains extra APN parameters

**Parameters**

| | |
|---|---|
| *apnId* | • APN id<br>• ID identifing the APN that the client would like to query the AMBR params<br>• Bit to check in ParamPresenceMask - **3** |

| | |
|---|---|
| *ambr_ul* | • APN AMBR uplink<br>• APN AMBR uplink values from 1 kbps to 8640 kbps<br>• Bit to check in ParamPresenceMask - **3** |
| *ambr_dl* | • APN AMBR downlink<br>• APN AMBR downlink values from 1 kbps to 8640 kbps<br>• Bit to check in ParamPresenceMask - **3** |
| *ambr_ul_ext* | • Extended APN AMBR uplink<br>• APN AMBR uplink values from 8700 kbps to 256 Mbps<br>• Bit to check in ParamPresenceMask - **3** |
| *ambr_dl_ext* | • Extended APN AMBR downlink<br>• APN AMBR downlink values from 8700 kbps to 256 Mbps<br>• Bit to check in ParamPresenceMask - **3** |
| *ambr_ul_ext2* | • Second extended APN AMBR uplink<br>• APN AMBR uplink values from 256 Mbps to 65280 Mbps<br>• Bit to check in ParamPresenceMask - **3** |
| *ambr_dl_ext2* | • Second extended APN AMBR downlink<br>• APN AMBR downlink values from 256 Mbps to 65280 Mbps<br>• Bit to check in ParamPresenceMask - **3** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

## 8.794.2 Field Documentation

### 8.794.2.1 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl

### 8.794.2.2 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext

### 8.794.2.3 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_dl_ext2

### 8.794.2.4 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul

### 8.794.2.5 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext

### 8.794.2.6 uint8_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ambr_ul_ext2

### 8.794.2.7 uint32_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::apnId

### 8.794.2.8 swi_uint256_t unpack_qos_SLQSQosSwiReadApnExtraParams_t::ParamPresenceMask

## 8.795 unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference

### Data Fields

- uint32_t apnId
- uint32_t total_tx_pkt
- uint32_t total_tx_pkt_drp
- uint32_t total_rx_pkt
- uint64_t total_tx_bytes
- uint64_t total_tx_bytes_drp
- uint64_t total_rx_bytes
- uint32_t numQosFlow
- unpack_QosFlowStat_t qosFlow [10]
- swi_uint256_t ParamPresenceMask

### 8.795.1 Detailed Description

Structure that contains APN data statistics

**Parameters**

| | |
|---|---|
| *apnId* | • APN id<br>• ID identifing the connected APN that the client would like to query the data statistic for<br>• Bit to check in ParamPresenceMask - **3** |
| *total_tx_pkt* | • sum of all packets sent<br>• Bit to check in ParamPresenceMask - **3** |
| *total_tx_pkt_drp* | • sum of all(TX) packets dropped<br>• Bit to check in ParamPresenceMask - **3** |
| *total_rx_pkt* | • sum of all packets received<br>• Bit to check in ParamPresenceMask - **3** |
| *total_tx_bytes* | • sum of all bytes sent<br>• Bit to check in ParamPresenceMask - **3** |
| *total_tx_bytes_drp* | • sum of all(TX) bytes dropped<br>• Bit to check in ParamPresenceMask - **3** |
| *total_rx_bytes* | • number of received bytes for the QoS flow ID<br>• Bit to check in ParamPresenceMask - **3** |
| *numQosFlow* | • pointer to number of QoS flow Stat<br>• Bit to check in ParamPresenceMask - **4** |

| | |
|---|---|
| *qosFlow[LITEQMI_MAX_QOS_FLOW_PER_APN_STATS]* | • Data statistic per QoS flow<br>• See unpack_QosFlowStat_t for more information<br>• See LITEQMI_MAX_QOS_FLOW_PER_APN_STATS for more information<br>• Bit to check in ParamPresenceMask - **4** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.795.2 Field Documentation

#### 8.795.2.1 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId

#### 8.795.2.2 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow

#### 8.795.2.3 swi_uint256_t unpack_qos_SLQSQosSwiReadDataStats_t::ParamPresenceMask

#### 8.795.2.4 unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]

#### 8.795.2.5 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes

#### 8.795.2.6 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt

#### 8.795.2.7 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes

#### 8.795.2.8 uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp

#### 8.795.2.9 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt

#### 8.795.2.10 uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp

## 8.796 unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference

### Data Fields

- uint8_t NumFlows
- unpack_qos_QosFlowInfo_t QosFlowInfo [8]
- swi_uint256_t ParamPresenceMask

### 8.796.1 Detailed Description

Structure with QoS event details

**Parameters**

| | |
|---|---|
| *NumFlows* | • Number of QoS flows available<br>• Bit to check in ParamPresenceMask - **16** |

| | |
|---|---|
| *QosFlowInfo* | • The Qos flow details, please check unpack_qos_QosFlowInfo_t for more information<br>• See LITEQMI_MAX_QOS_FLOWS for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.796.2 Field Documentation

#### 8.796.2.1 uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows

#### 8.796.2.2 swi_uint256_t unpack_qos_SLQSSetQosEventCallback_ind_t::ParamPresenceMask

#### 8.796.2.3 unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]

## 8.797 unpack_qos_SLQSSetQosEventCallback_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.797.1 Detailed Description

This structure contains unpack QOS SLQSSetQosEventCallback. param ParamPresenceMask

- bitmask representation to indicate valid parameters.

### 8.797.2 Field Documentation

#### 8.797.2.1 swi_uint256_t unpack_qos_SLQSSetQosEventCallback_t::ParamPresenceMask

## 8.798 unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference

### Data Fields

- uint8_t status
- swi_uint256_t ParamPresenceMask

### 8.798.1 Detailed Description

Structure with network's QoS status

**Parameters**

| | |
|---|---|
| *status* | Network QoS support status<br>• 0x00 - Current network does not support QoS<br>• 0x01 - Current network supports QoS<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

**Note**

- Technology Supported: CDMA

### 8.798.2 Field Documentation

#### 8.798.2.1 swi_uint256_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::ParamPresenceMask

#### 8.798.2.2 uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status

## 8.799 unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference

### Data Fields

- uint16_t event
- swi_uint256_t ParamPresenceMask

### 8.799.1 Detailed Description

Structure with QoS primary flow events

**Parameters**

| | |
|---|---|
| *event* | Event which causes this indication:<br>• 0x0001 - Primary flow QoS modify operation success<br>• 0x0002 - Primary flow QoS modify operation failure<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.799.2 Field Documentation

#### 8.799.2.1 uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event

#### 8.799.2.2 swi_uint256_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::ParamPresenceMask

## 8.800 unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference

### Data Fields

- uint32_t id
- uint8_t status
- uint8_t event
- uint8_t reason
- swi_uint256_t ParamPresenceMask

### 8.800.1 Detailed Description

Structure with QoS status indication details

**Parameters**

| | |
|---|---|
| *id* | • Index identifying the QoS flow whose status is being reported<br>• Bit to check in ParamPresenceMask - **1** |
| *status* | Current QoS flow status:<br>• 0x01 - QMI_QOS_STATUS_ACTIVATED<br>• 0x02 - QMI_QOS_STATUS_SUSPENDED<br>• 0x03 - QMI_QOS_STATUS_GONE<br>• Bit to check in ParamPresenceMask - **1** |
| *event* | • 0x01 - QMI_QOS_ACTIVATED_EV<br>• 0x02 - QMI_QOS_SUSPENDED_EV<br>• 0x03 - QMI_QOS_GONE_EV<br>• 0x04 - QMI_QOS_MODIFY_ACCEPTED_EV<br>• 0x05 - QMI_QOS_MODIFY_REJECTED_EV<br>• 0x06 - QMI_QOS_INFO_CODE_UPDATED_EV<br>• Bit to check in ParamPresenceMask - **1** |
| *reason* | • 0x01 - QMI_QOS_INVALID_PARAMS<br>• 0x02 - QMI_QOS_INTERNAL_CALL_ENDED<br>• 0x03 - QMI_QOS_INTERNAL_ERROR<br>• 0x04 - QMI_QOS_INSUFFICIENT_LOCAL_Resources<br>• 0x05 - QMI_QOS_TIMED_OUT_OPERATION<br>• 0x06 - QMI_QOS_INTERNAL_UNKNOWN_CAUSE_CODE<br>• 0x07 - QMI_QOS_INTERNAL_MODIFY_IN_PROGRESS<br>• 0x08 - QMI_QOS_NOT_SUPPORTED<br>• 0x09 - QMI_QOS_NOT_AVAILABLE<br>• 0x0A - QMI_QOS_NOT_GUARANTEED<br>• 0x0B - QMI_QOS_INSUFFICIENT_NETWORK_RESOURCES<br>• 0x0C - QMI_QOS_AWARE_SYSTEM<br>• 0x0D - QMI_QOS_UNAWARE_SYSTEM<br>• 0x0E - QOS_REJECTED_OPERATION<br>• 0x0F - QMI_QOS_WILL_GRANT_WHEN_QOS_RESUMED<br>• 0x10 - QMI_QOS_NETWORK_CALL_ENDED<br>• 0x11 - QMI_QOS_NETWORK_SERVICE_NOT_AVAILABLE<br>• 0x12 - QMI_QOS_NETWORK_L2_LINK_RELEASED<br>• 0x13 - QMI_QOS_NETWORK_L2_LINK_REESTAB_REJ<br>• 0x14 - QMI_QOS_NETWORK_L2_LINK_REESTAB_IND<br>• 0x15 - QMI_QOS_NETWORK_UNKNOWN_CAUSE_CODE<br>• 0x16 - QMI_NETWORK_BUSY<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.800.2 Field Documentation

#### 8.800.2.1 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event

#### 8.800.2.2 uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id

#### 8.800.2.3 swi_uint256_t unpack_qos_SLQSSetQosStatusCallback_ind_t::ParamPresenceMask

#### 8.800.2.4 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason

#### 8.800.2.5 uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status

## 8.801 unpack_qos_swiQosFilter_t Struct Reference

### Data Fields

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

### 8.801.1 Detailed Description

This structure contains the QoS Filter Request.

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

**Parameters**

| | |
|---|---|
| *index* | Mandatory parameter IP filter index Integer that uniquely identifies each filter instance This TLV must be present in the request |
| *version* | 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<br>• 0x04 - IPv4<br>• 0x06 - Ipv6 |
| *IPv4SrcAddr* | IPv4 filter soruce address See unpack_qos_IPv4Addr_t for more information<br>• Implemented only for unsolicited indication |
| *IPv4DstAddr* | IPv4 filter destination address See unpack_qos_IPv4Addr_t for more information<br>• Implemented only for unsolicited indication |
| *NxtHdrProto* | 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:<br>• 0x01 = ICMP<br>• 0x06 = TCP<br>• 0x11 = UDP<br>• 0x32 = ESP Note: The next header protocol field will be set to 0xFD (TCP & UDP) if a TFT is received specifying a source or destination port number, but IP next header type is not specified. |
| *IPv4Tos* | IPv4 filter type of service See unpack_qos_Tos_t for more information |
| *IPv6SrcAddr* | IPv6 filter soruce address See unpack_qos_IPv6Addr_t for more information<br>• Implemented only for unsolicited indication |
| *IPv6DstAddr* | IPv6 filter destination address See unpack_qos_IPv6Addr_t for more information<br>• Implemented only for unsolicited indication |
| *IPv6TrafCls* | IPv6 filter traffic class See unpack_qos_IPv6TrafCls_t for more information |
| *IPv6Label* | IPv6 flow label Packet matches the IPv6 flow label filter if: ( ∗pIPv6Label == flow label in the IPv6 header)<br>• Implemented only for unsolicited indication |
| *TCPSrcPort* | TCP filter source port filter See unpack_qos_Port_t for more information<br>• Implemented only for unsolicited indication |
| *TCPDstPort* | TCP filter destination port filter See unpack_qos_Port_t for more information<br>• Implemented only for unsolicited indication |
| *UDPSrcPort* | UDP filter source port filter See unpack_qos_Port_t for more information<br>• Implemented only for unsolicited indication |
| *UDPDstPort* | UDP filter destination port filter See unpack_qos_Port_t for more information<br>• Implemented only for unsolicited indication |

| | |
|---|---|
| *EspSpi* | ESP filter security policy index Security policy index to uniquely identify each IP flow for filtering encrypted packets for encapsulating security payload<br>• Implemented only for unsolicited indication |
| *Precedence* | 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 |
| *Id* | 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 |
| *TranSrcPort* | Transport protocolfilter source port See unpack_qos_Port_t for more information<br>• Implemented only for unsolicited indication |
| *UDPDstPort* | Transport protocol filter destination port See unpack_qos_Port_t for more information<br>• Implemented only for unsolicited indication |

## 8.801.2 Field Documentation

### 8.801.2.1 uint32_t unpack_qos_swiQosFilter_t::EspSpi

### 8.801.2.2 uint16_t unpack_qos_swiQosFilter_t::Id

### 8.801.2.3 uint8_t unpack_qos_swiQosFilter_t::index

### 8.801.2.4 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr

### 8.801.2.5 unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr

### 8.801.2.6 unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos

### 8.801.2.7 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr

### 8.801.2.8 uint32_t unpack_qos_swiQosFilter_t::IPv6Label

### 8.801.2.9 unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr

### 8.801.2.10 unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls

### 8.801.2.11 uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available

### 8.801.2.12 uint8_t unpack_qos_swiQosFilter_t::is_Id_Available

### 8.801.2.13 uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available

### 8.801.2.14 uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available

### 8.801.2.15 uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available

### 8.801.2.16 uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available

### 8.801.2.17 uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available

### 8.801.2.18 uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available

#### 8.801.2.19 uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available

#### 8.801.2.20 uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available

#### 8.801.2.21 uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available

#### 8.801.2.22 uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available

#### 8.801.2.23 uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available

#### 8.801.2.24 uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available

#### 8.801.2.25 uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available

#### 8.801.2.26 uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available

#### 8.801.2.27 uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available

#### 8.801.2.28 uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto

#### 8.801.2.29 uint16_t unpack_qos_swiQosFilter_t::Precedence

#### 8.801.2.30 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort

#### 8.801.2.31 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort

#### 8.801.2.32 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort

#### 8.801.2.33 unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort

#### 8.801.2.34 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort

#### 8.801.2.35 unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort

#### 8.801.2.36 uint8_t unpack_qos_swiQosFilter_t::version

## 8.802 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

### 8.802.1 Detailed Description

This structure contains the QoS Flow Request.

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

**Parameters**

| | |
|---|---|
| *index* | • Mandatory parameter<br>• IP flow index<br>• Integer that uniquely identifies each flow instance<br>• Unique index must be assigned by the control point to every flow_spec instance |
| *ProfileId3GPP2* | • IP flow 3GPP2 profile ID<br>• 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 |
| *val_3GPP2Pri* | • IP flow 3GPP2 flow priority<br>• Flow priority used by the network in case of contention between flows with same QoS; this parameter applies for CDMA devices |
| *TrafficClass* | • IP flow traffic class<br>• Integer that designates the requested traffic class:<br>• 0 - Conversational<br>• 1 - Streaming<br>• 2 - Interactive<br>• 3 - Background |
| *DataRate* | • IP flow data rate min max<br>• See unpack_qos_dataRate_t for more information |

| | |
|---|---|
| *TokenBucket* | • IP flow data rate token bucket<br>• See unpack_qos_tokenBucket_t for more information |
| *Latency* | • IP flow latency<br>• Maximum delay (in milliseconds) that can be tolerated by an IP packet during transfer through the wireless link |
| *Jitter* | • IP flow jitter<br>• Difference between the maximum and minimum latency (in milliseconds) that can be tolerated by an IP packet during the transfer through the wireless link |
| *PktErrRate* | • IP flow packet error rate<br>• See unpack_qos_pktErrRate_t for more information |
| *MinPolicedPktSz* | • IP flow minimum policed packet size<br>• 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 |
| *MaxAllowedPktSz* | • IP flow maximum allowed packet size<br>• 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 |
| *val_3GPPResResidualBER* | • IP flow 3GPP residual bit error rate<br>• residual_bit_error_rate<br>• 0 = 5∗10-2 residual BER<br>• 1 = 1∗10-2 residual BER<br>• 2 = 5∗10-3 residual BER<br>• 3 = 4∗10-3 residual BER<br>• 4 = 1∗10-3 residual BER<br>• 5 = 1∗10-4 residual BER<br>• 6 = 1∗10-5 residual BER<br>• 7 = 1∗10-6 residual BER<br>• 8 = 6∗10-8 residual BER<br>• Integer that indicates the undetected BER for each IP flow in the delivered packets; Tapplies only to 3GPP networks |
| *val_3GPPTraHdlPri* | • 3GPP traffic handling priority<br>• 0 - Relative traffic handling priority 1<br>• 1 - Relative traffic handling priority 2<br>• 2 - Relative traffic handling priority 3<br>• Defines the relative priority of the flow; applies only to 3GPP networks |

| | |
|---|---|
| *val_3GPPImCn* | • IP flow 3GPP IM CN flag<br>• IM CN subsystem signaling flag:<br>• 0x00 - FALSE<br>• 0x01 - TRUE<br>• This parameter applies only to 3GPP networks |
| *val_3GPPSigInd* | • IP flow 3GPP signaling indication<br>• 0x00 - FALSE<br>• 0x01 - TRUE<br>• This parameter applies only to 3GPP networks |
| *LteQci* | • LTE QoS Class Identifier<br>• QoS Class Identifier(QCI) is a required parameter to request QoS in LTE<br>• QCI values:<br>– QCI value 0 requests the network to assign the appropriate QCI value<br>– QCI values 1-4 are associated with guaranteed bitrates<br>– QCI values 5-9 are associated with nonguaranteed bitrates, so the values specified as guaranteed and maximum bitrates are ignored |

## 8.802.2 Field Documentation

### 8.802.2.1 unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate

### 8.802.2.2 uint8_t unpack_qos_swiQosFlow_t::index

### 8.802.2.3 uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available

### 8.802.2.4 uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available

### 8.802.2.5 uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available

### 8.802.2.6 uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available

### 8.802.2.7 uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available

### 8.802.2.8 uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available

### 8.802.2.9 uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available

### 8.802.2.10 uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available

### 8.802.2.11 uint8_t unpack_qos_swiQosFlow_t::is_TokenBucket_Available

### 8.802.2.12 uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available

### 8.802.2.13 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available

### 8.802.2.14 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available

#### 8.802.2.15 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available

#### 8.802.2.16 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available

#### 8.802.2.17 uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available

#### 8.802.2.18 uint32_t unpack_qos_swiQosFlow_t::Jitter

#### 8.802.2.19 uint32_t unpack_qos_swiQosFlow_t::Latency

#### 8.802.2.20 uint8_t unpack_qos_swiQosFlow_t::LteQci

#### 8.802.2.21 uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz

#### 8.802.2.22 uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz

#### 8.802.2.23 unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate

#### 8.802.2.24 uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2

#### 8.802.2.25 unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket

#### 8.802.2.26 uint8_t unpack_qos_swiQosFlow_t::TrafficClass

#### 8.802.2.27 uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri

#### 8.802.2.28 uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn

#### 8.802.2.29 uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER

#### 8.802.2.30 uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd

#### 8.802.2.31 uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri

## 8.803 unpack_qos_tokenBucket_t Struct Reference

### Data Fields

- uint32_t peakRate
- uint32_t tokenRate
- uint32_t bucketSz

### 8.803.1 Detailed Description

This structure contains the TP flow data rate token bucket

**Parameters**

| | |
|---|---|
| *peakRate* | Maximum rate at which data can be transmitted when the token bucket is full (bits per second) |
| *tokenRate* | 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 |
| *bucketSz* | 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 |

### 8.803.2 Field Documentation

#### 8.803.2.1 uint32_t unpack_qos_tokenBucket_t::bucketSz

#### 8.803.2.2 uint32_t unpack_qos_tokenBucket_t::peakRate

#### 8.803.2.3 uint32_t unpack_qos_tokenBucket_t::tokenRate

## 8.804 unpack_qos_Tos_t Struct Reference

### Data Fields

- uint8_t val
- uint8_t mask

### 8.804.1 Detailed Description

This structure contains the IPv4 filter type of service

**Parameters**

| | |
|---|---|
| *val* | Type of service value |
| *mask* | Packet matches the TOS filter if: (IPv4_filter_tos_val and IPv4_filter_tos_mask) == (TOS value in the IP packet & IPv4_filter_tos_mask) Example:<br>• IPv4_filter_tos_val = 00101000<br>• IPv4_filter_tos_mask = 11111100 The filter will compare only the first 6 bits in the IPv4_filter_-type_of_service with the first 6 bits in the TOS field of the IP packet. The first 6 bits in the TOS field of the IP packet must be 001010 to match the filter. The last 2 bits can be anything since they are ignored by filtering. |

### 8.804.2 Field Documentation

#### 8.804.2.1 uint8_t unpack_qos_Tos_t::mask

#### 8.804.2.2 uint8_t unpack_qos_Tos_t::val

## 8.805 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

### 8.805.1 Detailed Description

This structure contains the Data statistic per QoS flow

**Parameters**

| | |
|---|---|
| *bearerId* | • Bearer ID |
| *tx_pkt* | • number of sent packets for the QoS flow ID |
| *tx_pkt_drp* | • number of dropped(TX) packets for the QoS flow ID |
| *tx_bytes* | • number of sent bytes for the QoS flow ID |
| *tx_bytes_drp* | • number of dropped(TX) bytes for the QoS flow ID |

### 8.805.2 Field Documentation

#### 8.805.2.1 uint32_t unpack_QosFlowStat_t::bearerId

#### 8.805.2.2 uint64_t unpack_QosFlowStat_t::tx_bytes

#### 8.805.2.3 uint64_t unpack_QosFlowStat_t::tx_bytes_drp

#### 8.805.2.4 uint32_t unpack_QosFlowStat_t::tx_pkt

#### 8.805.2.5 uint32_t unpack_QosFlowStat_t::tx_pkt_drp

## 8.806 unpack_result_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.806.1 Detailed Description

unpack result struct

**Parameters**

| | | |
|---|---|---|
| `out` | *Tlvresult* | |
| `out` | *ParamPresence-Mask* | Parameter presence bit mask |

### 8.806.2 Field Documentation

#### 8.806.2.1 swi_uint256_t unpack_result_t::ParamPresenceMask

#### 8.806.2.2 uint16_t unpack_result_t::Tlvresult

## 8.807 unpack_rms_GetSMSWake_t Struct Reference

### Data Fields

- uint32_t enabled
- uint32_t wake_mask
- swi_uint256_t ParamPresenceMask

### 8.807.1 Detailed Description

Unpack get SMS wake parameters

**Parameters**

| | |
|---|---|
| *enabled* | • SMS wake functionality enabled<br>– 0 - Disabled<br>– 1 - Enabled<br>• Bit to check in ParamPresenceMask - **16** |
| *wake_mask* | • SMS wake mask to search for incoming messages (only relevant when enabled)<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.807.2 Field Documentation

#### 8.807.2.1 uint32_t unpack_rms_GetSMSWake_t::enabled

#### 8.807.2.2 swi_uint256_t unpack_rms_GetSMSWake_t::ParamPresenceMask

#### 8.807.2.3 uint32_t unpack_rms_GetSMSWake_t::wake_mask

## 8.808 unpack_rms_SetSMSWake_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.808.1 Detailed Description

This structure contains unpack Set SMS Wake.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.808.2 Field Documentation

#### 8.808.2.1 swi_uint256_t unpack_rms_SetSMSWake_t::ParamPresenceMask

## 8.809 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

### 8.809.1 Detailed Description

WDS packet RM Transfer Statistics data structure for individual session

**Parameters**

| | |
|---|---|
| *TxOkConutTlv* | • Tx Ok Packet Tlv Value. |
| *RxOkConutTlv* | • Rx Ok Packet Tlv Value. |
| *TxOkByteCount-Tlv* | • Tx Ok Byte Count Packet Tlv Value. |
| *RxOkByteCount-Tlv* | • Rx Ok Byte Count Packet Tlv Value. |
| *TxDropConutTlv* | • Tx Drop Count Packet Tlv Value. |
| *RxDropConutTlv* | • Rx Drop Count Packet Tlv Value. |

### 8.809.2 Field Documentation

#### 8.809.2.1 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv

#### 8.809.2.2 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::RxOkByteCountTlv

#### 8.809.2.3 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxOkConutTlv

#### 8.809.2.4 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxDropConutTlv

#### 8.809.2.5 wds_DataULongLongTlv unpack_RMTransferStatistics_ind_t::TxOkByteCountTlv

#### 8.809.2.6 wds_DataULongTlv unpack_RMTransferStatistics_ind_t::TxOkConutTlv

## 8.810 unpack_sar_SLQSGetRfSarState_t Struct Reference

### Data Fields

- uint32_t * pSarRFState
- swi_uint256_t ParamPresenceMask

### 8.810.1 Detailed Description

This structure contains unpack get RF SAR state parameter.

**Parameters**

| | |
|---|---|
| *pSarRFState* | • SAR RF State<br>– QMI_SAR_RF_STATE_DEFAULT<br>– QMI_SAR_RF_STATE_1<br>– QMI_SAR_RF_STATE_2<br>– QMI_SAR_RF_STATE_3<br>– QMI_SAR_RF_STATE_4<br>– QMI_SAR_RF_STATE_5<br>– QMI_SAR_RF_STATE_6<br>– QMI_SAR_RF_STATE_7<br>– QMI_SAR_RF_STATE_8<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.810.2 Field Documentation

#### 8.810.2.1 swi_uint256_t unpack_sar_SLQSGetRfSarState_t::ParamPresenceMask

#### 8.810.2.2 uint32_t* unpack_sar_SLQSGetRfSarState_t::pSarRFState

## 8.811 unpack_sms_GetSMSCAddress_t Struct Reference

### Data Fields

- uint8_t addressSize
- uint8_t * pSMSCAddress
- uint8_t typeSize
- uint8_t * pSMSCType
- swi_uint256_t ParamPresenceMask

### 8.811.1 Detailed Description

Structure contains Gets the SMS center address parameters.

**Parameters**

| | |
|---|---|
| *addressSize* | • The maximum number of characters (including NULL terminator) that the SMS center address array can contain.<br>• Bit to check in ParamPresenceMask - **1** |
| *pSMSCAddress* | • The SMS center address represented as a NULL terminated string.<br>• Bit to check in ParamPresenceMask - **1** |
| *typeSize* | • The maximum number of characters (including NULL terminator) that the SMS center address type array can contain.<br>• Bit to check in ParamPresenceMask - **1** |
| *pSMSCType* | • The SMS center address type represented as a NULL terminated string.<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.811.2 Field Documentation

#### 8.811.2.1 uint8_t unpack_sms_GetSMSCAddress_t::addressSize

#### 8.811.2.2 swi_uint256_t unpack_sms_GetSMSCAddress_t::ParamPresenceMask

#### 8.811.2.3 uint8_t∗ unpack_sms_GetSMSCAddress_t::pSMSCAddress

#### 8.811.2.4 uint8_t∗ unpack_sms_GetSMSCAddress_t::pSMSCType

#### 8.811.2.5 uint8_t unpack_sms_GetSMSCAddress_t::typeSize

## 8.812 unpack_sms_SaveSMS_t Struct Reference

### Data Fields

- uint32_t ∗ pMessageIndex
- swi_uint256_t ParamPresenceMask

### 8.812.1 Detailed Description

This structure contains unpack save SMS parameter.

**Parameters**

| | |
|---|---|
| *pMessageIndex* | • The message index assigned by the device<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.812.2 Field Documentation

#### 8.812.2.1 swi_uint256_t unpack_sms_SaveSMS_t::ParamPresenceMask

#### 8.812.2.2 uint32_t* unpack_sms_SaveSMS_t::pMessageIndex

## 8.813 unpack_sms_SendSMS_t Struct Reference

### Data Fields

- uint16_t messageID
- uint32_t messageFailureCode
- swi_uint256_t ParamPresenceMask

### 8.813.1 Detailed Description

This structure contains unpack send SMS parameters.

**Parameters**

| | |
|---|---|
| *messageID* | • WMS message ID<br>• Bit to check in ParamPresenceMask - **1** |
| *messageFailure-Code* | • pointer to message failure code.<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.813.2 Field Documentation

#### 8.813.2.1 uint32_t unpack_sms_SendSMS_t::messageFailureCode

#### 8.813.2.2 uint16_t unpack_sms_SendSMS_t::messageID

#### 8.813.2.3 swi_uint256_t unpack_sms_SendSMS_t::ParamPresenceMask

## 8.814 unpack_sms_SetNewSMSCallback_ind_t Struct Reference

### Data Fields

- newMTMessageTlv NewMMTlv
- transferRouteMessageTlv TRMessageTlv
- messageModeTlv MMTlv

- sMSEtwsMessageTlv ETWSTlv
- eTWSPLMNInfoTlv ETWSPLMNTlv
- sMSCAddressTlv SMSCTlv
- sMSOnIMSTlv IMSTlv
- swi_uint256_t ParamPresenceMask

### 8.814.1 Detailed Description

This structure contains unpack new SMS callback indication.

**Parameters**

| | |
|---|---|
| *NewMMTlv* | • MT message<br>• Bit to check in ParamPresenceMask - **16** |
| *TRMessageTlv* | • Transfer Route MT Message<br>• See transferRouteMessageTlv for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *MMTlv* | • Message mode<br>• See messageModeTlv for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *ETWSTlv* | • ETWS Message<br>• See sMSEtwsMessageTlv for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *ETWSPLMNTlv* | • ETWS PLMN Information<br>• See eTWSPLMNInfoTlv for more information<br>• Bit to check in ParamPresenceMask - **20** |
| *SMSCTlv* | • SMSC Address<br>• See sMSCAddressTlv for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *IMSTlv* | • SMS on IMS<br>• See sMSOnIMSTlv for more information<br>• Bit to check in ParamPresenceMask - **22** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.814.2 Field Documentation

#### 8.814.2.1 eTWSPLMNInfoTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv

#### 8.814.2.2 sMSEtwsMessageTlv unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv

#### 8.814.2.3 sMSOnIMSTlv unpack_sms_SetNewSMSCallback_ind_t::IMSTlv

#### 8.814.2.4 messageModeTlv unpack_sms_SetNewSMSCallback_ind_t::MMTlv

#### 8.814.2.5 newMTMessageTlv unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv

#### 8.814.2.6 swi_uint256_t unpack_sms_SetNewSMSCallback_ind_t::ParamPresenceMask

#### 8.814.2.7 sMSCAddressTlv unpack_sms_SetNewSMSCallback_ind_t::SMSCTlv

#### 8.814.2.8 transferRouteMessageTlv unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv

## 8.815 unpack_sms_SetNewSMSCallback_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.815.1 Detailed Description

This structure contains unpack set new SMS callback.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.815.2 Field Documentation

#### 8.815.2.1 swi_uint256_t unpack_sms_SetNewSMSCallback_t::ParamPresenceMask

## 8.816 unpack_sms_SetSMSCAddress_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.816.1 Detailed Description

This structure contains unpack Set SMS Address parameter.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.816.2 Field Documentation

#### 8.816.2.1 swi_uint256_t unpack_sms_SetSMSCAddress_t::ParamPresenceMask

## 8.817 unpack_sms_SLQSDeleteSMS_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.817.1 Detailed Description

This structure contains unpack delete SMS parameters.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.817.2 Field Documentation

#### 8.817.2.1 swi_uint256_t unpack_sms_SLQSDeleteSMS_t::ParamPresenceMask

## 8.818 unpack_sms_SLQSGetIndicationRegister_t Struct Reference

### Data Fields

- sms_getIndicationReg * pGetIndicationRegInfo
- swi_uint256_t ParamPresenceMask

### 8.818.1 Detailed Description

Structure containing Parameters that provides registration state of different WMS indications.

**Parameters**

| | |
|---|---|
| *pGetIndication-RegInfo* | • Pointer to structure of getIndicationReg<br>– See sms_getIndicationReg for more information<br>– pRegTransLayerInfoEvt<br>* Bit to check in ParamPresenceMask - **16**<br>– pRegTransNWRegInfoEvt<br>* Bit to check in ParamPresenceMask - **17**<br>– pRegCallStatInfoEvt<br>* Bit to check in ParamPresenceMask - **18** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.818.2 Field Documentation

#### 8.818.2.1 swi_uint256_t unpack_sms_SLQSGetIndicationRegister_t::ParamPresenceMask

#### 8.818.2.2 sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo

## 8.819 unpack_sms_SLQSGetMessageWaiting_t Struct Reference

### Data Fields

- sms_getMsgWaitingInfo * pGetMsgWaitingInfoResp
- swi_uint256_t ParamPresenceMask

### 8.819.1 Detailed Description

Structure contain Parameter that provide information about the message waiting information.

**Parameters**

| | |
|---|---|
| *pGetMsg-WaitingInfoResp* | • Pointer to structure of getMsgWaitingInfoResp<br>– See sms_getMsgWaitingInfo for more information<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.819.2 Field Documentation

#### 8.819.2.1 swi_uint256_t unpack_sms_SLQSGetMessageWaiting_t::ParamPresenceMask

#### 8.819.2.2 sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp

## 8.820 unpack_sms_SLQSGetSMS_t Struct Reference

### Data Fields

- uint32_t messageTag
- uint32_t messageFormat
- uint32_t messageSize
- uint8_t message [2048]
- swi_uint256_t ParamPresenceMask

### 8.820.1 Detailed Description

This sturcture contains unpack get SMS parameters.

**Parameters**

| | |
|---|---|
| *messageTag* | • Message tag<br>– 0 - Read<br>– 1 - Not read<br>– 2 - Mobile originated and sent<br>– 3 - Mobile originated but not yet sent |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *messageFormat* | • Message format<br>– 0 - CDMA (IS-637B)<br>– 1 - 5 (Reserved)<br>– 6 - GSM/WCDMA PP |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *messageSize* | • Upon input the maximum number of bytes that can be written to the message array.<br>• Upon successful output the actual number of bytes written to the message array.<br>• Bit to check in ParamPresenceMask - **1** |
| *message* | • The message contents array |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.820.2 Field Documentation

#### 8.820.2.1 uint8_t unpack_sms_SLQSGetSMS_t::message[2048]

#### 8.820.2.2 uint32_t unpack_sms_SLQSGetSMS_t::messageFormat

#### 8.820.2.3 uint32_t unpack_sms_SLQSGetSMS_t::messageSize

#### 8.820.2.4 uint32_t unpack_sms_SLQSGetSMS_t::messageTag

#### 8.820.2.5 swi_uint256_t unpack_sms_SLQSGetSMS_t::ParamPresenceMask

## 8.821 unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference

### Data Fields

- sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig
- sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig
- swi_uint256_t ParamPresenceMask

### 8.821.1 Detailed Description

Structure contain parameters that Provides Information about the SMS BroadcastConfiguration

**Parameters**

| | |
|---|---|
| *pBroadcast-Config* | • The data for 3GPP Broadcast Information(Optional).<br>• Bit to check in ParamPresenceMask - **16** |
| *pCDMA-BroadcastConfig* | • The data for 3GPP2 Broadcast Information(Optional).<br>• Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.821.2 Field Documentation

#### 8.821.2.1 swi_uint256_t unpack_sms_SLQSGetSmsBroadcastConfig_t::ParamPresenceMask

#### 8.821.2.2 sms_qaQmi3GPPBroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pBroadcastConfig

#### 8.821.2.3 sms_qaQmi3GPP2BroadcastCfgInfo* unpack_sms_SLQSGetSmsBroadcastConfig_t::pCDMABroadcastConfig

## 8.822 unpack_sms_SLQSGetSMSList_t Struct Reference

### Data Fields

- uint32_t messageListSize
- qmiSmsMessageList messageList [255]
- swi_uint256_t ParamPresenceMask

### 8.822.1 Detailed Description

This structure contains unpack get SMS list parameters.

**Parameters**

| | |
|---|---|
| *messageListSize* | • Upon input the maximum number of elements that the message list array can contain.<br>• Upon successful output the actual number of elements in the message list array.<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *messageList* | • Message List<br>• See qmiSmsMessageList for more information<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.822.2 Field Documentation

#### 8.822.2.1 qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]

#### 8.822.2.2 uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize

#### 8.822.2.3 swi_uint256_t unpack_sms_SLQSGetSMSList_t::ParamPresenceMask

## 8.823 unpack_sms_SLQSGetTransLayerInfo_t Struct Reference

### Data Fields

- sms_getTransLayerInfo * pGetTransLayerInfo
- swi_uint256_t ParamPresenceMask

### 8.823.1 Detailed Description

Structure contain parameters that gives information about the transport layer.

**Parameters**

| | |
|---|---|
| *pGetTransLayer-Info* | • Pointer to structure of getTransLayerInfo<br>– See sms_getTransLayerInfo for more information<br>– pGetTransLayerInfo->pRegInd<br>* Bit to check in ParamPresenceMask - **16**<br>– pGetTransLayerInfo->pTransLayerInfo<br>* Bit to check in ParamPresenceMask - **17** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.823.2 Field Documentation

#### 8.823.2.1 swi_uint256_t unpack_sms_SLQSGetTransLayerInfo_t::ParamPresenceMask

#### 8.823.2.2 sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo

## 8.824 unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference

### Data Fields

- sms_getTransNWRegInfo * pGetTransNWRegInfo
- swi_uint256_t ParamPresenceMask

### 8.824.1 Detailed Description

Structure containing parameters that provides transport layer network registration info.

**Parameters**

| | |
|---|---|
| *pGetTransNW-RegInfo* | • Pointer to structure of getTransNWRegInfo<br>– See sms_getTransNWRegInfo for more information<br>– Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.824.2 Field Documentation

#### 8.824.2.1 swi_uint256_t unpack_sms_SLQSGetTransNWRegInfo_t::ParamPresenceMask

#### 8.824.2.2 sms_getTransNWRegInfo* unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo

## 8.825 unpack_sms_SLQSModifySMSStatus_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.825.1 Detailed Description

This structure contains unpack modify SMS status parameters.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.825.2 Field Documentation

#### 8.825.2.1 swi_uint256_t unpack_sms_SLQSModifySMSStatus_t::ParamPresenceMask

## 8.826 unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference

### Data Fields

- uint8_t NWRegStat
- swi_uint256_t ParamPresenceMask

### 8.826.1 Detailed Description

Contains the parameters passed for indication about transport network registration change by the device.

**Parameters**

| | |
|---|---|
| *NWRegStat* | • provides the transport network registration information<br>• Values:<br>– 0x00 - No Service<br>– 0x01 - In Progress<br>– 0x02 - Failed<br>– 0x03 - Limited Service<br>– 0x04 - Full Service<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

**Note**

None

### 8.826.2 Field Documentation

#### 8.826.2.1 uint8_t unpack_sms_SLQSNWRegInfoCallback_ind_t::NWRegStat

#### 8.826.2.2 swi_uint256_t unpack_sms_SLQSNWRegInfoCallback_ind_t::ParamPresenceMask

## 8.827 unpack_sms_SLQSSendAsyncSMS_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.827.1 Detailed Description

This structure contains unpack SLQSSendAsyncSMS parameter.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.827.2 Field Documentation

#### 8.827.2.1 swi_uint256_t unpack_sms_SLQSSendAsyncSMS_t::ParamPresenceMask

## 8.828 unpack_sms_SLQSSetIndicationRegister_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.828.1 Detailed Description

This structure contains unpack SLQSSetIndicationRegister parameter.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.828.2 Field Documentation

#### 8.828.2.1 swi_uint256_t unpack_sms_SLQSSetIndicationRegister_t::ParamPresenceMask

## 8.829 unpack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.829.1 Detailed Description

This structure contains unpack SLQSSetSmsBroadcastActivation parameter.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.829.2 Field Documentation

#### 8.829.2.1 swi_uint256_t unpack_sms_SLQSSetSmsBroadcastActivation_t::ParamPresenceMask

## 8.830 unpack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.830.1 Detailed Description

This structure contains unpack SLQSSetSmsBroadcastConfig parameters.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.830.2 Field Documentation

#### 8.830.2.1 swi_uint256_t unpack_sms_SLQSSetSmsBroadcastConfig_t::ParamPresenceMask

## 8.831 unpack_sms_SLQSSetSmsStorage_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.831.1 Detailed Description

This structure contains unpack SLQSSetSmsStorage parameter.

**Parameters**

| | |
|---|---|
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.831.2 Field Documentation

#### 8.831.2.1 swi_uint256_t unpack_sms_SLQSSetSmsStorage_t::ParamPresenceMask

## 8.832 unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference

### Data Fields

- sms_maxStorageSizeResp * pMaxStorageSizeResp
- swi_uint256_t ParamPresenceMask

### 8.832.1 Detailed Description

Structure contain parameters that provides the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available

**Parameters**

| | |
|---|---|
| *pMaxStorage-SizeResp* | • Response parameters for SmsSLQSGetMaxStorageSize<br>  – See sms_maxStorageSizeResp for more information<br>• pMaxStorageSizeResp->maxStorageSize<br>  – Bit to check in ParamPresenceMask - **1**<br>• pMaxStorageSizeResp->freeSlots<br>  – Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.832.2 Field Documentation

#### 8.832.2.1 swi_uint256_t unpack_sms_SLQSSmsGetMaxStorageSize_t::ParamPresenceMask

#### 8.832.2.2 sms_maxStorageSizeResp* unpack_sms_SLQSSmsGetMaxStorageSize_t::pMaxStorageSizeResp

## 8.833 unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference

### Data Fields

- sms_msgProtocolResp * pMessageProtocol
- swi_uint256_t ParamPresenceMask

### 8.833.1 Detailed Description

Structure contain Parameters that get the message protocol currently in use for the WMS client.

**Parameters**

| | |
|---|---|
| *pMessageProtocol* | • Pointer to smsMsgprotocolResp<br>  – See sms_msgProtocolResp for more information<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.833.2 Field Documentation

#### 8.833.2.1 swi_uint256_t unpack_sms_SLQSSmsGetMessageProtocol_t::ParamPresenceMask

#### 8.833.2.2 sms_msgProtocolResp* unpack_sms_SLQSSmsGetMessageProtocol_t::pMessageProtocol

## 8.834 unpack_sms_SLQSSmsSetRoutes_t Struct Reference

### Data Fields

- swi_uint256_t ParamPresenceMask

### 8.834.1 Detailed Description

This structure contains unpack SLQSSmsSetRoutes parameter.

**Parameters**

| | |
|---|---|
| *ParamPresenceMask* | • bitmask representation to indicate valid parameters. |

### 8.834.2 Field Documentation

#### 8.834.2.1 swi_uint256_t unpack_sms_SLQSSmsSetRoutes_t::ParamPresenceMask

## 8.835 unpack_sms_SLQSSwiGetSMSStorage_t Struct Reference

### Data Fields

- uint32_t * pSmsStorage
- swi_uint256_t ParamPresenceMask

### 8.835.1 Detailed Description

Structure contain Parameters that return current SMS configuration that is applied to all incoming and outgoing messages.

**Parameters**

| | |
|---|---|
| *pSmsStorage* | • Values:<br>– 0x01 - device's permanent memory<br>– 0x02 - UICC<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.835.2 Field Documentation

#### 8.835.2.1 swi_uint256_t unpack_sms_SLQSSwiGetSMSStorage_t::ParamPresenceMask

#### 8.835.2.2 uint32_t* unpack_sms_SLQSSwiGetSMSStorage_t::pSmsStorage

## 8.836 unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference

### Data Fields

- uint8_t regInd
- sms_transLayerInfo * pTransLayerInfo
- swi_uint256_t ParamPresenceMask

### 8.836.1 Detailed Description

Contains the parameters passed for the indication about change in transport layer information

**Parameters**

| | |
|---|---|
| *regInd* | • Indicates whether the transport layer is registered or not<br>• Values:<br>– 0x00 - Transport layer is not registered<br>– 0x01 - Transport layer is registered<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *pTransLayerInfo* | • Optional parameter<br>• See sms_transLayerInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

**Note**

None

### 8.836.2 Field Documentation

#### 8.836.2.1 swi_uint256_t unpack_sms_SLQSTransLayerInfoCallback_ind_t::ParamPresenceMask

#### 8.836.2.2 sms_transLayerInfo∗ unpack_sms_SLQSTransLayerInfoCallback_ind_t::pTransLayerInfo

#### 8.836.2.3 uint8_t unpack_sms_SLQSTransLayerInfoCallback_ind_t::regInd

## 8.837 unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t Struct Reference

### Data Fields

- uint16_t sendStatus
- uint16_t messageID
- uint16_t causeCode
- uint8_t errorClass
- uint16_t RPCause
- uint8_t TPCause
- uint8_t msgDelFailureType
- uint8_t msgDelFailureCause
- uint8_t alphaIDLen
- uint8_t ∗ pAlphaID
- uint32_t userData
- swi_uint256_t ParamPresenceMask

### 8.837.1 Detailed Description

This structure contains SMS parameters for indication of RAW ASYNC SEND

**Parameters**

| | |
|---|---|
| *sendStatus* | • Send Status<br>• Values:<br>– QMI_ERR_NONE – No error in the request<br>– QMI_ERR_CAUSE_CODE - SMS cause code<br>– QMI_ERR_MESSAGE_DELIVERY_FAILURE - Message could not be delivered<br>– QMI_ERR_NO_MEMORY - Device could not allocate memory to formulate a response<br>• Bit to check in ParamPresenceMask - **1** |

| | |
|---|---|
| *messageID* | • Unique ID assigned by WMS for non-retry messages.<br>• Bit to check in ParamPresenceMask - **16** |
| *causeCode* | • WMS cause code<br>• Bit to check in ParamPresenceMask - **17** |
| *errorClass* | • Error Class<br>• Values:<br>– 0x00 - ERROR_CLASS_TEMPORARY<br>– 0x01 - ERROR_CLASS_PERMANENT<br>• Bit to check in ParamPresenceMask - **18** |
| *RPCause* | • GW RP cause<br>• Bit to check in ParamPresenceMask - **19** |
| *TPCause* | • GW TP Cause<br>• Bit to check in ParamPresenceMask - **19** |
| *msgDelFailure-Type* | • Message delivery failure type<br>• Values:<br>– 0x00 - WMS_MESSAGE_DELIVERY_FAILURE_TEMPORARY<br>– 0x01 - WMS_MESSAGE_DELIVERY_FAILURE_PERMANENT<br>• Bit to check in ParamPresenceMask - **20** |
| *msgDelFailure-Cause* | • Message delivery failure cause<br>• Values:<br>– 0x00 - WMS_MESSAGE_BLOCKED_DUE_TO_CALL_CONTROL<br>• Bit to check in ParamPresenceMask - **21** |
| *alphaIDLen* | • Number of sets of the pAlphaID<br>• Bit to check in ParamPresenceMask - **22** |
| *pAlphaID* | • Alpha ID<br>• Bit to check in ParamPresenceMask - **22** |
| *userData* | • Identifies the request associated with this indication.<br>• Bit to check in ParamPresenceMask - **23** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.837.2 Field Documentation

#### 8.837.2.1 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::alphaIDLen

#### 8.837.2.2 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::causeCode

#### 8.837.2.3 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::errorClass

#### 8.837.2.4 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::messageID

#### 8.837.2.5 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::msgDelFailureCause

#### 8.837.2.6 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::msgDelFailureType

#### 8.837.2.7 uint8_t* unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::pAlphaID

#### 8.837.2.8 swi_uint256_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::ParamPresenceMask

#### 8.837.2.9 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::RPCause

#### 8.837.2.10 uint16_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::sendStatus

#### 8.837.2.11 uint8_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::TPCause

#### 8.837.2.12 uint32_t unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t::userData

## 8.838 unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference

### Data Fields

- uint8_t storageType
- uint8_t messageMode
- swi_uint256_t ParamPresenceMask

### 8.838.1 Detailed Description

This structure contains unpack SMS memory full callback indication.

**Parameters**

| | |
|---|---|
| *storageType* | • SMS message storage type<br>– 0 - UIM - Invalid in case of CDMA device that does not require SIM<br>– 1 - NV<br>• Bit to check in ParamPresenceMask - **1** |
| *messageMode* | • 0x00 - CDMA, LTE (if network type is CDMA)<br>• 0x01 - GW, LTE (if network type is UMTS)<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.838.2 Field Documentation

#### 8.838.2.1 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode

#### 8.838.2.2 swi_uint256_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::ParamPresenceMask

#### 8.838.2.3 uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType

## 8.839 unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t Struct Reference

### Data Fields

- uint8_t numInstances
- sms_messageWaitingInfoContent msgWaitInfo [0xFF]
- swi_uint256_t ParamPresenceMask

### 8.839.1 Detailed Description

This structure holds information related to message waiting information indication

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of sets of the elements in structure sms_messageWaitingInfoContent<br>• Bit to check in ParamPresenceMask - **1** |
| *msgWaitInfo* | • Pointer to structure of sms_messageWaitingInfoContent.<br>  – See sms_messageWaitingInfoContent for more information.<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.839.2 Field Documentation

#### 8.839.2.1 sms_messageWaitingInfoContent unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::msgWaitInfo[0xFF]

#### 8.839.2.2 uint8_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::numInstances

#### 8.839.2.3 swi_uint256_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::ParamPresenceMask

## 8.840 unpack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference

### Data Fields

- uint8_t Profile
- uint8_t EarMute
- uint8_t MicMute
- uint8_t Generator
- uint8_t Volume
- uint8_t CwtMute

- swi_uint256_t ParamPresenceMask

### 8.840.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile unpack parameters.

**Parameters**

| | |
|---|---|
| *Profile* | • Audio Profile<br>  – 0-5<br>• Bit to check in ParamPresenceMask - **1** |
| *EarMute* | • Ear Mute<br>  – 0 - Mute<br>  – 1 - Unmute<br>• Bit to check in ParamPresenceMask - **2** |
| *MicMute* | • MIC Mute<br>  – 0 - Mute<br>  – 1 - Unmute<br>• Bit to check in ParamPresenceMask - **3** |
| *Generator* | • Generator<br>  – 0 - Voice<br>• Bit to check in ParamPresenceMask - **4** |
| *Volume* | • RX volume level<br>  – 0-5<br>• Bit to check in ParamPresenceMask - **5** |
| *CwtMute* | • Call waiting tone Mute<br>  – 0 - Mute<br>  – 1 - Unmute<br>• Bit to check in ParamPresenceMask - **6** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.840.2 Field Documentation

#### 8.840.2.1 uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::CwtMute

#### 8.840.2.2 uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::EarMute

#### 8.840.2.3 uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::Generator

#### 8.840.2.4 uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::MicMute

#### 8.840.2.5 swi_uint256_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::ParamPresenceMask

#### 8.840.2.6 uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::Profile

#### 8.840.2.7 uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::Volume

## 8.841 unpack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference

### Data Fields

- uint8_t Level
- swi_uint256_t ParamPresenceMask

### 8.841.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume unpack parameters.

**Parameters**

| | |
|---|---|
| *Level* | • The RX Volume Level<br>  – 0-5<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.841.2 Field Documentation

#### 8.841.2.1 uint8_t unpack_swiaudio_SLQSGetM2MAudioVolume_t::Level

#### 8.841.2.2 swi_uint256_t unpack_swiaudio_SLQSGetM2MAudioVolume_t::ParamPresenceMask

## 8.842 unpack_swiaudio_SLQSGetM2MAVMute_t Struct Reference

### Data Fields

- uint8_t EarMute
- uint8_t MicMute
- uint8_t CwtMute
- swi_uint256_t ParamPresenceMask

### 8.842.1 Detailed Description

This structure contains the SLQSGetM2MAVMute unpack parameters.

**Parameters**

| | |
|---|---|
| *pEarMute* | • Ear Mute<br>– 0-Mute<br>– 1-UnMute<br>• Bit to check in ParamPresenceMask - **1** |
| *pMicMute* | • Mic Mute<br>– 0-Mute<br>– 1-unmute<br>• Bit to check in ParamPresenceMask - **2** |
| *CwtMute* | • Waiting tone Mute<br>– 0-5<br>• Bit to check in ParamPresenceMask - **3** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.842.2 Field Documentation

#### 8.842.2.1 uint8_t unpack_swiaudio_SLQSGetM2MAVMute_t::CwtMute

#### 8.842.2.2 uint8_t unpack_swiaudio_SLQSGetM2MAVMute_t::EarMute

#### 8.842.2.3 uint8_t unpack_swiaudio_SLQSGetM2MAVMute_t::MicMute

#### 8.842.2.4 swi_uint256_t unpack_swiaudio_SLQSGetM2MAVMute_t::ParamPresenceMask

## 8.843 unpack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference

### Data Fields

- uint16_t Value
- swi_uint256_t ParamPresenceMask

### 8.843.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain unpack parameters.

**Parameters**

| | |
|---|---|
| *Value* | • RX speakerphone gain<br>– 0x0 - 0x7fff<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.843.2 Field Documentation

#### 8.843.2.1 swi_uint256_t unpack_swiaudio_SLQSGetM2MSpkrGain_t::ParamPresenceMask

#### 8.843.2.2 uint16_t unpack_swiaudio_SLQSGetM2MSpkrGain_t::Value

## 8.844 unpack_swiavms_SLQSAVMSEventReportInd_t Struct Reference

### Data Fields

- UnpackSwiAvmsEventReportBinaryUpdateSessionInfo * pBinaryUpdateSessionInfo
- UnpackSwiAvmsEventReportConfig * pConfig
- UnpackSwiAvmsEventReportNotification * pNotification
- UnpackSwiAvmsEventReportPackageID * pPackageID
- UnpackSwiAvmsEventReportConnectionRequest * pConnectionRequest
- UnpackSwiAvmsEventReportWAMSParamChange * pWAMSParaChanged
- UnpackSwiAvmsEventReportRegStatus * pRegStatus
- UnpackSwiAvmsEventReportDataSessionStatus * pDataSessionStatus
- UnpackSwiAvmsEventReportSessionType * pSessionType
- UnpackSwiAvmsEventReportHTTPStatus * pHTTPStatus
- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.844.1 Detailed Description

Structure that contains the session type response for AVMS event report indication command

**Parameters**

| | |
|---|---|
| *pBinaryUpdateSessionInfo[OUT]* | • See UnpackSwiAvmsEventReportBinaryUpdateSessionInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pConfig[OUT]* | • See UnpackSwiAvmsEventReportConfig for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pNotification[OUT]* | • See UnpackSwiAvmsEventReportNotification for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *pPackageID[OUT]* | • See UnpackSwiAvmsEventReportPackageID for more information<br>• Bit to check in ParamPresenceMask - **21** |
| *pConnectionRequest[OUT]* | • See UnpackSwiAvmsEventReportConnectionRequest for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *pWAMSParaChanged[OUT]* | • See UnpackSwiAvmsEventReportWAMSParamChange for more information<br>• Bit to check in ParamPresenceMask - **20** |

| | |
|---|---|
| *pRegStatus[OUT]* | • See UnpackSwiAvmsEventReportRegStatus for more information<br>• Bit to check in ParamPresenceMask - **22** |
| *pDataSessionStatus[OUT]* | • See UnpackSwiAvmsEventReportDataSessionStatus for more information<br>• Bit to check in ParamPresenceMask - **23** |
| *pSessionType[OUT]* | • See UnpackSwiAvmsEventReportSessionType for more information<br>• Bit to check in ParamPresenceMask - **24** |
| *pHTTPStatus[OUT]* | • See UnpackSwiAvmsEventReportHTTPStatus for more information<br>• Bit to check in ParamPresenceMask - **25** |
| *resultcode* | - Tlv Result Code. |

### 8.844.2 Field Documentation

#### 8.844.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSEventReportInd_t::ParamPresenceMask

#### 8.844.2.2 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo* unpack_swiavms_SLQSAVMSEventReportInd_t::pBinaryUpdateSessionInfo

#### 8.844.2.3 UnpackSwiAvmsEventReportConfig* unpack_swiavms_SLQSAVMSEventReportInd_t::pConfig

#### 8.844.2.4 UnpackSwiAvmsEventReportConnectionRequest* unpack_swiavms_SLQSAVMSEventReportInd_t::pConnectionRequest

#### 8.844.2.5 UnpackSwiAvmsEventReportDataSessionStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pDataSessionStatus

#### 8.844.2.6 UnpackSwiAvmsEventReportHTTPStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pHTTPStatus

#### 8.844.2.7 UnpackSwiAvmsEventReportNotification* unpack_swiavms_SLQSAVMSEventReportInd_t::pNotification

#### 8.844.2.8 UnpackSwiAvmsEventReportPackageID* unpack_swiavms_SLQSAVMSEventReportInd_t::pPackageID

#### 8.844.2.9 UnpackSwiAvmsEventReportRegStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pRegStatus

#### 8.844.2.10 UnpackSwiAvmsEventReportSessionType* unpack_swiavms_SLQSAVMSEventReportInd_t::pSessionType

#### 8.844.2.11 UnpackSwiAvmsEventReportWAMSParamChange* unpack_swiavms_SLQSAVMSEventReportInd_t::pWAMSParaChanged

#### 8.844.2.12 uint32_t unpack_swiavms_SLQSAVMSEventReportInd_t::resultcode

## 8.845 unpack_swiavms_SLQSAVMSGetSettings_t Struct Reference

### Data Fields

- uint32_t OMADMEnabled

- uint8_t AutoConnect
- uint8_t AutoReboot
- uint8_t FwPromptdownload
- uint8_t FwPromptUpdate
- uint8_t FwAutoSDM
- uint32_t * pPollingTimer
- PackSwiAVMSSettingsConnectionRetryTimers * pConnectionRetryTimers
- PackSwiAVMSSettingsAPNInfo * pAPNInfo
- uint8_t * pNotificationStore
- PackSwiAVMSSettingsPeriodsInfo * pPeroidsInfo
- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.845.1 Detailed Description

Structure that contains the session type response for AVMS Set Settings command

**Parameters**

| | |
|---|---|
| *OMADMEnabled* | - OMA DM Enabled.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *AutoConnect* | - Auto Connect.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **6**

**Parameters**

| | |
|---|---|
| *AutoReboot* | - Auto Reboot.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **21**

**Parameters**

| | |
|---|---|
| *Fw-Autodownload* | - Firmware auto download.<br>• 0 - Disabled.<br>• 1 - Enabled.<br>– Bit to check in ParamPresenceMask - **3** |
| *FwAutoUpdate* | - Firmware auto update.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **4**

**Parameters**

| | |
|---|---|
| *FwAutoSDM* | - OMA Automatic UI Alert Response.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **5**

**Parameters**

| | |
|---|---|
| *pPollingTimer* | - Polling timer to connect to AVMS server.<br>• 0-525600 (min)<br>• 0:disabled |

- Bit to check in ParamPresenceMask - **16**

**Parameters**

| | |
|---|---|
| *pConnection-RetryTimers* | - Connection Retry timers.<br>• See PackSwiAVMSSettingsConnectionRetryTimers for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pAPNInfo* | - APN Information.<br>• See PackSwiAVMSSettingsAPNInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *pNotification-Store* | - Notification Storing When Disabled or Offline.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **19**

**Parameters**

| | |
|---|---|
| *pPeroidsInfo* | - Min and Max Period of an Observation.<br>• See PackSwiAVMSSettingsPeriodsInfo for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *resultcode* | - Tlv Result Code. |

## 8.845.2 Field Documentation

### 8.845.2.1 uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::AutoConnect

### 8.845.2.2 uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::AutoReboot

### 8.845.2.3 uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwAutoSDM

#### 8.845.2.4 uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwPromptdownload

#### 8.845.2.5 uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwPromptUpdate

#### 8.845.2.6 uint32_t unpack_swiavms_SLQSAVMSGetSettings_t::OMADMEnabled

#### 8.845.2.7 PackSwiAVMSSettingsAPNInfo* unpack_swiavms_SLQSAVMSGetSettings_t::pAPNInfo

#### 8.845.2.8 swi_uint256_t unpack_swiavms_SLQSAVMSGetSettings_t::ParamPresenceMask

#### 8.845.2.9 PackSwiAVMSSettingsConnectionRetryTimers* unpack_swiavms_SLQSAVMSGetSettings_t::pConnection-RetryTimers

#### 8.845.2.10 uint8_t* unpack_swiavms_SLQSAVMSGetSettings_t::pNotificationStore

#### 8.845.2.11 PackSwiAVMSSettingsPeriodsInfo* unpack_swiavms_SLQSAVMSGetSettings_t::pPeroidsInfo

#### 8.845.2.12 uint32_t* unpack_swiavms_SLQSAVMSGetSettings_t::pPollingTimer

#### 8.845.2.13 uint32_t unpack_swiavms_SLQSAVMSGetSettings_t::resultcode

## 8.846 unpack_swiavms_SLQSAVMSGetSettings_v2_t Struct Reference

### Data Fields

- uint32_t OMADMEnabled
- uint8_t AutoConnect
- uint8_t FwPromptdownload
- uint8_t FwPromptUpdate
- uint8_t FwAutoSDM
- uint32_t * pPollingTimer
- PackSwiAVMSSettingsConnectionRetryTimers * pConnectionRetryTimers
- PackSwiAVMSSettingsAPNInfo * pAPNInfo
- uint8_t * pNotificationStore
- PackSwiAVMSSettingsPeriodsInfo * pPeroidsInfo
- uint8_t * pAutoReboot
- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.846.1 Detailed Description

Structure that contains the sresponse for AVMS Get Settings command It maps with SLQSAVMSSetSettings_v2 (For AVC2 service)

**Parameters**

| | |
|---|---|
| *OMADMEnabled* | - OMA DM Enabeld.<br>• 0 - Disabled.<br>• 1 - Enabled. |

- Bit to check in ParamPresenceMask - **1**

**Parameters**

| | |
|---|---|
| *AutoConnect* | - Auto Connect.<br>• 0 - Disabeld.<br>• 1 - Enabeld. |

- Bit to check in ParamPresenceMask - **6**

**Parameters**

| | |
|---|---|
| *Fw-Autodownload* | - Firmware auto downalod.<br>• 0 - Disabeld.<br>• 1 - Enabeld.<br>– Bit to check in ParamPresenceMask - **3** |
| *FwAutoUpdate* | - Firmware auto update.<br>• 0 - Disabeld.<br>• 1 - Enabeld. |

- Bit to check in ParamPresenceMask - **4**

**Parameters**

| | |
|---|---|
| *FwAutoSDM* | - OMA Automatic UI Alert Response.<br>• 0 - Disabeld.<br>• 1 - Enabeld. |

- Bit to check in ParamPresenceMask - **5**

**Parameters**

| | |
|---|---|
| *pPollingTimer[OPTIONAL]* | - Polling timer to connect to AVMS server.<br>• 0-525600 (min)<br>• 0:disabled<br>– Bit to check in ParamPresenceMask - **16** |
| *pConnectionRetryTimers[OPTIONAL]* | - Connection Retry timers.<br>• See PackSwiAVMSSettingsConnectionRetryTimers for more information.<br>• Bit to check in ParamPresenceMask - **17** |
| *pAPNInfo[OPTIONAL]* | - APN Information.<br>• See PackSwiAVMSSettingsAPNInfo for more information.<br>• Bit to check in ParamPresenceMask - **18** |
| *pNotificationStore[OPTIONAL]* | - Notification Storing When Disabled or Offline.<br>• 0 - Disabled.<br>• 1 - Enabeld.<br>– Bit to check in ParamPresenceMask - **19** |

| | |
|---|---|
| *pPeroidsInfo[OPTIONAL]* | - Min and Max Period of an Observation.<br>• See PackSwiAVMSSettingsPeriodsInfo for more information.<br>• Bit to check in ParamPresenceMask - **20** |
| *pAutoReboot[OPTIONAL]* | - Auto Reboot.<br>• 0 - Disabeld.<br>• 1 - Enabeld.<br>– Bit to check in ParamPresenceMask - **21** |
| *resultcode* | - Tlv Result Code. |

### 8.846.2 Field Documentation

#### 8.846.2.1 uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::AutoConnect

#### 8.846.2.2 uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::FwAutoSDM

#### 8.846.2.3 uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::FwPromptdownload

#### 8.846.2.4 uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::FwPromptUpdate

#### 8.846.2.5 uint32_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::OMADMEnabled

#### 8.846.2.6 PackSwiAVMSSettingsAPNInfo* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pAPNInfo

#### 8.846.2.7 swi_uint256_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::ParamPresenceMask

#### 8.846.2.8 uint8_t* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pAutoReboot

#### 8.846.2.9 PackSwiAVMSSettingsConnectionRetryTimers* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pConnectionRetryTimers

#### 8.846.2.10 uint8_t* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pNotificationStore

#### 8.846.2.11 PackSwiAVMSSettingsPeriodsInfo* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pPeroidsInfo

#### 8.846.2.12 uint32_t* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pPollingTimer

#### 8.846.2.13 uint32_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::resultcode

## 8.847 unpack_swiavms_SLQSAVMSSendSelection_t Struct Reference

### Data Fields

- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.847.1 Detailed Description

Structure that contains the session type response for AVMS Selection command

**Parameters**

| | |
|---|---|
| *resultcode* | - Tlv Result Code.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.847.2 Field Documentation

#### 8.847.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSSendSelection_t::ParamPresenceMask

#### 8.847.2.2 uint32_t unpack_swiavms_SLQSAVMSSendSelection_t::resultcode

## 8.848 unpack_swiavms_SLQSAVMSSessionGetInfo_t Struct Reference

### Data Fields

- UnpackSwiAvmsEventReportBinaryUpdateSessionInfo * pBinaryUpdateSessionInfo
- UnpackSwiAvmsEventReportConfig * pConfig
- UnpackSwiAvmsEventReportNotification * pNotification
- UnpackSwiAvmsEventReportPackageID * pPackageID
- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.848.1 Detailed Description

Structure that contains the session type response for AVMS get session info command

**Parameters**

| | |
|---|---|
| *pBinaryUpdateSessionInfo[OUT]* | • See UnpackSwiAvmsEventReportBinaryUpdateSessionInfo for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pConfig[OUT]* | • See UnpackSwiAvmsEventReportConfig for more information<br>• Bit to check in ParamPresenceMask - **17** |
| *pNotification[OUT]* | • See UnpackSwiAvmsEventReportNotification for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *pPackageID[OUT]* | • See UnpackSwiAvmsEventReportPackageID for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *resultcode* | - Tlv Result Code. |

### 8.848.2 Field Documentation

#### 8.848.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSSessionGetInfo_t::ParamPresenceMask

#### 8.848.2.2 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo* unpack_swiavms_SLQSAVMSSessionGetInfo_t::pBinaryUpdateSessionInfo

#### 8.848.2.3 UnpackSwiAvmsEventReportConfig* unpack_swiavms_SLQSAVMSSessionGetInfo_t::pConfig

#### 8.848.2.4 UnpackSwiAvmsEventReportNotification* unpack_swiavms_SLQSAVMSSessionGetInfo_t::pNotification

#### 8.848.2.5 UnpackSwiAvmsEventReportPackageID* unpack_swiavms_SLQSAVMSSessionGetInfo_t::pPackageID

#### 8.848.2.6 uint32_t unpack_swiavms_SLQSAVMSSessionGetInfo_t::resultcode

## 8.849 unpack_swiavms_SLQSAvmsSetEventReport_t Struct Reference

### Data Fields

- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.849.1 Detailed Description

Structure that contains the session type response for AVMS set event report command

**Parameters**

| | |
|---|---|
| *resultcode* | - Tlv Result Code.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.849.2 Field Documentation

#### 8.849.2.1 swi_uint256_t unpack_swiavms_SLQSAvmsSetEventReport_t::ParamPresenceMask

#### 8.849.2.2 uint32_t unpack_swiavms_SLQSAvmsSetEventReport_t::resultcode

## 8.850 unpack_swiavms_SLQSAVMSSetSettings_t Struct Reference

### Data Fields

- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.850.1 Detailed Description

Structure that contains the session type response for AVMS Set Settings command

**Parameters**

| | |
|---|---|
| *resultcode* | - Tlv Result Code.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.850.2 Field Documentation

#### 8.850.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSSetSettings_t::ParamPresenceMask

#### 8.850.2.2 uint32_t unpack_swiavms_SLQSAVMSSetSettings_t::resultcode

## 8.851 unpack_swiavms_SLQSAVMSSetSettings_v2_t Struct Reference

### Data Fields

- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.851.1 Detailed Description

Structure that contains the session type response for AVMS Set Settings command

**Parameters**

| | |
|---|---|
| *resultcode* | - Tlv Result Code.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.851.2 Field Documentation

#### 8.851.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSSetSettings_v2_t::ParamPresenceMask

#### 8.851.2.2 uint32_t unpack_swiavms_SLQSAVMSSetSettings_v2_t::resultcode

## 8.852 unpack_swiavms_SLQSAVMSStartSession_t Struct Reference

### Data Fields

- uint32_t sessionResponse
- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.852.1 Detailed Description

Structure that contains the session type response for AVMS start session command

**Parameters**

| | |
|---|---|
| *Session-Response* | - OMA-DM availability CHECK for the requested item.<br>• 0x00000001 - Available.<br>• 0x00000002 - Not Available.<br>• 0x00000003 - Check Timed Out.<br>• Bit to check in ParamPresenceMask - **1** |
| *resultcode* | - Tlv Result Code. |

### 8.852.2 Field Documentation

#### 8.852.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSStartSession_t::ParamPresenceMask

#### 8.852.2.2 uint32_t unpack_swiavms_SLQSAVMSStartSession_t::resultcode

#### 8.852.2.3 uint32_t unpack_swiavms_SLQSAVMSStartSession_t::sessionResponse

## 8.853 unpack_swiavms_SLQSAVMSStopSession_t Struct Reference

### Data Fields

- uint32_t resultcode
- swi_uint256_t ParamPresenceMask

### 8.853.1 Detailed Description

Structure that contains the session type response for AVMS Stop session command

**Parameters**

| | |
|---|---|
| *resultcode* | - Tlv Result Code.<br>• Bit to check in ParamPresenceMask - **2** |

### 8.853.2 Field Documentation

#### 8.853.2.1 swi_uint256_t unpack_swiavms_SLQSAVMSStopSession_t::ParamPresenceMask

#### 8.853.2.2 uint32_t unpack_swiavms_SLQSAVMSStopSession_t::resultcode

## 8.854 unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference

### Data Fields

- swidms_mtuSize3gppTlv * pMTUSize3gpp
- swidms_hrpdMTUSizeTlv * pHrpdMTUSize
- swidms_ehrpdMTUSizeTlv * pEhrpdMTUSize
- swidms_usbMTUSizeTlv * pUsbMTUSize
- swi_uint256_t ParamPresenceMask

### 8.854.1 Detailed Description

This structure contains the Get MTU Response parameter.

**Parameters**

| | |
|---|---|
| *pMTUSize3gpp* | [Optional]<br>• See swidms_mtuSize3gppTlv for more information<br>• Bit to check in ParamPresenceMask - **16** |
| *pHrpdMTUSize* | [Optional]<br>• See swidms_hrpdMTUSizeTlv for more information<br>• Bit to check in ParamPresenceMask - **17** |

| | |
|---|---|
| *pEhrpdMTUSize* | [Optional]<br>• See swidms_ehrpdMTUSizeTlv for more information<br>• Bit to check in ParamPresenceMask - **18** |
| *pUsbMTUSize* | [Optional]<br>• See swidms_usbMTUSizeTlv for more information<br>• Bit to check in ParamPresenceMask - **19** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.854.2 Field Documentation

#### 8.854.2.1 swi_uint256_t unpack_swidms_SLQSSwiDmsGetMTU_t::ParamPresenceMask

#### 8.854.2.2 swidms_ehrpdMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pEhrpdMTUSize

#### 8.854.2.3 swidms_hrpdMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pHrpdMTUSize

#### 8.854.2.4 swidms_mtuSize3gppTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pMTUSize3gpp

#### 8.854.2.5 swidms_usbMTUSizeTlv* unpack_swidms_SLQSSwiDmsGetMTU_t::pUsbMTUSize

## 8.855 unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference

### Data Fields

- uint8_t usbNetNum
- swi_uint256_t ParamPresenceMask

### 8.855.1 Detailed Description

This structure contains usb net numbers to get from remote endpoint for QMAP configuration

**Parameters**

| | |
|---|---|
| *usbNetNum* | • value of usb net numbers on the device<br>• Bit to check in ParamPresenceMask - **1** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.855.2 Field Documentation

#### 8.855.2.1 swi_uint256_t unpack_swidms_SLQSSwiDmsGetUsbNetNum_t::ParamPresenceMask

#### 8.855.2.2 uint8_t unpack_swidms_SLQSSwiDmsGetUsbNetNum_t::usbNetNum

## 8.856 unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference

### Data Fields

- uint16_t Tlvresult
- swi_uint256_t ParamPresenceMask

### 8.856.1 Detailed Description

This structure is used to store unpack_swidms_SLQSSwiDmsSetMTU_t parameters.

**Parameters**

| | |
|---|---|
| *Tlvresult* | • unpack Tlv result<br>• Bit to check in ParamPresenceMask - **2** |
| *ParamPresence-Mask* | • bitmask representation to indicate valid parameters. |

### 8.856.2 Field Documentation

#### 8.856.2.1 swi_uint256_t unpack_swidms_SLQSSwiDmsSetMTU_t::ParamPresenceMask

#### 8.856.2.2 uint16_t unpack_swidms_SLQSSwiDmsSetMTU_t::Tlvresult

## 8.857 unpack_swiloc_SwiLocGetAutoStart_t Struct Reference

### Data Fields

- uint8_t function
- int function_reported
- uint8_t fix_type
- int fix_type_reported
- uint8_t max_time
- int max_time_reported
- uint32_t max_dist
- int max_dist_reported
- uint32_t fix_rate
- int fix_rate_reported
- swi_uint256_t ParamPresenceMask

### 8.857.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

**Parameters**

| | |
|---|---|
| *function* | • Setting to indicate when modem should start an automatic GNSS fix<br>– 0 - disabled<br>– 1 - At bootup<br>– 2 - When NMEA port is opened<br>• Bit to check in ParamPresenceMask - **16** |
| *function_reported* | • 0 - not reported by modem<br>• I - reported by modem |
| *fix_type* | • Type of GNSS fix:<br>– 1 - Default Engine mode<br>– 2 - MS-Based<br>– 3 - MS-Assisted<br>– 4 - Standalone<br>• Bit to check in ParamPresenceMask - **17** |
| *fix_type_reported* | • 0 - not reported by modem<br>• I - reported by modem |
| *max_time* | • Maximum time allowed for the receiver to get a fix in seconds<br>• Valid range: 1-255<br>• Bit to check in ParamPresenceMask - **18** |
| *max_time_reported* | • 0 - not reported by modem<br>• I - reported by modem |
| *max_dist* | • Maximum uncertainty of a fix measured by distance in meters<br>• Valid range: 1 - 4294967280<br>• Bit to check in ParamPresenceMask - **19** |
| *max_dist_reported* | • 0 - not reported by modem<br>• I - reported by modem |
| *fix_rate* | • Time between fixes in seconds<br>• Valid range: 1–65535<br>• Bit to check in ParamPresenceMask - **20** |
| *fix_rate_reported* | • 0 - not reported by modem<br>• I - reported by modem |

### 8.857.2 Field Documentation

#### 8.857.2.1 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate

#### 8.857.2.2 int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported

#### 8.857.2.3 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type

#### 8.857.2.4 int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported

#### 8.857.2.5 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function

#### 8.857.2.6 int unpack_swiloc_SwiLocGetAutoStart_t::function_reported

#### 8.857.2.7 uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist

#### 8.857.2.8 int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported

#### 8.857.2.9 uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time

#### 8.857.2.10 int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported

#### 8.857.2.11 swi_uint256_t unpack_swiloc_SwiLocGetAutoStart_t::ParamPresenceMask

## 8.858 unpack_swioma_SLQSOMADMAlertCallback_ind_t Struct Reference

### Data Fields

- uint32_t eventType
- unpack_omaDmFotaTlv_t SessionInfoFota
- unpack_omaDmConfigTlv_t SessionInfoConfig
- unpack_omaDmNotificationsTlv_t SessionInfoNotification

### 8.858.1 Detailed Description

Structure that contains OMA indication information based on eventType Strucutres for which the event is not valid will have values set to 0

**Parameters**

| | |
|---|---|
| *eventType* | • 0x00 - SWIOMA-DM FOTA<br>• 0x01 - SWIOMA-DM Config<br>• 0x02 - SWIOMA-DM Notification<br>• 0xff - indication missing event information. |
| *SessionInfoFota[OUT]* | • See unpack_omaDmFotaTlv_t for more information |
| *SessionInfoConfig[OUT]* | • See unpack_omaDmConfigTlv_t for more information |
| *SessionInfoNotification[OUT]* | • See unpack_omaDmNotificationsTlv_t for more information |

### 8.858.2 Field Documentation

#### 8.858.2.1 uint32_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::eventType

#### 8.858.2.2 unpack_omaDmConfigTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoConfig

#### 8.858.2.3 unpack_omaDmFotaTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoFota

#### 8.858.2.4 unpack_omaDmNotificationsTlv_t unpack_swioma_SLQSOMADMAlertCallback_ind_t::SessionInfoNotification

## 8.859 unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference

### Data Fields

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

### 8.859.1 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**

| | |
|---|---|
| *Status* | • 1 Byte parameter indicating status<br>– 0x01 - No Firmware available<br>– 0x02 - Query Firmware Download<br>– 0x03 - Firmware Downloading<br>– 0x04 - Firmware Downloaded<br>– 0x05 - Query Firmware Update<br>– 0x06 - Firmware Updating<br>– 0x07 - Firmware Updated<br>– 0xff - Invalid Session information. |
| *Update-CompleteStatus* | • 2 byte parameter indicating Update Complete Status<br>– See qaGobiApiTableSwiOMADMUpdateCompleteStatus.h Update Complete Status |

| | |
|---|---|
| *Severity* | • 1 byte parameter indicating severity<br>– 0x01 - Mandatory<br>– 0x02 - Optional |
| *SourceLength* | • 2 byte parameter indicating Length of Vendor Name String in Bytes. |
| *Source* | • Variable length parameter indicating Vendor Name in ASCII<br>• See LITEQMI_MAX_SWIOMA_STR_LEN for more information |
| *PkgNameLength* | • 2 byte parameter indicating Length of Package Name String in Bytes. |
| *PkgName* | • Variable length parameter indicating Package Name in ASCII<br>• See LITEQMI_MAX_SWIOMA_STR_LEN for more information |
| *PkgDescLength* | • 2 byte parameter indicating Length of Package Description String in Bytes. |
| *PkgDescription* | • Variable length parameter indicating Package Description in ASCII<br>• See LITEQMI_MAX_SWIOMA_STR_LEN for more information |
| *DateLength* | • 2 byte parameter indicating Length of Package Description String in Bytes. |
| *Date* | • Variable length parameter indicating Package Description in ASCII<br>• See LITEQMI_MAX_SWIOMA_STR_LEN for more information |
| *TimeLength* | • 2 byte parameter indicating Length of Time String in Bytes. |
| *Time* | • Variable length parameter indicating Time String in ASCII<br>• See LITEQMI_MAX_SWIOMA_STR_LEN for more information |
| *SessionType* | • 1 byte parameter reflects the last session started for Sprint<br>– 0x00 - No session since boot<br>– 0x01 - Sprint CI-DC Session<br>– 0x02 - Sprint CI-PRL Session<br>– 0x03 - Sprint CI-FUMO Session<br>– 0x04 - Sprint HFA-DC Session<br>– 0x05 - Sprint HFA-PRL Session<br>– 0x06 - Sprint HFA-FUMO Session<br>– 0x07 - Sprint NI Session |

| | |
|---|---|
| *SessionState* | • 1 byte parameter indicating session state<br>– 0x01 - idle<br>– 0x02 - active<br>– 0x03 - pending |
| *RetryCount* | • 1 byte parameter indicating retries left count<br>– valid values 0 to 6 |

### 8.859.2 Field Documentation

#### 8.859.2.1 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]

#### 8.859.2.2 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength

#### 8.859.2.3 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength

#### 8.859.2.4 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]

#### 8.859.2.5 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]

#### 8.859.2.6 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength

#### 8.859.2.7 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount

#### 8.859.2.8 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState

#### 8.859.2.9 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType

#### 8.859.2.10 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity

#### 8.859.2.11 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]

#### 8.859.2.12 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength

#### 8.859.2.13 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Status

#### 8.859.2.14 uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Time[255]

#### 8.859.2.15 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::TimeLength

#### 8.859.2.16 uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::UpdateCompleteStatus

## 8.860 unpack_swioma_SLQSOMADMGetSettings_t Struct Reference

### Data Fields

- uint32_t OMADMEnabled
- uint8_t FOTAdownload
- uint8_t FOTAUpdate
- uint8_t Autosdm

- uint8_t FwAutoCheck

### 8.860.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

**Parameters**

| | |
|---|---|
| *OMADMEnabled* | [out]<br>• Optional 4 byte parameter indicating OMADM service enabled<br>– 0x00000001 - Client-initiated device configuration<br>– 0x00000002 - Network-initiated device configuration<br>– 0x00000010 - Client-initiated FUMO<br>– 0x00000020 - Network-initiated FUMO<br>• function SLQSOMADMGetSettings2() returns a default value 0xFFFFFFFF in case this parameter is not returned by the modem. |
| *FOTAdownload* | [out]<br>• Optional 1 Byte parameter indicating support for FOTA Automatic download<br>– 0x00 - Host permission required before downloading<br>– 0x01 - Automatically start downloading, no host permission required<br>– 0x02 - Automatically start downloading, while not roaming<br>– 0x03 - Automatically reject download<br>– 0x04 - Automatically reject download with “Enterprise Reject Policy<br>– 0xff - Invalid data. |

- function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem.

**Parameters**

| | |
|---|---|
| *FOTAUpdate* | [out]<br>• Optional 1 byte parameter indicating FOTA Automatic update<br>– 0x00 - User permission required before updating firmware<br>– 0x01 - No user permission required before updating firmware<br>– 0x02 - User permission required, auto update on power up<br>• function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem. |
| *Autosdm* | [out]<br>• Optional 1 byte parameter indicating OMA Automatic UI Alert Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled Accept<br>– 0x02 - Enabled Reject<br>• function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem. |

| | |
|---|---|
| *FwAutoCheck* | [out]<br>• Optional 1 byte parameter indicating OMA Automatic Check for Firmware Update on Power-Up Response<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>• function SLQSOMADMGetSettings2() returns a default value 0xFF in case this parameter is not returned by the modem. |

### 8.860.2 Field Documentation

#### 8.860.2.1 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::Autosdm

#### 8.860.2.2 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAdownload

#### 8.860.2.3 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FOTAUpdate

#### 8.860.2.4 uint8_t unpack_swioma_SLQSOMADMGetSettings_t::FwAutoCheck

#### 8.860.2.5 uint32_t unpack_swioma_SLQSOMADMGetSettings_t::OMADMEnabled

## 8.861 unpack_swioma_SLQSOMADMStartSession_t Struct Reference

### Data Fields

- uint32_t FwAvailability

### 8.861.1 Detailed Description

Structure that contains the responses for OMA start session command

**Parameters**

| | |
|---|---|
| *FwAvailability* | [out]<br>• OMA-DM CHECK FW Available<br>– 0x00000001 - FW Available. For CIDC and CIPRL, this value will be returned by the modem. CIDC and CIPRL are asynchronous OMADM sessions.<br>– 0x00000002 - FW Not Available<br>– 0x00000003 - FW Check Timed Out<br>– 0xFFFFFFFF - Invalid data. |

### 8.861.2 Field Documentation

#### 8.861.2.1 uint32_t unpack_swioma_SLQSOMADMStartSession_t::FwAvailability

## 8.862 unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.862.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.862.2 Field Documentation

#### 8.862.2.1 uint16_t unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t::Tlvresult

## 8.863 unpack_tmd_SLQSTmdGetMitigationDevList_t Struct Reference

### Data Fields

- uint8_t MitigationDevListLen
- tmd_mitigationDevList MitigationDevList [255]
- uint16_t Tlvresult

### 8.863.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

**Parameters**

| | |
|---|---|
| *MitigationDevListLen* | • Mitigation Device List Length (Optional)<br>• Number of sets of the following elements<br>  – MitigationDevList |
| *pMitigationDevList* | • Mitigation Device List (Optional)<br>• See tmd_mitigationDevList for more information. |
| *Tlvresult* | • Unpack Result |

### 8.863.2 Field Documentation

#### 8.863.2.1 tmd_mitigationDevList unpack_tmd_SLQSTmdGetMitigationDevList_t::MitigationDevList[255]

#### 8.863.2.2 uint8_t unpack_tmd_SLQSTmdGetMitigationDevList_t::MitigationDevListLen

#### 8.863.2.3 uint16_t unpack_tmd_SLQSTmdGetMitigationDevList_t::Tlvresult

## 8.864 unpack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference

### Data Fields

- uint8_t CurrentmitigationLvl
- uint8_t ReqMitigationLvl
- uint16_t Tlvresult

### 8.864.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *Currentmitigation-Lvl* | • Current thermal mitigation level (Optional) |
| *ReqMitigationLvl* | • Requested Thermal Mitigation Level (Optional)<br>• The requested thermal mitigation level from the client. The default is zero if the client has not previously set the mitigation level. |
| *Tlvresult* | • Unpack Result |

### 8.864.2 Field Documentation

#### 8.864.2.1 uint8_t unpack_tmd_SLQSTmdGetMitigationLvl_t::CurrentmitigationLvl

#### 8.864.2.2 uint8_t unpack_tmd_SLQSTmdGetMitigationLvl_t::ReqMitigationLvl

#### 8.864.2.3 uint16_t unpack_tmd_SLQSTmdGetMitigationLvl_t::Tlvresult

## 8.865 unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t Struct Reference

### Data Fields

- uint8_t deviceIdLen
- char deviceID [255]
- uint8_t lvl

### 8.865.1 Detailed Description

Mitigation Level Report

**Parameters**

| | |
|---|---|
| *deviceIdLen* | • Mitigation Device ID Length |
| *deviceID* | • Mitigation Device ID |
| *lvl* | • Current thermal mitigation level |

### 8.865.2 Field Documentation

#### 8.865.2.1 char unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::deviceID[255]

#### 8.865.2.2 uint8_t unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::deviceIdLen

#### 8.865.2.3 uint8_t unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t::lvl

## 8.866 unpack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.866.1 Detailed Description

This structure contains mitigation devices Level request parameters

**Parameters**

| | |
|---|---|
| *Tlvresult* | • Unpack Result |

### 8.866.2 Field Documentation

#### 8.866.2.1 uint16_t unpack_tmd_SLQSTmdRegNotMitigationLvl_t::Tlvresult

## 8.867 unpack_uim_ChangePin_t Struct Reference

### Data Fields

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

### 8.867.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See uim_remainingRetries for more information.<br>• NULL pointer - Invalid data. |
| *pEncryptedPI-N1(optional)* | • See uim_encryptedPIN1 for more information.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result is provided in a subsequent indication.<br>• 0xFFFFFFFF, if unavailable<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.867.2 Field Documentation

#### 8.867.2.1 uim_encryptedPIN1∗ unpack_uim_ChangePin_t::pEncryptedPIN1

#### 8.867.2.2 uint32_t∗ unpack_uim_ChangePin_t::pIndicationToken

#### 8.867.2.3 uim_remainingRetries∗ unpack_uim_ChangePin_t::pRemainingRetries

#### 8.867.2.4 uint16_t unpack_uim_ChangePin_t::Tlvresult

## 8.868 unpack_uim_GetCardStatus_t Struct Reference

### Data Fields

- uim_cardStatus ∗ pCardStatus
- uim_hotSwapStatus ∗ pHotSwapStatus
- uint16_t Tlvresult

### 8.868.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pCard-Status(optional)* | • See uim_cardStatus for more information.<br>• NULL pointer - Invalid data. |
| *pHotSwap-Status(optional)* | • See uim_hotSwapStatus for more information.<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.868.2 Field Documentation

#### 8.868.2.1 uim_cardStatus∗ unpack_uim_GetCardStatus_t::pCardStatus

#### 8.868.2.2 uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus

#### 8.868.2.3 uint16_t unpack_uim_GetCardStatus_t::Tlvresult

## 8.869 unpack_uim_ReadTransparent_t Struct Reference

### Data Fields

- uim_cardResult * pCardResult
- uim_readResult * pReadResult
- uint32_t * pIndicationToken
- uint8_t * pEncryptedData
- uint16_t Tlvresult

### 8.869.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pCardResult* | • See uim_cardResult for more information.<br>• NULL pointer - Invalid data. |
| *pReadResult* | • See uim_readResult for more information.<br>• NULL pointer - Invalid data. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• NULL pointer - Invalid data. |
| *pEncrypted-Data(optional)* | • Encrypted Data.<br>• Indicates whether the data from the card passed in read_result is encrypted.<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.869.2 Field Documentation

#### 8.869.2.1 uim_cardResult* unpack_uim_ReadTransparent_t::pCardResult

#### 8.869.2.2 uint8_t* unpack_uim_ReadTransparent_t::pEncryptedData

#### 8.869.2.3 uint32_t* unpack_uim_ReadTransparent_t::pIndicationToken

#### 8.869.2.4 uim_readResult* unpack_uim_ReadTransparent_t::pReadResult

#### 8.869.2.5 uint16_t unpack_uim_ReadTransparent_t::Tlvresult

## 8.870 unpack_uim_SetPinProtection_t Struct Reference

### Data Fields

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

### 8.870.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See uim_remainingRetries for more information.<br>• NULL pointer - Invalid data. |
| *pEncryptedPI-N1(optional)* | • See uim_encryptedPIN1 for more information.<br>• NULL pointer - Invalid data. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result is provided in a subsequent indication.<br>• 0xFFFFFFFF, if unavailable<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.870.2 Field Documentation

#### 8.870.2.1 uim_encryptedPIN1* unpack_uim_SetPinProtection_t::pEncryptedPIN1

#### 8.870.2.2 uint32_t* unpack_uim_SetPinProtection_t::pIndicationToken

#### 8.870.2.3 uim_remainingRetries* unpack_uim_SetPinProtection_t::pRemainingRetries

#### 8.870.2.4 uint16_t unpack_uim_SetPinProtection_t::Tlvresult

## 8.871 unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference

### Data Fields

- slots_t slotsstatusChange
- uint8_t bNumberOfPhySlots

### 8.871.1 Detailed Description

Structure consist of card status params

**Parameters**

| | |
|---|---|
| *slotsstatus-Change* | • See slot_t for more information |
| *bNumberOfPhy-Slots* | • Number of Physical Slot(s)<br>• 0xff - Invalid card slot status. |

### 8.871.2 Field Documentation

#### 8.871.2.1 uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots

#### 8.871.2.2 slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange

## 8.872 unpack_uim_SLQSUIMAuthenticate_t Struct Reference

### Data Fields

- uim_cardResult * pCardResult
- uim_authenticateResult * pAuthenticateResult
- uint32_t * pIndicationToken

### 8.872.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMAuthenticate.

**Parameters**

| | |
|---|---|
| *pCard-Result(optional)* | • See uim_cardResult for more information.<br>• NULL pointer - Invalid data. |
| *pAuthenticate-Result(optional)* | • See uim_authenticateResult for more information.<br>• NULL pointer - Invalid data. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.872.2 Field Documentation

#### 8.872.2.1 uim_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult

#### 8.872.2.2 uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult

#### 8.872.2.3 uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken

## 8.873 unpack_uim_SLQSUIMDepersonalization_t Struct Reference

### Data Fields

- uim_remainingRetries * pRemainingRetries

### 8.873.1 Detailed Description

This structure contains information of the response parameters associated with a Depersonalization.

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See uim_remainingRetries for more information.<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.873.2 Field Documentation

#### 8.873.2.1 uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries

## 8.874 unpack_uim_SLQSUIMEventRegister_t Struct Reference

### Data Fields

- uint32_t eventMask

### 8.874.1 Detailed Description

This structure contains unpack event register parameter.

**Parameters**

| | |
|---|---|
| *eventMask* | - bit 0 - card status<br>• bit 1 - SAP connection<br>• bit 4 - physical slot status<br>• 0xffffffff - Invalid data. |

### 8.874.2 Field Documentation

#### 8.874.2.1 uint32_t unpack_uim_SLQSUIMEventRegister_t::eventMask

## 8.875 unpack_uim_SLQSUIMGetConfiguration_t Struct Reference

### Data Fields

- uint8_t * pAutoSelection
- uim_personalizationStatus * pPersonalizationStatus
- uint8_t * pHaltSubscription

### 8.875.1 Detailed Description

This structure contains information of the response parameters associated with Get Configuration.

**Parameters**

| | |
|---|---|
| *pAutoSelection(optional)* | • Indicates whether the modem is configured to automatically select the provisioning sessions at powerup.<br>• Valid values<br>– 0 - Automatic provisioning is off<br>– 1 - Automatic provisioning is on<br>• NULL pointer - Invalid data. |
| *pPersonalizationStatus(optional)* | • See uim_personalizationStatus for more information.<br>• NULL pointer - Invalid data. |
| *pHaltSubscription(optional)* | • Indicates if the modem is configured to publish the subscription after successful initialization.<br>• Valid values<br>– 0 - Modem proceeds with publishing the subscription<br>– 1 - Modem does not publish the subscription<br>• NULL pointer - Invalid data. |

### 8.875.2 Field Documentation

#### 8.875.2.1 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection

#### 8.875.2.2 uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription

#### 8.875.2.3 uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus

## 8.876 unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference

### Data Fields

- uim_cardResult * pCardResult
- uim_fileAttributes * pFileAttributes
- uint32_t * pIndicationToken

### 8.876.1 Detailed Description

This structure contains information of the response parameters associated with a Get File Attributes

**Parameters**

| | |
|---|---|
| *pCard-Result(optional)* | • See uim_cardResult for more information.<br>• NULL pointer - Invalid data. |
| *pFile-Attributes(optional)* | • See uim_fileAttributes for more information.<br>• NULL pointer - Invalid data. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result must be provided in a subsequent indication.<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.876.2 Field Documentation

#### 8.876.2.1 uim_cardResult∗ unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult

#### 8.876.2.2 uim_fileAttributes∗ unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes

#### 8.876.2.3 uint32_t∗ unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken

## 8.877 unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference

### Data Fields

- uint8_t ∗ pNumberOfPhySlot
- slots_t ∗ pUimSlotsStatus

### 8.877.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pNumberOfPhy-Slot* | • Number of sets of the Slot Status.<br>• NULL pointer - Invalid data. |
| *pUimSlotsStatus* | • Slots Status See slots_t for more information.. |

### 8.877.2 Field Documentation

#### 8.877.2.1 uint8_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pNumberOfPhySlot

#### 8.877.2.2 slots_t* unpack_uim_SLQSUIMGetSlotsStatus_t::pUimSlotsStatus

## 8.878 unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference

### Data Fields

- uint8_t TlvPresent
- uim_refreshevent refreshEvent

### 8.878.1 Detailed Description

This structure hold parameters about UIM refresh event indication.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if refresh event TLV is present in indication, otherwise 0 |
| *refreshEvent[OPTIONAL]* | • see uim_refreshevent |

### 8.878.2 Field Documentation

#### 8.878.2.1 uim_refreshevent unpack_uim_SLQSUIMRefreshCallback_Ind_t::refreshEvent

#### 8.878.2.2 uint8_t unpack_uim_SLQSUIMRefreshCallback_Ind_t::TlvPresent

## 8.879 unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference

### Data Fields

- uim_refreshevent * pRefreshEvent

### 8.879.1 Detailed Description

This structure contains information of the response parameters associated with a Refresh Get Last Event.

**Parameters**

| | |
|---|---|
| *refreshEvent(Optional)* | • See uim_refreshevent for more information.<br>• NULL pointer - Invalid data. |

### 8.879.2 Field Documentation

#### 8.879.2.1 uim_refreshevent* unpack_uim_SLQSUIMRefreshGetLastEvent_t::pRefreshEvent

## 8.880 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference

### Data Fields

- uim_cardStatus * pCardStatus

### 8.880.1 Detailed Description

This structure contains information about Status change callback.

**Parameters**

| | |
|---|---|
| *pCardStatus* | Card Status<br>• See uim_cardStatus for more information.<br>• NULL pointer - Invalid data. |

### 8.880.2 Field Documentation

#### 8.880.2.1 uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus

## 8.881 unpack_uim_UnblockPin_t Struct Reference

### Data Fields

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

### 8.881.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See uim_remainingRetries for more information.<br>• NULL pointer - Invalid data. |
| *pEncryptedPI-N1(optional)* | • See uim_encryptedPIN1 for more information.<br>• NULL pointer - Invalid data. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result is provided in a subsequent indication.<br>• 0xFFFFFFFF, if unavailable<br>• NULL pointer - Invalid data. |

### 8.881.2 Field Documentation

#### 8.881.2.1 uim_encryptedPIN1∗ unpack_uim_UnblockPin_t::pEncryptedPIN1

#### 8.881.2.2 uint32_t∗ unpack_uim_UnblockPin_t::pIndicationToken

#### 8.881.2.3 uim_remainingRetries∗ unpack_uim_UnblockPin_t::pRemainingRetries

#### 8.881.2.4 uint16_t unpack_uim_UnblockPin_t::Tlvresult

## 8.882 unpack_uim_VerifyPin_t Struct Reference

### Data Fields

- uim_remainingRetries ∗ pRemainingRetries
- uim_encryptedPIN1 ∗ pEncryptedPIN1
- uint32_t ∗ pIndicationToken
- uint16_t Tlvresult

### 8.882.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *pRemaining-Retries(optional)* | • See uim_remainingRetries for more information.<br>• NULL pointer - Invalid data. |
| *pEncryptedPI-N1(optional)* | • See uim_encryptedPIN1 for more information.<br>• NULL pointer - Invalid data. |
| *pIndication-Token(optional)* | • Response in Indication.<br>• When this TLV is present, it indicates that the result is provided in a subsequent indication.<br>• 0xFFFFFFFF, if unavailable<br>• NULL pointer - Invalid data. |

**Note**

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

### 8.882.2 Field Documentation

#### 8.882.2.1 uim_encryptedPIN1∗ unpack_uim_VerifyPin_t::pEncryptedPIN1

#### 8.882.2.2 uint32_t∗ unpack_uim_VerifyPin_t::pIndicationToken

#### 8.882.2.3 uim_remainingRetries∗ unpack_uim_VerifyPin_t::pRemainingRetries

#### 8.882.2.4 uint16_t unpack_uim_VerifyPin_t::Tlvresult

## 8.883 unpack_voice_allCallStatusCallback_ind_t Struct Reference

### Data Fields

- voice_arrCallInfo arrCallInfomation
- voice_arrRemotePartyNum * pArrRemotePartyNum
- voice_arrRemotePartyName * pArrRemotePartyName
- voice_arrAlertingType * pArrAlertingType
- voice_arrSvcOption * pArrSvcOption
- voice_arrCallEndReason * pArrCallEndReason
- voice_arrAlphaID * pArrAlphaID
- voice_arrConnectPartyNum * pArrConnectPartyNum
- voice_arrDiagInfo * pArrDiagInfo
- voice_arrCalledPartyNum * pArrCalledPartyNum
- voice_arrRedirPartyNum * pArrRedirPartyNum
- voice_arrAlertingPattern * pArrAlertingPattern

### 8.883.1 Detailed Description

This structure contains information about the indication parameters for all the calls originating or terminating from a particular device.

**Parameters**

| | |
|---|---|
| *arrCallInfo(mandatory)* | • See voice_arrCallInfo for more information.<br>• NULL pointer - Invalid data. |
| *pArrRemotePartyNum(optional)* | • See voice_arrRemotePartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pArrRemotePartyName(optional)* | • See voice_arrRemotePartyName for more information.<br>• NULL pointer - Invalid data. |
| *pArrAlertingType(optional)* | • See voice_arrAlertingType for more information.<br>• NULL pointer - Invalid data. |
| *pArrUUSInfo(optional)* | • See voice_arrUUSInfo for more information.<br>• NULL pointer - Invalid data. |
| *pArrSvcOption(optional)* | • See voice_arrSvcOption for more information.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pOTASP-Status(optional)* | • OTASP status for the OTASP call.<br>• Applicable only for 3GPP2 devices.<br>– 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP<br>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP<br>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP<br>– 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)<br>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP<br>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP<br>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP<br>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP<br>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP<br>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA)<br>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA)<br>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA)<br>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA)<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pVoice-Privacy(optional)* | • Voice Privacy.<br>• Values.<br>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy<br>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pArrCallEnd-Reason(optional)* | • See voice_arrCallEndReason for more information.<br>• NULL pointer - Invalid data. |
| *pArrAlphaI-D(optional)* | • See voice_arrAlphaID for more information.<br>• NULL pointer - Invalid data. |
| *pArrConnect-Party-Num(optional)* | • See voice_arrConnectPartyNum for more information.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pArrDiag-Info(optional)* | • See voice_arrDiagInfo for more information.<br>• NULL pointer - Invalid data. |
| *pArrCalledParty-Num(optional)* | • See voice_arrCalledPartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pArrRedirParty-Num(optional)* | • See voice_arrRedirPartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pArrAlerting-Pattern(optional)* | • See voice_arrAlertingPattern for more information.<br>• NULL pointer - Invalid data. |

### 8.883.2 Field Documentation

#### 8.883.2.1 voice_arrCallInfo unpack_voice_allCallStatusCallback_ind_t::arrCallInfomation

#### 8.883.2.2 voice_arrAlertingPattern∗ unpack_voice_allCallStatusCallback_ind_t::pArrAlertingPattern

#### 8.883.2.3 voice_arrAlertingType∗ unpack_voice_allCallStatusCallback_ind_t::pArrAlertingType

#### 8.883.2.4 voice_arrAlphaID∗ unpack_voice_allCallStatusCallback_ind_t::pArrAlphaID

#### 8.883.2.5 voice_arrCalledPartyNum∗ unpack_voice_allCallStatusCallback_ind_t::pArrCalledPartyNum

#### 8.883.2.6 voice_arrCallEndReason∗ unpack_voice_allCallStatusCallback_ind_t::pArrCallEndReason

#### 8.883.2.7 voice_arrConnectPartyNum∗ unpack_voice_allCallStatusCallback_ind_t::pArrConnectPartyNum

#### 8.883.2.8 voice_arrDiagInfo∗ unpack_voice_allCallStatusCallback_ind_t::pArrDiagInfo

#### 8.883.2.9 voice_arrRedirPartyNum∗ unpack_voice_allCallStatusCallback_ind_t::pArrRedirPartyNum

#### 8.883.2.10 voice_arrRemotePartyName∗ unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyName

#### 8.883.2.11 voice_arrRemotePartyNum∗ unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyNum

#### 8.883.2.12 voice_arrSvcOption∗ unpack_voice_allCallStatusCallback_ind_t::pArrSvcOption

## 8.884 unpack_voice_DTMFEventCallback_ind_t Struct Reference

### Data Fields

- voice_DTMFInfo DTMFInformation
- uint8_t ∗ pOnLength
- uint8_t ∗ pOffLength

### 8.884.1 Detailed Description

This structure contains the parameters passed for DTMF event indication by the device.

**Parameters**

| | |
|---|---|
| *DTMF-Information(mandatory)* | See voice_DTMFInfo for more information. |
| *pOn-Length(optional)* | • DTMF Pulse Width<br>– 0x00 - DTMF_ONLENGTH_95MS - 95 ms<br>– 0x01 - DTMF_ONLENGTH_150MS - 150 ms<br>– 0x02 - DTMF_ONLENGTH_200MS - 200 ms<br>– 0x03 - DTMF_ONLENGTH_250MS - 250 ms<br>– 0x04 - DTMF_ONLENGTH_300MS - 300 ms<br>– 0x05 - DTMF_ONLENGTH_350MS - 350 ms<br>– 0x06 - DTMF_ONLENGTH_SMS - SMS Tx special pulse width |
| *pOff-Length(optional)* | • DTMF Interdigit Interval<br>– 0x00 - DTMF_OFFLENGTH_60MS - 60 ms<br>– 0x01 - DTMF_OFFLENGTH_100MS - 100 ms<br>– 0x02 - DTMF_OFFLENGTH_150MS - 150 ms<br>– 0x03 - DTMF_OFFLENGTH_200MS - 200 ms |

**Note**

None

### 8.884.2 Field Documentation

#### 8.884.2.1 voice_DTMFInfo unpack_voice_DTMFEventCallback_ind_t::DTMFInformation

#### 8.884.2.2 uint8_t* unpack_voice_DTMFEventCallback_ind_t::pOffLength

#### 8.884.2.3 uint8_t* unpack_voice_DTMFEventCallback_ind_t::pOnLength

## 8.885 unpack_voice_OTASPStatusCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID
- uint8_t OTASPStatus

### 8.885.1 Detailed Description

This structure consist of OTASP or OTAPA event params

**Parameters**

| | |
|---|---|
| *callID* | • Call identifier for the call. |
| *OTASPStatus* | • OTASP status for the OTASP call. Values:<br>– 0x00 - OTASP_STATUS_SPL_UNLOCKED.SPL unlocked; only for user-initiated OTASP<br>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED. SPC retries exceeded; only for user-initiated OTASP<br>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED.A-key exchanged; only for user-initiated O-TASP<br>– 0x03 - OTASP_STATUS_SSD_UPDATED. SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)<br>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP<br>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP<br>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP<br>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP<br>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP<br>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP(OTAPA)<br>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP(OTAPA)<br>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP(OTAPA)<br>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP(OTAPA) |

### 8.885.2 Field Documentation

#### 8.885.2.1 uint8_t unpack_voice_OTASPStatusCallback_ind_t::callID

#### 8.885.2.2 uint8_t unpack_voice_OTASPStatusCallback_ind_t::OTASPStatus

## 8.886 unpack_voice_SLQSOriginateUSSD_t Struct Reference

### Data Fields

- uint16_t * pfailureCause
- voice_alphaIDInfo * pAlphaIDInfo
- struct voice_USSInfo * pUSSDInfo
- uint8_t * pCcResultType
- uint8_t * pCallId
- voice_ccSUPSType * pCCSuppsType

### 8.886.1 Detailed Description

This structure contains the parameters of USS response.

**Parameters**

*pfailureCause*

- Failure Cause.
- 0 - QMI_FAILURE_CAUSE_OFFLINE - Phone is offline
- 20 - QMI_FAILURE_CAUSE_CDMA_LOCK - Phone is CDMA locked until a power cycle; CDMA only
- 21 - QMI_FAILURE_CAUSE_NO_SRV - Phone has no service
- 22 - QMI_FAILURE_CAUSE_FADE - Call has ended abnormally
- 23 - QMI_FAILURE_CAUSE_INTERCEPT - Received intercept from the base station; originating only; CDMA only
- 24 - QMI_FAILURE_CAUSE_REORDER - Received reorder from the base station; originating only; CDMA only
- 25 - QMI_FAILURE_CAUSE_REL_NORMAL - Received release from the base station; no reason was given
- 26 - QMI_FAILURE_CAUSE_REL_SO_REJ - Received release from the base station; SO reject; CDMA only
- 27 - QMI_FAILURE_CAUSE_INCOM_CALL - Received incoming call from the base station
- 28 - QMI_FAILURE_CAUSE_ALERT_STOP - Received alert stop from the base station; incoming only; CDMA only
- 29 - QMI_FAILURE_CAUSE_CLIENT_END - Client ended the call
- 30 - QMI_FAILURE_CAUSE_ACTIVATION - Received end activation; OTASP call only; CDMA only
- 31 - QMI_FAILURE_CAUSE_MC_ABORT - MC aborted the origination/conversation; CDMA only
- 32 - QMI_FAILURE_CAUSE_MAX_ACCESS_PROBE - Maximum access probes were transmitted; CDMA only
- 33 - QMI_FAILURE_CAUSE_PSIST_N - Persistence test failure; FEATURE_JCDMA only; CDMA only
- 34 - QMI_FAILURE_CAUSE_UIM_NOT_PRESENT - R-UIM is not present
- 35 - QMI_FAILURE_CAUSE_ACC_IN_PROG - Access attempt is already in progress
- 36 - QMI_FAILURE_CAUSE_ACC_FAIL - Access failure for a reason other than the above
- 37 - QMI_FAILURE_CAUSE_RETRY_ORDER - Received retry order; originating only; IS 2000; CDMA only
- 38 - QMI_FAILURE_CAUSE_CCS_NOT_SUPPORTED_BYBS - Concurrent service is not supported by the base station
- 39 - QMI_FAILURE_CAUSE_NO_RESPONSE_FROM_BS - No response was received from the base station
- 40 - QMI_FAILURE_CAUSE_REJECTED_BY_BS - Call was rejected by the base station; CDMA only
- 41 - QMI_FAILURE_CAUSE_INCOMPATIBLE - Concurrent services requested were not compatible; CDMA only
- 42 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station; CDMA only
- 43 - QMI_FAILURE_CAUSE_ALREADY_IN_TC - Corresponds to CM_CALL_ORIGERR_ALREADY_IN_TC
- 44 - QMI_FAILURE_CAUSE_EMERGENCY_FLASHED - Call is ended because an emergency call was flashed over this call; CDMA only
- 45 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_GPS - Used if CM is ending a GPS call in preference of a user call
- 46 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_SMS - Used if CM is ending an SMS call in preference of a user call
- 47 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_DATA - Used if CM is ending a data call in preference of an emergency call
- 48 - QMI_FAILURE_CAUSE_REDIR_OR_HANDOFF - Call was rejected because of a redirection or handoff
- 49 - QMI_FAILURE_CAUSE_ACCESS_BLOCK_ALL - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only
- 50 - QMI_FAILURE_CAUSE_OTASP_SPC_ERR - To support OTASP SPC Error indication
- 51 - QMI_FAILURE_CAUSE_IS707B_MAX_ACC - Maximum access probes for an IS-707B

| | |
|---|---|
| *AlphaIdentifier* | • see voice_alphaIDInfo definition<br>• NULL pointer - Invalid data. |
| *pUSSDInfo* | • USS Data from Network (See structure voice_USSInfo) |

- NULL pointer - Invalid data.

**Parameters**

| | |
|---|---|
| *pCcResultType* | • CC result code<br>• NULL pointer - Invalid data. |
| *pCallId* | • Caller ID information<br>• NULL pointer - Invalid data. |
| *pCCSuppsType* | • See structure 'voice_ccSUPSType' definition<br>• NULL pointer - Invalid data. |

### 8.886.2 Field Documentation

#### 8.886.2.1 voice_alphaIDInfo* unpack_voice_SLQSOriginateUSSD_t::pAlphaIDInfo

#### 8.886.2.2 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCallId

#### 8.886.2.3 uint8_t* unpack_voice_SLQSOriginateUSSD_t::pCcResultType

#### 8.886.2.4 voice_ccSUPSType* unpack_voice_SLQSOriginateUSSD_t::pCCSuppsType

#### 8.886.2.5 uint16_t* unpack_voice_SLQSOriginateUSSD_t::pfailureCause

#### 8.886.2.6 struct voice_USSInfo* unpack_voice_SLQSOriginateUSSD_t::pUSSDInfo

## 8.887 unpack_voice_SLQSVoiceAnswerCall_t Struct Reference

### Data Fields

- uint8_t * pCallId

### 8.887.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

**Parameters**

| | |
|---|---|
| *pCallId* | • Unique call identifier for the call that must be answered.<br>• NULL pointer - Invalid data. |

### 8.887.2 Field Documentation

#### 8.887.2.1 uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId

## 8.888 unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference

### Data Fields

- uint8_t * pCallID

### 8.888.1 Detailed Description

This structure contains Voice Burst DTMF Information for unpack.

**Parameters**

| | |
|---|---|
| *pCallID* | • Call ID associated with call on which the DTMF information has to be sent.<br>A burst DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.<br>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.<br>• If the call ID value received is 0, no value has been returned by the device<br>• NULL pointer - Invalid data. |

### 8.888.2 Field Documentation

#### 8.888.2.1 uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID

## 8.889 unpack_voice_SLQSVoiceDialCall_t Struct Reference

### Data Fields

- uint8_t * pCallID
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- voice_ccSUPSType * pCCSUPSType

### 8.889.1 Detailed Description

This structure contains Voice Call Response Parameters

**Parameters**

| | |
|---|---|
| *pCallID(optional)* | • Unique call identifier for the dialed call<br>• NULL pointer - Invalid data. |
| *pAlphaID-Info(optional)* | • Pointer to structure of alphaIDInfo<br>– See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResult-Type(optional)* | • Call Control Result Type.<br>– 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>– NULL pointer - Invalid data. |
| *pCCSUPS-Type(optional)* | • Pointer to structure of ccSUPSType<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

### 8.889.2 Field Documentation

#### 8.889.2.1 voice_alphaIDInfo∗ unpack_voice_SLQSVoiceDialCall_t::pAlphaIDInfo

#### 8.889.2.2 uint8_t∗ unpack_voice_SLQSVoiceDialCall_t::pCallID

#### 8.889.2.3 uint8_t∗ unpack_voice_SLQSVoiceDialCall_t::pCCResultType

#### 8.889.2.4 voice_ccSUPSType∗ unpack_voice_SLQSVoiceDialCall_t::pCCSUPSType

## 8.890 unpack_voice_SLQSVoiceEndCall_t Struct Reference

### Data Fields

- uint8_t ∗ pCallId

### 8.890.1 Detailed Description

This structure contains unpack voice end call parameter.

**Parameters**

| | |
|---|---|
| *pCallId* | • Unique call identifier for the call that must be ended<br>• NULL pointer - Invalid data. |

### 8.890.2 Field Documentation

#### 8.890.2.1 uint8_t* unpack_voice_SLQSVoiceEndCall_t::pCallId

## 8.891 unpack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference

### Data Fields

- voice_arrCallInfo * pArrCallInfo
- voice_arrRemotePartyNum * pArrRemotePartyNum
- voice_arrRemotePartyName * pArrRemotePartyName
- voice_arrAlertingType * pArrAlertingType
- voice_arrUUSInfo * pArrUUSInfo
- voice_arrSvcOption * pArrSvcOption
- uint8_t * pOTASPStatus
- uint8_t * pVoicePrivacy
- voice_arrCallEndReason * pArrCallEndReason
- voice_arrAlphaID * pArrAlphaID
- voice_arrConnectPartyNum * pArrConnectPartyNum
- voice_arrDiagInfo * pArrDiagInfo
- voice_arrCalledPartyNum * pArrCalledPartyNum
- voice_arrRedirPartyNum * pArrRedirPartyNum
- voice_arrAlertingPattern * pArrAlertingPattern

### 8.891.1 Detailed Description

This structure contains information about the response parameters with all the calls originating or terminating from a particular device.

**Parameters**

| | |
|---|---|
| *pArrCallInfo(optional)* | • See voice_arrCallInfo for more information.<br>• NULL pointer - Invalid data. |
| *pArrRemotePartyNum(optional)* | • See voice_arrRemotePartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pArrRemotePartyName(optional)* | • See voice_arrRemotePartyName for more information.<br>• NULL pointer - Invalid data. |
| *pArrAlertingType(optional)* | • See voice_arrAlertingType for more information.<br>• NULL pointer - Invalid data. |
| *pArrUUSInfo(optional)* | • See voice_arrUUSInfo for more information.<br>• NULL pointer - Invalid data. |
| *pArrSvcOption(optional)* | • See voice_arrSvcOption for more information.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pOTASP-Status(optional)* | • OTASP status for the OTASP call.<br>• Applicable only for 3GPP2 devices.<br>– 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP<br>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP<br>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP<br>– 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)<br>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP<br>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP<br>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP<br>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP<br>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP<br>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA)<br>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA)<br>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA)<br>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA)<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pVoice-Privacy(optional)* | • Voice Privacy.<br>• Values.<br>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy<br>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pArrCallEnd-Reason(optional)* | • See voice_arrCallEndReason for more information.<br>• NULL pointer - Invalid data. |
| *pArrAlphaI-D(optional)* | • See voice_arrAlphaID for more information.<br>• NULL pointer - Invalid data. |
| *pArrConnect-Party-Num(optional)* | • See voice_arrConnectPartyNum for more information.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pArrDiag-Info(optional)* | • See voice_arrDiagInfo for more information.<br>• NULL pointer - Invalid data. |
| *pArrCalledParty-Num(optional)* | • See voice_arrCalledPartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pArrRedirParty-Num(optional)* | • See voice_arrRedirPartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pArrAlerting-Pattern(optional)* | • See voice_arrAlertingPattern for more information.<br>• NULL pointer - Invalid data. |

### 8.891.2 Field Documentation

#### 8.891.2.1 voice_arrAlertingPattern∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern

#### 8.891.2.2 voice_arrAlertingType∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType

#### 8.891.2.3 voice_arrAlphaID∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID

#### 8.891.2.4 voice_arrCalledPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum

#### 8.891.2.5 voice_arrCallEndReason∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason

#### 8.891.2.6 voice_arrCallInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo

#### 8.891.2.7 voice_arrConnectPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum

#### 8.891.2.8 voice_arrDiagInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo

#### 8.891.2.9 voice_arrRedirPartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum

#### 8.891.2.10 voice_arrRemotePartyName∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName

#### 8.891.2.11 voice_arrRemotePartyNum∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum

#### 8.891.2.12 voice_arrSvcOption∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption

#### 8.891.2.13 voice_arrUUSInfo∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo

#### 8.891.2.14 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus

#### 8.891.2.15 uint8_t∗ unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy

## 8.892 unpack_voice_SLQSVoiceGetCallBarring_t Struct Reference

### Data Fields

- uint8_t * pSvcClass
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.892.1 Detailed Description

This structure contains Voice Get Call Barring Response Parameters

**Parameters**

| | |
|---|---|
| *pSvcClass* | • Service class is a combination (sum) of information class constants (optional)<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes.<br>• Service Class is set to 0 if call waiting is not active for any of the information classes.<br>• 0xFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>  – 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>  – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>  – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>  – See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.892.2 Field Documentation

#### 8.892.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallBarring_t::pAlphaIDInfo

#### 8.892.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCallID

#### 8.892.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType

#### 8.892.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType

#### 8.892.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause

#### 8.892.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass

## 8.893 unpack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference

### Data Fields

- voice_getCallFWInfo * pGetCallFWInfo
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType
- voice_getCallFWExtInfo * pGetCallFWExtInfo

### 8.893.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Unpack Parameters

**Parameters**

| | |
|---|---|
| *pGetCallFWInfo* | • Pointer to structure of getCallFWInfo (optional)<br>  – See voice_getCallFWInfo for more information<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pCCResType* | • Call Control Result Type (optional)<br>– 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |
| *pGetCallFWExt-Info* | • Pointer to structure of getCallFWExtInfo (optional)<br>– See voice_getCallFWExtInfo for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.893.2 Field Documentation

#### 8.893.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo

#### 8.893.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID

#### 8.893.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType

#### 8.893.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType

#### 8.893.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pFailCause

#### 8.893.2.6 voice_getCallFWExtInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWExtInfo

#### 8.893.2.7 voice_getCallFWInfo* unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pGetCallFWInfo

## 8.894 unpack_voice_SLQSVoiceGetCallInfo_t Struct Reference

### Data Fields

- voice_callInfo * pCallInfo
- voice_remotePartyNum * pRemotePartyNum

- uint16_t * pSrvOpt
- uint8_t * pVoicePrivacy
- uint8_t * pOTASPStatus
- voice_remotePartyName * pRemotePartyName
- voice_UUSInfo * pUUSInfo
- uint8_t * pAlertType
- voice_alphaIDInfo * pAlphaIDInfo
- voice_connectNumInfo * pConnectNumInfo
- voice_diagInfo * pDiagInfo
- uint32_t * pAlertingPattern

### 8.894.1 Detailed Description

This structure contains information of the unpack parameters associated with a call.

**Parameters**

| | |
|---|---|
| *pCallInfo(optional)* | • See voice_callInfo for more information.<br>• NULL pointer - Invalid data. |
| *pRemotePartyNum(optional)* | • See voice_remotePartyNum for more information.<br>• NULL pointer - Invalid data. |
| *pSrvOpt* | • Service option(optional)<br>• Applicable only for 3GPP2 devices.<br>• See Table8 qaGobiApiTableServiceOptions.h for standard service option number assignments.<br>• NULL pointer - Invalid data. |
| *pVoicePrivacy* | • Voice Privacy.(optional)<br>• Applicable only for 3GPP2 devices.<br>• Values.<br>  – 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy<br>  – 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pOTASPStatus* | • OTASP status for the OTASP call.(optional)<br>• Applicable only for 3GPP2 devices.<br>– 0x00 - OTASP_STATUS_SPL_UNLOCKED - SPL unlocked; only for user-initiated OTASP<br>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED - SPC retries exceeded; only for user-initiated OTASP<br>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED - A-key exchanged; only for user-initiated OTASP<br>– 0x03 - OTASP_STATUS_SSD_UPDATED - SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)<br>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP<br>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP<br>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP<br>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP<br>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP<br>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP (OTAPA)<br>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP (OTAPA)<br>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP (OTAPA)<br>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP (OTAPA)<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pRemoteParty-Name(optional)* | • Applicable only for 3GPP devices.<br>• See voice_remotePartyName for more information.<br>• NULL pointer - Invalid data. |
| *pUUS-Info(optional)* | • Applicable only for 3GPP devices.<br>• See voice_UUSInfo for more information.<br>• NULL pointer - Invalid data. |
| *pAlert-Type(optional)* | • Alerting type.<br>• Applicable only for 3GPP devices.<br>– 0x00 - ALERTING_LOCAL - Local<br>– 0x01 - ALERTING_REMOTE - Remote<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pAlphaID-Info(optional)* | • Applicable only for 3GPP devices.<br>• See voice_alphaIDInfo for more information.<br>• NULL pointer - Invalid data. |
| *pConnectNum-Info(optional)* | • See voice_connectNumInfo for more information.<br>• NULL pointer - Invalid data. |
| *pDiag-Info(optional)* | • See voice_diagInfo for more information.<br>• NULL pointer - Invalid data. |
| *pAlertingPattern* | • Alerting pattern.(optional)<br>– 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1<br>– 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2<br>– 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3<br>– 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5<br>– 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6<br>– 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7<br>– 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8<br>– 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |

## 8.894.2 Field Documentation

### 8.894.2.1 uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern

### 8.894.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType

### 8.894.2.3 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo

### 8.894.2.4 voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo

### 8.894.2.5 voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo

### 8.894.2.6 voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo

### 8.894.2.7 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus

### 8.894.2.8 voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName

### 8.894.2.9 voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum

### 8.894.2.10 uint16_t* unpack_voice_SLQSVoiceGetCallInfo_t::pSrvOpt

### 8.894.2.11 voice_UUSInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pUUSInfo

### 8.894.2.12 uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pVoicePrivacy

## 8.895 unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference

### Data Fields

- uint8_t * pSvcClass
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.895.1 Detailed Description

This structure contains Voice Get Call Waiting Unpack Parameters

**Parameters**

| | |
|---|---|
| *pSvcClass* | • Service class is a combination (sum) of information class constants (optional)<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes.<br>• Service Class is set to 0 if call waiting is not active for any of the information classes.<br>• 0xFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>  – 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>  – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>  – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.895.2 Field Documentation

#### 8.895.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallWaiting_t::pAlphaIDInfo

#### 8.895.2.2 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCallID

#### 8.895.2.3 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCResType

#### 8.895.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCallWaiting_t::pCCSUPSType

#### 8.895.2.5 uint16_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pFailCause

#### 8.895.2.6 uint8_t* unpack_voice_SLQSVoiceGetCallWaiting_t::pSvcClass

## 8.896 unpack_voice_SLQSVoiceGetCLIP_t Struct Reference

### Data Fields

- voice_CLIPResp * pCLIPResp
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.896.1 Detailed Description

This structure contains Voice Get Calling Line Identification Presentation(CLIP) Response Parameters

**Parameters**

| | |
|---|---|
| *pCLIPResp* | • Pointer to structure of CLIPResp (optional)<br>– See voice_CLIPResp for more information<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See voice_alphaIDInfo for more information |
| *pCCResType* | • Call Control Result Type (optional)<br>– 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.896.2 Field Documentation

#### 8.896.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIP_t::pAlphaIDInfo

#### 8.896.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCallID

#### 8.896.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIP_t::pCCResType

#### 8.896.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType

#### 8.896.2.5 voice_CLIPResp* unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp

#### 8.896.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIP_t::pFailCause

## 8.897 unpack_voice_SLQSVoiceGetCLIR_t Struct Reference

### Data Fields

- voice_CLIRResp * pCLIRResp
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.897.1 Detailed Description

This structure contains Voice Get Calling Line Identification Restriction (CLIR) Response Parameters

**Parameters**

| | |
|---|---|
| *pCLIRResp* | • Pointer to structure of CLIRResp (optional)<br>  – See voice_CLIRResp for more information<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>  – 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>  – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>  – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>  – See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.897.2 Field Documentation

#### 8.897.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCLIR_t::pAlphaIDInfo

#### 8.897.2.2 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCallID

#### 8.897.2.3 uint8_t* unpack_voice_SLQSVoiceGetCLIR_t::pCCResType

#### 8.897.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType

#### 8.897.2.5 voice_CLIRResp* unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp

#### 8.897.2.6 uint16_t* unpack_voice_SLQSVoiceGetCLIR_t::pFailCause

## 8.898 unpack_voice_SLQSVoiceGetCNAP_t Struct Reference

### Data Fields

- voice_CNAPResp * pCNAPResp
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.898.1 Detailed Description

This structure contains Voice Get Calling Name Presentation(CNAP) Response Parameters

**Parameters**

| | |
|---|---|
| *pCNAPResp* | • Pointer to structure of CNAPResp (optional)<br>  – See voice_CNAPResp for more information<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>  – 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>  – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>  – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.898.2 Field Documentation

#### 8.898.2.1 voice_alphaIDInfo∗ unpack_voice_SLQSVoiceGetCNAP_t::pAlphaIDInfo

#### 8.898.2.2 uint8_t∗ unpack_voice_SLQSVoiceGetCNAP_t::pCallID

#### 8.898.2.3 uint8_t∗ unpack_voice_SLQSVoiceGetCNAP_t::pCCResType

#### 8.898.2.4 voice_ccSUPSType∗ unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType

#### 8.898.2.5 voice_CNAPResp∗ unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp

#### 8.898.2.6 uint16_t∗ unpack_voice_SLQSVoiceGetCNAP_t::pFailCause

## 8.899 unpack_voice_SLQSVoiceGetCOLP_t Struct Reference

### Data Fields

- voice_COLPResp ∗ pCOLPResp
- uint16_t ∗ pFailCause
- voice_alphaIDInfo ∗ pAlphaIDInfo
- uint8_t ∗ pCCResType
- uint8_t ∗ pCallID
- voice_ccSUPSType ∗ pCCSUPSType

### 8.899.1 Detailed Description

This structure contains Voice Get Connected Line Identification Presentation(COLP) Response Parameters

**Parameters**

| | |
|---|---|
| *pCOLPResp* | • Pointer to structure of COLPResp (optional)<br>– See voice_COLPResp for more information<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>– See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>– 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>– See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.899.2 Field Documentation

#### 8.899.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLP_t::pAlphaIDInfo

#### 8.899.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCallID

#### 8.899.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLP_t::pCCResType

#### 8.899.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType

#### 8.899.2.5 voice_COLPResp* unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp

#### 8.899.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLP_t::pFailCause

## 8.900 unpack_voice_SLQSVoiceGetCOLR_t Struct Reference

## Data Fields

- voice_COLRResp * pCOLRResp
- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.900.1 Detailed Description

This structure contains Voice Get Connected Line Identification Restriction(COLR) Response Parameters

**Parameters**

| | |
|---|---|
| *pCOLRResp* | • Pointer to structure of COLRResp (optional)<br>  – See voice_COLRResp for more information<br>• NULL pointer - Invalid data. |
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>  – 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>  – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>  – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>  – See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.900.2 Field Documentation

#### 8.900.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo

#### 8.900.2.2 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCallID

#### 8.900.2.3 uint8_t* unpack_voice_SLQSVoiceGetCOLR_t::pCCResType

#### 8.900.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType

#### 8.900.2.5 voice_COLRResp* unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp

#### 8.900.2.6 uint16_t* unpack_voice_SLQSVoiceGetCOLR_t::pFailCause

## 8.901 unpack_voice_SLQSVoiceGetConfig_t Struct Reference

### Data Fields

- uint8_t * pAutoAnswerStat
- voice_airTimer * pAirTimerCnt
- voice_roamTimer * pRoamTimerCnt
- uint8_t * pCurrTTYMode
- voice_prefVoiceSO * pCurPrefVoiceSO
- voice_curAMRConfig * pCurAMRConfig
- uint8_t * pCurVoicePrivacyPref
- uint8_t * pCurVoiceDomainPref

### 8.901.1 Detailed Description

This structure contains Voice Get Configuration Response Parameters.

**Parameters**

| | |
|---|---|
| *pAutoAnswerStat(optional)* | • Auto Answer Status<br>• Value returned is read from NV_AUTO_ANSWER_I.<br>– 0x00 - Disabled<br>– 0x01 - Enabled<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pAirTimerCnt(optional)* | • Air Timer Count<br>• Value returned is read from NV_AIR_CNT_I.<br>• See voice_airTimer for more information<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pRoamTimer-Cnt(optional)* | • Roam Timer Count<br>• Value returned is read from NV_ROAM_CNT_I.<br>• See voice_roamTimer for more information<br>• NULL pointer - Invalid data. |
| *pCurrTTY-Mode(optional)* | • Current TTY Mode<br>• Value returned is read from NV_TTY_I.<br>– 0x00 - TTY_MODE_FULL - Full<br>– 0x01 - TTY_MODE_VCO - Voice carry over<br>– 0x02 - TTY_MODE_HCO - Hearing carry over<br>– 0x03 - TTY_MODE_OFF - Off<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCurPrefVoiceS-O(optional)* | • Current Preferred Voice SO<br>• Value returned is read from NV_PREF_VOICE_SO_I.<br>• See voice_prefVoiceSO for more information<br>• NULL pointer - Invalid data. |
| *pCurAMR-Config(optional)* | • Current Adaptive Multi-Rate Configuration.<br>• Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_AMR_COD-EC_PREFERENCE_CONFIG_I.<br>• See voice_curAMRConfig for more information<br>• NULL pointer - Invalid data. |
| *pCurVoice-Privacy-Pref(optional)* | • Current Voice Privacy Preference<br>• Value returned is read from NV_VOICE_PRIV_I.<br>– 0x00 - Standard privacy<br>– 0x01 - Enhanced privacy<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCurVoice-Domain-Pref(optional)* | • Current Voice Domain Preference.<br>– 0x00 - Circuit-switched (CS) only<br>– 0x01 - Packet-switched (PS) only<br>– 0x02 - CS is preferred; PS is secondary<br>– 0x03 - PS is preferred; CS is secondary<br>– 0xFF - Not Available<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.901.2 Field Documentation

#### 8.901.2.1 voice_airTimer* unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt

#### 8.901.2.2 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat

#### 8.901.2.3 voice_curAMRConfig* unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig

#### 8.901.2.4 voice_prefVoiceSO* unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO

#### 8.901.2.5 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode

#### 8.901.2.6 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref

#### 8.901.2.7 uint8_t* unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref

#### 8.901.2.8 voice_roamTimer* unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt

## 8.902 unpack_voice_SLQSVoiceManageCalls_t Struct Reference

### Data Fields

- uint16_t * pFailCause

### 8.902.1 Detailed Description

This structure contains Failure cause Information. Populated when API Fails.

**Parameters**

| | |
|---|---|
| *pFailCause* | • Supplementary service failure causes (optional, supply NULL if not required).<br>• See Table8 qaGobiApiTableVoiceCallEndReasons.h for supplementary services failure cause<br>  – 0xFFFF is the value when the information is not received from device<br>• NULL pointer - Invalid data. |

### 8.902.2 Field Documentation

#### 8.902.2.1 uint16_t* unpack_voice_SLQSVoiceManageCalls_t::pFailCause

## 8.903 unpack_voice_SLQSVoiceSendFlash_t Struct Reference

### Data Fields

- uint8_t * pCallID

### 8.903.1 Detailed Description

This structure contains the flash information associated with a call.

**Parameters**

| | |
|---|---|
| *pCallID* | • Unique call identifier associated with the current call.<br>• NULL pointer - Invalid data. |

### 8.903.2 Field Documentation

#### 8.903.2.1 uint8_t* unpack_voice_SLQSVoiceSendFlash_t::pCallID

## 8.904 unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference

### Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.904.1 Detailed Description

This structure contains Voice Set Call Barring Password Unpack Parameters

**Parameters**

| | |
|---|---|
| *pFailCause* | • Supplementary services failure cause (optional)<br>• see qaGobiApiTableVoiceCallEndReasons.h for more information.<br>• 0xFFFF,if Not Available<br>• NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo (optional)<br>  – See voice_alphaIDInfo for more information<br>• NULL pointer - Invalid data. |
| *pCCResType* | • Call Control Result Type (optional)<br>  – 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>  – 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>  – 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>  – 0xFF - Not Available<br>• NULL pointer - Invalid data. |
| *pCallID* | • Call ID of the voice call that resulted from call control. (optional)<br>• It is present when pCCResType is present and is Voice.<br>• If zero(0) then invalid.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pCCSUPSType* | • Supplementary service data that resulted from call control (optional)<br>• Data is present when pCCResultType is present and is other than Voice.<br>  – See voice_ccSUPSType for more information<br>• NULL pointer - Invalid data. |

**Note**

Using NULL for the pointers would make sure that the parameter is not returned or has default value.

### 8.904.2 Field Documentation

#### 8.904.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphaIDInfo

#### 8.904.2.2 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID

#### 8.904.2.3 uint8_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType

#### 8.904.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType

#### 8.904.2.5 uint16_t* unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause

## 8.905 unpack_voice_SLQSVoiceSetConfig_t Struct Reference

### Data Fields

- uint8_t * pAutoAnsStatus
- uint8_t * pAirTimerStatus
- uint8_t * pRoamTimerStatus
- uint8_t * pTTYConfigStatus
- uint8_t * pPrefVoiceSOStatus
- uint8_t * pVoiceDomainPrefStatus

### 8.905.1 Detailed Description

This structure contains information about the Set Configuration Response Parameters.

**Parameters**

| | |
|---|---|
| *pAutoAnsStatus* | • Auto Answer Status. (optional)<br>• Values:<br>  – 0x00 - Information was written successfully<br>  – 0x01 - Information write failed<br>  – 0xFF - Not Available.<br>  – NULL pointer - Invalid data. |

| | |
|---|---|
| *pAirTimerStatus* | • Air Timer Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available.<br>– NULL pointer - Invalid data. |
| *pRoamTimer-Status* | • Roam Timer Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available.<br>– NULL pointer - Invalid data. |
| *pTTYConfig-Status* | • TTY Config Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available.<br>– NULL pointer - Invalid data. |
| *pPrefVoiceSO-Status* | • Preferred Voice SO Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available.<br>– NULL pointer - Invalid data. |
| *pVoiceDomain-PrefStatus* | • Voice-Domain Preference Status. (optional)<br>• Values:<br>– 0x00 - Information was written successfully<br>– 0x01 - Information write failed<br>– 0xFF - Not Available.<br>– NULL pointer - Invalid data. |

**Note**

Parameters which are mentioned as NULL will be ignored.

## 8.905.2 Field Documentation

### 8.905.2.1 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus

### 8.905.2.2 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus

#### 8.905.2.3 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOStatus

#### 8.905.2.4 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus

#### 8.905.2.5 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus

#### 8.905.2.6 uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus

## 8.906 unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference

### Data Fields

- uint16_t * pFailCause
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCCResultType
- uint8_t * pCallID
- voice_ccSUPSType * pCCSUPSType

### 8.906.1 Detailed Description

This structure contains Supplementary Service response parameters related to different features and their activation, deactivation, registration and erasure (applicable only for 3GPP)

**Parameters**

| | |
|---|---|
| *pFailCause* | • Supplementary service failure causes (optional, supply NULL if not required).<br>– 0xFFFF is the value when the information is not received from device<br>– NULL pointer - Invalid data. |
| *pAlphaIDInfo* | • Pointer to structure of alphaIDInfo. The parameter used to pass the alpha (if any) given by the SIM/R-UIM after call control (optional, supply NULL if not required)<br>– See voice_alphaIDInfo for more information<br>– NULL pointer - Invalid data. |
| *pCCResultType* | • Call control result types (optional, supply NULL if not required)<br>– 0x00 - CC_RESULT_TYPE_VOICE - Voice<br>– 0x01 - CC_RESULT_TYPE_SUPS - Supplementary service<br>– 0x02 - CC_RESULT_TYPE_USSD - Unstructured supplementary service<br>– 0xFF - if the device does not provide this information<br>– NULL pointer - Invalid data. |
| *pCallID* | • Unique call identifier for the dialed call (optional, supply NULL if not required)<br>– 0x00 - if the device does not provide this information<br>– NULL pointer - Invalid data. |

| | |
|---|---|
| *pCCSUPSType* | • Data is present when pCCResultType is present and is other than Voice. (optional, supply NULL if not required)<br>– See voice_ccSUPSType for more information<br>– NULL pointer - Invalid data. |

### 8.906.2 Field Documentation

#### 8.906.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSetSUPSService_t::pAlphaIDInfo

#### 8.906.2.2 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCallID

#### 8.906.2.3 uint8_t* unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType

#### 8.906.2.4 voice_ccSUPSType* unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType

#### 8.906.2.5 uint16_t* unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause

## 8.907 unpack_voice_SLQSVoiceStartContDTMF_t Struct Reference

### Data Fields

- uint8_t * pCallID

### 8.907.1 Detailed Description

This structure contains parameters of continuous DTMF

**Parameters**

| | |
|---|---|
| *pCallID* | • Call ID associated with call on which the DTMF information has to be sent.<br>Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.<br>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.<br>• If the call ID value received is 0, no value has been returned by the device<br>• NULL pointer - Invalid data. |

### 8.907.2 Field Documentation

#### 8.907.2.1 uint8_t* unpack_voice_SLQSVoiceStartContDTMF_t::pCallID

## 8.908 unpack_voice_SLQSVoiceStopContDTMF_t Struct Reference

### Data Fields

- uint8_t callID

### 8.908.1 Detailed Description

This structure contains parameters of stop continuous DTMF

**Parameters**

| | |
|---|---|
| *pCallID* | • Call ID associated with call on which the DTMF information has to be sent.<br>Stop continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.<br>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.<br>• If the call ID value received is 0, no value has been returned by the device |

### 8.908.2 Field Documentation

#### 8.908.2.1 uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID

## 8.909 unpack_voice_SLQSVoiceSUPSCallback_ind_t Struct Reference

### Data Fields

- voice_SUPSInfo SUPSInformation
- uint8_t * pSvcClass
- uint8_t * pReason
- uint8_t * pCallFWNum
- uint8_t * pCallFWTimerVal
- struct voice_USSInfo * pUSSInfo
- uint8_t * pCallID
- voice_alphaIDInfo * pAlphaIDInfo
- uint8_t * pCallBarPasswd
- voice_newPwdData * pNewPwdData
- uint8_t * pDataSrc
- uint16_t * pFailCause
- voice_getCallFWInfo * pCallFwdInfo
- voice_CLIRResp * pCLIRstatus
- voice_CLIPResp * pCLIPstatus
- voice_COLPResp * pCOLPstatus
- voice_COLRResp * pCOLRstatus
- voice_CNAPResp * pCNAPstatus

### 8.909.1 Detailed Description

This structure contains the parameters passed for SUPS info indication by the device.

**Parameters**

| | |
|---|---|
| *SUPS-Information(mandatory)* | See voice_SUPSInfo for more information. |
| *pSvc-Class(optional)* | • Service class is a combination (sum) of information class constants (optional)<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes. |

| | |
|---|---|
| *p-Reason(optional)* | • See qaGobiApiTableCallControlReturnReasons.h for return reasons. |
| *pCallFW-Num(optional)* | • Call forwarding number to be registered with the network.<br>• ASCII String, NULL terminated. |
| *pCallFWTimer-Val(optional)* | • Call Forwarding No Reply Timer.<br>– Range: 5 to 30 in steps of 5. |
| *pUSS-Info(optional)* | • See voice_USSInfo for more information. |
| *pCallID(optional)* | • Call identifier of the voice call that has been modified to a supplementary service as a result of call control. |
| *pAlphaID-Info(optional)* | • See voice_alphaIDInfo for more information. |
| *pCallBar-Passwd(optional)* | • Password is required if call barring is provisioned using a password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999.<br>• This also serves as the old password in the register password scenario. |
| *pNewPwd-Data(optional)* | • See voice_newPwdData for more information. |
| *pData-Src(optional)* | • Sups Data Source.<br>• Used to distinguish between the supplementary service data sent to the network and the response received from the network.<br>• If absent, the supplementary service data in this indication can be assumed as a request sent to the network. |
| *pFail-Cause(optional)* | • Supplementary services failure cause.<br>• See qaGobiApiTableVoiceCallEndReasons.h for more information. |
| *pCallFwd-Info(optional)* | • See voice_getCallFWInfo for more information. |
| *pCLI-Rstatus(optional)* | • See voice_CLIRResp for more information. |
| *pCLI-Pstatus(optional)* | • See voice_CLIPResp for more information. |
| *pCOL-Pstatus(optional)* | • See voice_COLPResp for more information. |

| | |
|---|---|
| *pCOL-Rstatus(optional)* | • See voice_COLRResp for more information. |
| *pCNA-Pstatus(optional)* | • See voice_CNAPResp for more information. |

**Note**

None

### 8.909.2 Field Documentation

#### 8.909.2.1 voice_alphaIDInfo* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pAlphaIDInfo

#### 8.909.2.2 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallBarPasswd

#### 8.909.2.3 voice_getCallFWInfo* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFwdInfo

#### 8.909.2.4 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWNum

#### 8.909.2.5 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWTimerVal

#### 8.909.2.6 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallID

#### 8.909.2.7 voice_CLIPResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIPstatus

#### 8.909.2.8 voice_CLIRResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIRstatus

#### 8.909.2.9 voice_CNAPResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCNAPstatus

#### 8.909.2.10 voice_COLPResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLPstatus

#### 8.909.2.11 voice_COLRResp* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLRstatus

#### 8.909.2.12 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pDataSrc

#### 8.909.2.13 uint16_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pFailCause

#### 8.909.2.14 voice_newPwdData* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pNewPwdData

#### 8.909.2.15 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pReason

#### 8.909.2.16 uint8_t* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pSvcClass

#### 8.909.2.17 struct voice_USSInfo* unpack_voice_SLQSVoiceSUPSCallback_ind_t::pUSSInfo

#### 8.909.2.18 voice_SUPSInfo unpack_voice_SLQSVoiceSUPSCallback_ind_t::SUPSInformation

## 8.910 unpack_voice_SUPSNotificationCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID

- uint8_t notifType
- uint16_t * pCUGIndex
- voice_ECTNum * pECTNum

### 8.910.1 Detailed Description

Contains the parameters passed for SUPS notification indication by the device.

**Parameters**

| | |
|---|---|
| *callID* | • Unique identifier of the call for which the notification is applicable. (mandatory) |
| *notifType* | • Notification type parameter (mandatory)<br>– 0x01 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_FORWARDED<br>Originated MO call is being forwarded to another user<br>– 0x02 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_WAITING<br>Originated MO call is waiting at the called user<br>– 0x03 - NOTIFICATION_TYPE_OUTGOING_CUG_CALL<br>Outgoing call is a CUG call<br>– 0x04 - NOTIFICATION_TYPE_OUTGOING_CALLS_BARRED<br>Outgoing calls are barred<br>– 0x05 - NOTIFICATION_TYPE_OUTGOING_CALL_IS_DEFLECTED<br>Outgoing call is deflected<br>– 0x06 - NOTIFICATION_TYPE_INCOMING_CUG_CALL<br>Incoming call is a CUG call<br>– 0x07 - NOTIFICATION_TYPE_INCOMING_CALLS_BARRED<br>Incoming calls are barred<br>– 0x08 - NOTIFICATION_TYPE_INCOMING_FORWARDED_CALL<br>Incoming call received is a forwarded call<br>– 0x09 - NOTIFICATION_TYPE_INCOMING_DEFLECTED_CALL<br>Incoming call is a deflected call<br>– 0x0A - NOTIFICATION_TYPE_INCOMING_CALL_IS_FORWARDED<br>Incoming call is forwarded to another user<br>– 0x0B - NOTIFICATION_TYPE_UNCOND_CALL_FORWARD_ACTIVE<br>Unconditional call forwarding is active<br>– 0x0C - NOTIFICATION_TYPE_COND_CALL_FORWARD_ACTIVE<br>Conditional call forwarding is active<br>– 0x0D - NOTIFICATION_TYPE_CLIR_SUPPRESSION_REJECTED<br>CLIR suppression is rejected<br>– 0x0E - NOTIFICATION_TYPE_CALL_IS_ON_HOLD<br>Call is put on hold at the remote party<br>– 0x0F - NOTIFICATION_TYPE_CALL_IS_RETRIEVED<br>Call is retrieved at the remote party from the hold state<br>– 0x10 - NOTIFICATION_TYPE_CALL_IS_IN_MPTY<br>Call is in a conference<br>– 0x11 - NOTIFICATION_TYPE_INCOMING_CALL_IS_ECT<br>Incoming call is an explicit call transfer |
| *pCUGIndex* | • The CUG Index used to indicate that the incoming/outgoing call is a CUG call. (optional, NULL when not present)<br>Range: 0x00 to 0x7FFF. |

| | |
|---|---|
| *pECTNum* | • The ECT Number is used to indicate that the incoming call is an explicitly transferred call. (optional, NULL when not present)<br>Refer ECTNum for details. |

**Note**

None

### 8.910.2 Field Documentation

#### 8.910.2.1 uint8_t unpack_voice_SUPSNotificationCallback_ind_t::callID

#### 8.910.2.2 uint8_t unpack_voice_SUPSNotificationCallback_ind_t::notifType

#### 8.910.2.3 uint16_t* unpack_voice_SUPSNotificationCallback_ind_t::pCUGIndex

#### 8.910.2.4 voice_ECTNum* unpack_voice_SUPSNotificationCallback_ind_t::pECTNum

## 8.911 unpack_voice_USSDNotificationCallback_ind_t Struct Reference

### Data Fields

- uint8_t notification_Type
- voice_USSDNotificationNetworkInfo USSDNotificationNetworkInfo

### 8.911.1 Detailed Description

Structure for storing the USSD notification indication parameters.

**Parameters**

| | |
|---|---|
| *notification_Type(mandatory)* | Values: -0x01-FURTHER_USER_ACTION_NOT_REQUIRED -0x02-FURTHER_USER_ACTION_REQUIRED |
| *USSDNotificationNetworkInfo(optional)* | • USSD network info (See voice_USSDNotificationNetworkInfo) |

### 8.911.2 Field Documentation

#### 8.911.2.1 uint8_t unpack_voice_USSDNotificationCallback_ind_t::notification_Type

#### 8.911.2.2 voice_USSDNotificationNetworkInfo unpack_voice_USSDNotificationCallback_ind_t::USSDNotificationNetworkInfo

## 8.912 unpack_voice_VoiceInfoRecCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID

- voice_signalInfo * pSignalInfo
- voice_callerIDInfo * pCallerIDInfo
- uint8_t * pDispInfo
- uint8_t * pExtDispInfo
- uint8_t * pCallerNameInfo
- uint8_t * pCallWaitInd
- voice_connectNumInfo * pConnectNumInfo
- voice_connectNumInfo * pCallingPartyInfo
- voice_calledPartyInfo * pCalledPartyInfo
- voice_redirNumInfo * pRedirNumInfo
- uint8_t * pCLIRCause
- voice_NSSAudioCtrl * pNSSAudioCtrl
- uint8_t * pNSSRelease
- voice_lineCtrlInfo * pLineCtrlInfo
- voice_extDispRecInfo * pExtDispRecInfo

### 8.912.1 Detailed Description

This structure contains Voice record Information

**Parameters**

| | |
|---|---|
| *callID* | [Mandatory]<br>• Call identifier for the call. |
| *pSignalInfo[-Optional]* | • Signal Information<br>• See voice_signalInfo for more information |
| *pCallerIDInfo[-Optional]* | • Caller ID Information<br>• See voice_callerIDInfo for more information |
| *pDispInfo[-Optional]* | • Display Information |
| *pExtDispInfo[-Optional]* | • Extended Display Information |
| *pCallerName-Info[Optional]* | • Caller Name Information |
| *pCallWaitInd[-Optional]* | • Call Waiting Indicator |
| *pConnectNum-Info[Optional]* | • Connected Number Information<br>• see voice_connectNumInfo for more information |
| *pCallingParty-Info[Optional]* | • Calling Party Number Information<br>• This structure is having exactly same elements as connectNumInfo<br>• see voice_connectNumInfo for more information |

| | |
|---|---|
| *pCalledPartyInfo[Optional]* | • Called Party Number Information<br>• see voice_calledPartyInfo for more information |
| *pRedirNumInfo[Optional]* | • Redirecting Number Information<br>• see voice_redirNumInfo for more information |
| *pCLIRCause[Optional]* | • National Supplementary Services - CLIR<br>• see voice_NSSAudioCtrl for more information |
| *pNSSAudioCtrl[Optional]* | • National Supplementary Services - Audio Control |
| *pNSSRelease[Optional]* | • National Supplementary Services - Release |
| *pLineCtrlInfo[Optional]* | • Line Control Information<br>• see voice_lineCtrlInfo for more information |
| *pExtDispRecInfo[Optional]* | • Extended Display Record Information<br>• see voice_extDispRecInfo for more information |

### 8.912.2 Field Documentation

#### 8.912.2.1 uint8_t unpack_voice_VoiceInfoRecCallback_ind_t::callID

#### 8.912.2.2 voice_calledPartyInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCalledPartyInfo

#### 8.912.2.3 voice_callerIDInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerIDInfo

#### 8.912.2.4 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerNameInfo

#### 8.912.2.5 voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallingPartyInfo

#### 8.912.2.6 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallWaitInd

#### 8.912.2.7 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCLIRCause

#### 8.912.2.8 voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pConnectNumInfo

#### 8.912.2.9 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pDispInfo

#### 8.912.2.10 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDispInfo

#### 8.912.2.11 voice_extDispRecInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDispRecInfo

#### 8.912.2.12 voice_lineCtrlInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pLineCtrlInfo

#### 8.912.2.13 voice_NSSAudioCtrl* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSAudioCtrl

#### 8.912.2.14 uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSRelease

#### 8.912.2.15 voice_redirNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pRedirNumInfo

#### 8.912.2.16 voice_signalInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pSignalInfo

## 8.913 unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference

### Data Fields

- uint8_t callID
- uint8_t voicePrivacy

### 8.913.1 Detailed Description

Contains the parameters passed for voice privacy change indication

**Parameters**

| | |
|---|---|
| *callID* | • Unique identifier of the call for which the voice privacy is applicable. (mandatory) |
| *voicePrivacy* | • Voice Privacy (mandatory)<br>– 0x00 - VOICE_PRIVACY_STANDARD - Standard privacy<br>– 0x01 - VOICE_PRIVACY_ENHANCED - Enhanced privacy |

**Note**

None

### 8.913.2 Field Documentation

#### 8.913.2.1 uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::callID

#### 8.913.2.2 uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::voicePrivacy

## 8.914 unpack_wds_DHCPv4ClientLease_ind_t Struct Reference

### Data Fields

- wds_DHCPProfileIdTlv ProfileIdTlv
- wds_DHCPLeaseStateTlv DHCPv4LeaseStateTlv
- wds_IPv4AdTlv IPv4AddrTlv
- wds_DHCPLeaseOptTlv DHCPv4LeaseOptTlv

### 8.914.1 Detailed Description

This structure contains DHCP IPv4 client lease information

**Parameters**

| | |
|---|---|
| *ProfileIdTlv* | profile identifier information, see wds_DHCPProfileIdTlv for more details |
| *DHCPv4Lease-StateTlv* | lease state, see wds_DHCPLeaseStateTlv for more details |
| *IPv4AddrTlv* | IPv4 address, see wds_IPv4AdTlv for more details |
| *DHCPv4Lease-OptTlv* | DHCP lease option, see wds_DHCPLeaseOptTlv for more details |

### 8.914.2 Field Documentation

#### 8.914.2.1 wds_DHCPLeaseOptTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv

#### 8.914.2.2 wds_DHCPLeaseStateTlv unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv

#### 8.914.2.3 wds_IPv4AdTlv unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv

#### 8.914.2.4 wds_DHCPProfileIdTlv unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv

## 8.915 unpack_wds_GetAutoconnect_t Struct Reference

### Data Fields

- uint32_t ∗ psetting

### 8.915.1 Detailed Description

auto connect data session setting parameter.

**Parameters**

| | |
|---|---|
| *pSetting* | • NDIS auto connect setting<br>– 0 - Disabled<br>– 1 - Enabled |

### 8.915.2 Field Documentation

#### 8.915.2.1 uint32_t∗ unpack_wds_GetAutoconnect_t::psetting

## 8.916 unpack_wds_GetByteTotals_t Struct Reference

### Data Fields

- uint64_t ∗ pTXTotalBytes
- uint64_t ∗ pRXTotalBytes

### 8.916.1 Detailed Description

This structure contains unpack get byte totals information.

**Parameters**

| | |
|---|---|
| *pTXTotalBytes* | • Bytes transmitted without error<br>• NULL pointer - Invalid data. |
| *pRXTotalBytes* | • Bytes received without error<br>• NULL pointer - Invalid data. |

### 8.916.2 Field Documentation

#### 8.916.2.1 uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes

#### 8.916.2.2 uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes

## 8.917 unpack_wds_GetConnectionRate_t Struct Reference

### Data Fields

- uint32_t currentChannelTXRate
- uint32_t currentChannelRXRate
- uint32_t maxChannelTXRate
- uint32_t maxChannelRXRate

### 8.917.1 Detailed Description

This structure contains unpack get connection rate information.

**Parameters**

| | |
|---|---|
| *currentChannel-TXRate* | • Current channel Tx rate (in bps) |
| *currentChannel-RXRate* | • Current channel Rx rate (in bps) |
| *maxChannelTX-Rate* | • Maximum Tx rate (bps) that may be assigned to device by serving system. |
| *maxChannelRX-Rate* | • Maximum Rx rate (bps) that may be assigned to device by serving system. |

### 8.917.2 Field Documentation

#### 8.917.2.1 uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate

#### 8.917.2.2 uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate

#### 8.917.2.3 uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate

#### 8.917.2.4 uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate

## 8.918 unpack_wds_GetDataBearerTechnology_t Struct Reference

### Data Fields

- uint32_t * pDataBearer

### 8.918.1 Detailed Description

This structure contains unpack get data bearer technology information.

**Parameters**

| | |
|---|---|
| *pDataBearer[OUT]* | • Data bearer technology<br>– 0x01 - CDMA2000 1x<br>– 0x02 - CDMA 1xEV-DO Rev 0<br>– 0x03 - GSM<br>– 0x04 - UMTS<br>– 0x05 - CDMA2000 HRPD (1xEV-DO Rev A)<br>– 0x06 - EDGE<br>– 0x07 - HSDPA AND WCDMA<br>– 0x08 - WCDMA AND HSUPA<br>– 0x09 - HSDPA AND HSUPA<br>– 0x0A - LTE<br>– 0x0B - CDMA2000 EHRPD<br>– 0x0C - HSDPA+ and WCDMA<br>– 0x0D - HSDPA+ and HSUPA<br>– 0x0E - DC_HSDPA+ and WCDMA<br>– 0x0F - DC_HSDPA+ and HSUPA<br>– 0x10 - HSDPA+ and 64QAM<br>– 0x11 - HSDPA+, 64QAM and HSUPA<br>– 0x12 - TDSCDMA<br>– 0x13 - TDSCDMA and HSDPA<br>– 0xFF - Unknown |

### 8.918.2 Field Documentation

#### 8.918.2.1 uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer

## 8.919 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]

### 8.919.1 Detailed Description

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

**Parameters**

| | |
|---|---|
| *pdptype* | • 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<br>– 0 - PDP-IP (IPv4)<br>– 1 - PDP-PPP<br>– 2 - PDP-IPv6<br>– 3 - PDP-IPv4v6<br>– 0xff - Invalid data. |
| *ipaddr* | • Preferred IPv4 address to be assigned to device<br>• 0xffffffff - Invalid data. |
| *pridns* | • Primary DNS IPv4 address preference<br>• 0xffffffff - Invalid data. |
| *secdns* | • Secondary DNS IPv4 address preference<br>• 0xffffffff - Invalid data. |
| *ipaddrv6* | • Preferred IPv6 address to be assigned to device<br>• 0xffff - Invalid data. |
| *pridnsv6* | • Primary DNS Pv6 address preference<br>• 0xffff - Invalid data. |
| *secdnsv6* | • Secondary DNS IPv6 address preference<br>• 0xffff - Invalid data. |

| | |
|---|---|
| *auth* | • Bitmap that indicates authentication algorithm preference<br>– 0x00000001 - PAP preference<br>* 0 - Never performed<br>* 1 - May be performed<br>– 0x00000002 - CHAP preference<br>* 0 - Never performed<br>* 1 - May be performed<br>– All other bits are reserved and must be set to 0<br>– If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.<br>– 0xffffffff - Invalid data. |
| *namesize* | • Maximum number of characters (including NULL terminator) that profile name array can contain.<br>• 0xff - Invalid name data. |
| *name* | • Profile name |
| *apnsize* | • Maximum number of characters (including NULL terminator) that APN name array can contain<br>• 0xff - Invalid apnname data. |
| *apnname* | • Access point name. NULL-terminated string parameter that is a logical name used to select GGSN and external packet data network.<br>• If value is NULL or omitted, then subscription default value will be requested. |
| *usersize* | • Maximum number of characters (including NULL terminator) that username array can contain.<br>• 0xff - Invalid username data. |
| *username* | • Username used during network authentication |

### 8.919.2 Field Documentation

#### 8.919.2.1 int8_t unpack_wds_GetDefaultProfile_t::apnname[255]

#### 8.919.2.2 uint8_t unpack_wds_GetDefaultProfile_t::apnsize

#### 8.919.2.3 uint32_t unpack_wds_GetDefaultProfile_t::auth

#### 8.919.2.4 uint32_t unpack_wds_GetDefaultProfile_t::ipaddr

#### 8.919.2.5 uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6

#### 8.919.2.6 int8_t unpack_wds_GetDefaultProfile_t::name[255]

#### 8.919.2.7 uint8_t unpack_wds_GetDefaultProfile_t::namesize

#### 8.919.2.8 uint32_t unpack_wds_GetDefaultProfile_t::pdptype

#### 8.919.2.9 uint32_t unpack_wds_GetDefaultProfile_t::pridns

#### 8.919.2.10 uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6

#### 8.919.2.11 uint32_t unpack_wds_GetDefaultProfile_t::secdns

#### 8.919.2.12 uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6

#### 8.919.2.13 int8_t unpack_wds_GetDefaultProfile_t::username[255]

#### 8.919.2.14 uint8_t unpack_wds_GetDefaultProfile_t::usersize

## 8.920 unpack_wds_GetDefaultProfileNum_t Struct Reference

### Data Fields

- uint8_t index

### 8.920.1 Detailed Description

This structure contains unpack Get Default Profile Number information.

**Parameters**

| | |
|---|---|
| *index* | profile index |

### 8.920.2 Field Documentation

#### 8.920.2.1 uint8_t unpack_wds_GetDefaultProfileNum_t::index

## 8.921 unpack_wds_GetDormancyState_t Struct Reference

### Data Fields

- uint32_t dormancyState

### 8.921.1 Detailed Description

This structure contains unpack get dormancy state information.

**Parameters**

| | |
|---|---|
| *dormancyState* | • Dormancy state of current packet data session<br>– 1 - Traffic channel dormant<br>– 2 - Traffic channel active |

### 8.921.2 Field Documentation

#### 8.921.2.1 uint32_t unpack_wds_GetDormancyState_t::dormancyState

## 8.922 unpack_wds_GetLastMobileIPError_t Struct Reference

### Data Fields

- uint32_t error

### 8.922.1 Detailed Description

structure to store last mobile IP error.

**Parameters**

| | |
|---|---|
| *error* | • Status of last MIP call (or attempt)<br>– Zero - Success<br>– NonZero - Error code<br>See qaGobiApiTableCallEndReasons.h for Mobile IP Error codes |

### 8.922.2 Field Documentation

#### 8.922.2.1 uint32_t unpack_wds_GetLastMobileIPError_t::error

## 8.923 unpack_wds_GetMobileIP_t Struct Reference

### Data Fields

- uint32_t mipMode

### 8.923.1 Detailed Description

This structure contains unpack get mobile IP information.

**Parameters**

| | |
|---|---|
| *mipMode* | • Mobile IP setting<br>– 0 - Mobile IP off (simple IP only)<br>– 1 - Mobile IP preferred<br>– 2 - Mobile IP only |

### 8.923.2 Field Documentation

#### 8.923.2.1 uint32_t unpack_wds_GetMobileIP_t::mipMode

## 8.924 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 HAState
- uint32_t AAAState

### 8.924.1 Detailed Description

This structure contains unpack get mobile IP profile information.

**Parameters**

| | |
|---|---|
| *enabled* | • Profile enabled:<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Invalid data. |
| *address* | • Home IPv4 address:<br>– 0xFFFFFFFF - Invalid data. |
| *primaryHA* | • Primary home agent IPv4 address<br>– 0xFFFFFFFF - Invalid data. |
| *secondaryHA* | • Secondary home agent IPv4 address<br>– 0xFFFFFFFF - Invalid data. |
| *revTunneling* | • Reverse tunneling enabled<br>– 0 - Disabled<br>– 1 - Enabled<br>– 0xFF - Invalid data. |
| *naiSize* | • The maximum number of characters (including NULL terminator) that the NAI array can contain.<br>• 0xFF - Invalid data. |
| *NAI* | • Network access identifier string<br>• 0xFF - Invalid data. |

| | |
|---|---|
| *HASPI* | • Home agent security parameter index<br>• 0xFF - Invalid data. |
| *AAASPI* | • AAA server security parameter index<br>– 0xFFFFFFFF - Unknown |
| *HAState* | • Home agent key state<br>– 0 - Unset<br>– 1 - Set, default value<br>– 2 - Set, modified from default<br>– 3 - 0xFFFFFFFF - Unknown |
| *AAAState* | • AAA key state<br>– 0 - Unset<br>– 1 - Set, default value<br>– 2 - Set, modified from default<br>– 3 - 0xFFFFFFFF - Unknown |

### 8.924.2 Field Documentation

#### 8.924.2.1 uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI

#### 8.924.2.2 uint32_t unpack_wds_GetMobileIPProfile_t::AAAState

#### 8.924.2.3 uint32_t unpack_wds_GetMobileIPProfile_t::address

#### 8.924.2.4 uint8_t unpack_wds_GetMobileIPProfile_t::enabled

#### 8.924.2.5 uint32_t unpack_wds_GetMobileIPProfile_t::HASPI

#### 8.924.2.6 uint32_t unpack_wds_GetMobileIPProfile_t::HAState

#### 8.924.2.7 int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]

#### 8.924.2.8 uint8_t unpack_wds_GetMobileIPProfile_t::naiSize

#### 8.924.2.9 uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA

#### 8.924.2.10 uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling

#### 8.924.2.11 uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA

## 8.925 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

### 8.925.1 Detailed Description

This structure contains unpack get packet statistics information.

**Parameters**

| | |
|---|---|
| *pTXPacket-Successes* | • No of transmitted Packets without error.<br>• NULL pointer - Invalid data. |
| *pRXPacket-Successes* | • No of received Packets without error.<br>• NULL pointer - Invalid data. |
| *pTXPacketErrors* | • Number of outgoing packets with framing errors.<br>• NULL pointer - Invalid data. |
| *pRXPacket-Errors* | • Number of incoming packets with framing errors.<br>• NULL pointer - Invalid data. |
| *pTXPacket-Overflows* | • Number of packets dropped because Tx buffer overflowed (out of memory).<br>• NULL pointer - Invalid data. |
| *pRXPacket-Overflows* | • Number of packets dropped because Rx buffer overflowed (out of memory).<br>• NULL pointer - Invalid data. |
| *pTXOkBytes-Count* | • Number of bytes transmitted without error.<br>• NULL pointer - Invalid data. |
| *pRXOkBytes-Count* | • Number of bytes received without error.<br>• NULL pointer - Invalid data. |

| | |
|---|---|
| *pTXOKBytes-LastCall* | • Number of bytes transmitted without error during the last data call (0 if no call was made earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support<br>• NULL pointer - Invalid data. |
| *pRXOKBytes-LastCall* | • 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<br>• NULL pointer - Invalid data. |
| *pTXDropped-Count* | • Number of outgoing packets dropped.<br>• NULL pointer - Invalid data. |
| *pRXDropped-Count* | • Number of incoming packets dropped.<br>• NULL pointer - Invalid data. |

### 8.925.2 Field Documentation

#### 8.925.2.1 uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount

#### 8.925.2.2 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount

#### 8.925.2.3 uint64_t* unpack_wds_GetPacketStatistics_t::pRXOKBytesLastCall

#### 8.925.2.4 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors

#### 8.925.2.5 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows

#### 8.925.2.6 uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses

#### 8.925.2.7 uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount

#### 8.925.2.8 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount

#### 8.925.2.9 uint64_t* unpack_wds_GetPacketStatistics_t::pTXOKBytesLastCall

#### 8.925.2.10 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors

#### 8.925.2.11 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows

#### 8.925.2.12 uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses

## 8.926 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

### 8.926.1 Detailed Description

Returns the packet data transfer statistics since the start of the current packet data.

**Parameters**

| | |
|---|---|
| *tXPacket-Successes* | • No. of packets transmitted without error<br>• 0xffffffff - Invalid data. |
| *rXPacket-Successes* | • No. of packets received without error<br>• 0xffffffff - Invalid data. |
| *tXPacketErrors* | • No. of outgoing packets with framing errors<br>• 0xffffffff - Invalid data. |
| *rXPacketErrors* | • No. of incoming packets with framing errors<br>• 0xffffffff - Invalid data. |
| *txPacket-Overflows* | • Number of packets dropped because Tx buffer overflowed<br>• 0xffffffff - Invalid data. |
| *rxPacket-Overflows* | • Number of packets dropped because Rx buffer overflowed<br>• 0xffffffff - Invalid data. |
| *txOkBytesCount* | • No of bytes transmitted without error.<br>• 0xffffffffffffffff - Invalid data. |
| *rxOkBytesCount* | • No of bytes received without error.<br>• 0xffffffffffffffff - Invalid data. |
| *txOKBytesLast-Call* | • 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<br>• 0xffffffffffffffff - Invalid data. |

| | |
|---|---|
| *rxOKBytesLast-Call* | • 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<br>• 0xffffffffffffffff - Invalid data. |
| *txDroppedCount* | • Number of outgoing packets dropped.<br>• 0xffffffff - Invalid data. |
| *rxDroppedCount* | • Number of incoming packets dropped.<br>• 0xffffffff - Invalid data. |

### 8.926.2 Field Documentation

#### 8.926.2.1 uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount

#### 8.926.2.2 uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount

#### 8.926.2.3 uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall

#### 8.926.2.4 uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors

#### 8.926.2.5 uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows

#### 8.926.2.6 uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses

#### 8.926.2.7 uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount

#### 8.926.2.8 uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount

#### 8.926.2.9 uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall

#### 8.926.2.10 uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors

#### 8.926.2.11 uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows

#### 8.926.2.12 uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses

## 8.927 unpack_wds_GetSessionDuration_t Struct Reference

### Data Fields

- uint64_t callDuration

### 8.927.1 Detailed Description

This structure contains unpack get session duration information.

**Parameters**

| | |
|---|---|
| *callDuration* | • Duration of the current packet session in milliseconds |

### 8.927.2 Field Documentation

#### 8.927.2.1 uint64_t unpack_wds_GetSessionDuration_t::callDuration

## 8.928 unpack_wds_GetSessionState_t Struct Reference

### Data Fields

- uint32_t connectionStatus

### 8.928.1 Detailed Description

This structure contains unpack get session state information.

**Parameters**

| | |
|---|---|
| *connection-Status* | - state of the current packet data session.<br>• 1 - Disconnected.<br>• 2 - Connected.<br>• 3 - Suspended.<br>• 4 - Authenticating. |

### 8.928.2 Field Documentation

#### 8.928.2.1 uint32_t unpack_wds_GetSessionState_t::connectionStatus

## 8.929 unpack_wds_RMSetTransferStatistics_t Struct Reference

### 8.929.1 Detailed Description

This structure contains unpack fetch current data system transfer statistics information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.930 unpack_wds_SetMobileIPProfile_t Struct Reference

### 8.930.1 Detailed Description

This structure contains set mobile IP profile information.

**Parameters**

| | |
|---|---|
| *NULL* | • this is a dummy structure |

## 8.931 unpack_wds_SLQSCreateProfile_t Struct Reference

### Data Fields

- PackCreateProfileOut * pCreateProfileOut
- uint8_t * pProfileID
- uint16_t Tlvresult

### 8.931.1 Detailed Description

This structure contains unpack create profile information.

**Parameters**

| | |
|---|---|
| *pCreateProfile-Out* | • SLQS Create profile Information<br>• See PackCreateProfileOut |
| *pProfileID* | • SLQS profile identifier information |
| *Tlvresult* | • TLV present flag, TURE if presented, otherwise FALSE |

### 8.931.2 Field Documentation

#### 8.931.2.1 PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut

#### 8.931.2.2 uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID

#### 8.931.2.3 uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult

## 8.932 unpack_wds_SLQSDeleteProfile_t Struct Reference

### Data Fields

- uint16_t extendedErrorCode

### 8.932.1 Detailed Description

This structure contains unpack delete profile information.

**Parameters**

| | |
|---|---|
| *extendedErrorCode[OUT]* | • The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_REG_xxx.<br>• Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device.<br>• See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description.<br>• 0xffff - Invalid data. |

### 8.932.2 Field Documentation

#### 8.932.2.1 uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode

## 8.933 unpack_wds_SLQSDUNCallInfoCallBack_ind_t Struct Reference

### Data Fields

- wds_ConnStatusTlv CSTlv
- wds_LastMdmCallEndRsnTlv LMCERTlv
- wds_TXBytesOKTlv TXBOTlv
- wds_RXBytesOKTlv RXBOTlv
- wds_DormStatTlv DSTlv
- wds_DataBearTechTlv DBTTlv
- wds_ChannelRateTlv CRTlv

### 8.933.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

**Parameters**

| | |
|---|---|
| *CSTlv* | • Connection status tlv<br>• See wds_ConnStatusTlv for more information |
| *LMCERTlv* | • last modem call end reason tlv<br>• see wds_LastMdmCallEndRsnTlv for more information |
| *TXBOTlv* | • Tx bytes OK tlv<br>• see wds_TXBytesOKTlv for more information |
| *RXBOTlv* | • Rx bytes OK tlv<br>• see wds_RXBytesOKTlv for more information |
| *DSTlv* | • Dormancy status tlv<br>• see wds_DormStatTlv for more information |

| | |
|---|---|
| *DBTTlv* | • Data bear technology tlv<br>• see wds_DataBearTechTlv for more information |
| *CRTlv* | • Channel rate Tlv<br>• see wds_ChannelRateTlv for more information |

**Note**

: None

### 8.933.2 Field Documentation

#### 8.933.2.1 wds_ChannelRateTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CRTlv

#### 8.933.2.2 wds_ConnStatusTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CSTlv

#### 8.933.2.3 wds_DataBearTechTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DBTTlv

#### 8.933.2.4 wds_DormStatTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DSTlv

#### 8.933.2.5 wds_LastMdmCallEndRsnTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::LMCERTlv

#### 8.933.2.6 wds_RXBytesOKTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::RXBOTlv

#### 8.933.2.7 wds_TXBytesOKTlv unpack_wds_SLQSDUNCallInfoCallBack_ind_t::TXBOTlv

## 8.934 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

### 8.934.1 Detailed Description

This structure contains unpack get 3GPP configure item information.

**Parameters**

| | | |
|---|---|---|
| | *LTEAttachProfile* | • LTE Attach Profile<br>  – points to a single WORD Value indicating the attached LTE Profile<br>  – Optional parameter with possible values 1-16 (EM/MC73xx or earlier)<br>• This setting is deprecated on MC/EM74xx<br>• 0xffff - Invalid data. |

| | | |
|---|---|---|
| | *profileList* | Profile List<br>• an array of 4 profile configurations<br>• Each element points to a single WORD value indicating profile<br>• Optional parameter with possible values<br>– 1 - 16 (MC/EM73xx and before)<br>– 1 - 24 (MC/EM74xx and onwards)<br>• function SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present<br>• 0xffff - Invalid data.<br>• Note: the 5th entry is currently ignored, please set it to zero |
| `out` | *defaultPDN-Enabled* | • 0 - disabled<br>• 1 - enabled<br>• 0xff - Invalid data. |
| `out` | *_3gppRelease* | 3GPP release<br>• 0 - Release_99<br>• 1 - Release_5<br>• 2 - Release_6<br>• 3 - Release_7<br>• 4 - Release_8<br>• 5 - Release_9 (In 9x30 and toworads)<br>• 6 - Release_10 (In 9x30 and toworads)<br>• 7 - Release_11 (In 9x30 and toworads)<br>• 0xff - Invalid data. |
| `out` | *LTEAttach-ProfileList* | • pointer to WORD array indiciating LTE Attach Profile List<br>– Optional parameter<br>– possible values: 1-24<br>– This setting is only supported for MC/EM74xx onwards<br>– Please provide attach profiles in order of decreasing priority in this list. |
| `in,out` | *LTEAttach-ProfileListLen* | • Number of element in pLTEAttachProfileList<br>– valid range: 1-24<br>– This setting is only supported for MC/EM74xx onwards<br>• 0xffff - Invalid LTEAttachProfileList data. |

## 8.934.2 Field Documentation

### 8.934.2.1 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::_3gppRelease

### 8.934.2.2 uint8_t unpack_wds_SLQSGet3GPPConfigItem_t::defaultPDNEnabled

### 8.934.2.3 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile

### 8.934.2.4 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]

#### 8.934.2.5 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen

#### 8.934.2.6 uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]

## 8.935 unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference

### Data Fields

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

### 8.935.1 Detailed Description

Data System Status

**Parameters**

| | |
|---|---|
| *prefNetwork* | • Preferred Network<br>• Values:<br>– 0 - 3GPP<br>– 1 - 3GPP2<br>– 0xff - Invalid data system status. |
| *networkInfoLen* | • As input parameter size assigned to next parameter i.e. network information<br>• As output the actual number of network information elements returned by the device |
| *currNetworkInfo* | • Network information<br>• See currNetworkInfo for more details |

### 8.935.2 Field Documentation

#### 8.935.2.1 currNetworkInfo unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]

#### 8.935.2.2 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen

#### 8.935.2.3 uint8_t unpack_wds_SLQSGetCurrDataSystemStat_t::prefNetwork

## 8.936 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

### 8.936.1 Detailed Description

This structure contains unpack get current channel rate information.

**Parameters**

| | |
|---|---|
| *current_channel-_tx_rate* | • Current Channel Tx Rate.<br>• Instantaneous channel Tx rate in bits per second.<br>• In 9x15, this is the total current channel rate for all PDNs combined.<br>• In 9x30 and later, this is the channel rate for a specific PDN. |
| *current_channel-_rx_rate* | • Current Channel Rx Rate.<br>• Instantaneous channel Rx rate in bits per second.<br>• In 9x15, this is the total current channel rate for all PDNs combined.<br>• In 9x30 and later, this is the channel rate for a specific PDN |
| *max_channel_-tx_rate* | • Max Channel Tx Rate.<br>• Maximum total Tx rate that modem is able to support in current serving system in bits per second.<br>• In 9x15, this is a default hard coded value for the current serving system. |
| *max_channel_-rx_rate* | • Max Channel Rx Rate.<br>• Maximum total Rx rate that modem is able to support in current serving system in bits per second.<br>• In 9x15, this is a default hard coded value for the current serving system. |

### 8.936.2 Field Documentation

#### 8.936.2.1 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_rx_rate

#### 8.936.2.2 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::current_channel_tx_rate

#### 8.936.2.3 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_rx_rate

#### 8.936.2.4 uint32_t unpack_wds_SLQSGetCurrentChannelRate_t::max_channel_tx_rate

## 8.937 unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference

### Data Fields

- uint8_t dataBearerMask
- qmiWDSDataBearerTechnology curDataBearerTechnology
- qmiWDSDataBearerTechnology lastCallDataBearerTechnology

### 8.937.1 Detailed Description

Structure to hold the data bearer technology values

**Parameters**

| | |
|---|---|
| *dataBearerMask[OUT]* | • This bit mask indicates if data bearer information for the current and/or last call has been received from the device. If a bit is set, then the information is available in the corresponding structure i.e. the one provided by the caller. Refer to liteQmiDataBearerMasks for bit-mask positions.<br>• 0xff - Invalid data. |
| *curDataBearerTechnology[OUT]* | • current data bearer technology value.<br>• See qmiWDSDataBearerTechnology<br>– NULL if the parameter is not required |
| *lastCallDataBearerTechnology[OUT]* | • last call data bearer technology value.<br>• See qmiWDSDataBearerTechnology<br>– NULL if the parameter is not required |

### 8.937.2 Field Documentation

#### 8.937.2.1 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::curDataBearerTechnology

#### 8.937.2.2 uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask

#### 8.937.2.3 qmiWDSDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology

## 8.938 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

### 8.938.1 Detailed Description

This structure contains the DUN Call Info response parameters

**Parameters**

| | |
|---|---|
| *connectionStatus* | • See connectionStatus for more information |

| | |
|---|---|
| *callEndReason* | • Last modem call end reason<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason<br>• Only valid if the last call made was DUN, else zero is returned<br>• 0xffff - Invalid data. |
| *txOKBytesCount* | • Number of bytes transmitted without error<br>• Returned only if a data call is up<br>• 0xffffffffffffffff - Invalid data. |
| *rxOKBytesCount* | • Number of bytes received without error<br>• Returned only if a data call is up<br>• 0xffffffffffffffff - Invalid data. |
| *dormancyStatus* | • Current traffic channel status<br>• Returned if a data call is up<br>– 0x01 - Traffic channel dormant<br>– 0x02 - Traffic channel active<br>– 0xff - Invalid data. |
| *dataBearerTech* | • Current data bearer technology<br>• Returned only if a data call is up<br>– 0x01 - cdma2000 1X<br>– 0x02 - cdma2000 HRPD (1xEV-DO)<br>– 0x03 - GSM<br>– 0x04 - UMTS<br>– 0x05 - cdma200 HRPD ( 1xEV-DO RevA)<br>– 0x06 - EDGE<br>– 0x07 - HSDPA and WCDMA<br>– 0x08 - WCDMA and HSUPA<br>– 0x09 - HSDPA and HSUPA<br>– 0x0A - LTE<br>– 0x0B - cdma2000 EHRPD<br>– 0x0C - HSDPA+ and WCDMA<br>– 0x0D - HSDPA+ and HSUPA<br>– 0x0E - DC_HSDPA+ and WCDMA<br>– 0x0F - DC_HSDPA+ and HSUPA<br>– 0x10 - HSDPA+ and 64QAM<br>– 0x11 - HSDPA+, 64QAM and HSUPA<br>– 0x12 - TDSCDMA<br>– 0x13 - TDSCDMA and HSDPA<br>– 0xFF - Unknown |
| *channelRate* | • See dunchannelRate for more information |

| | |
|---|---|
| *lastCallTXOK-BytesCnt* | • Number of bytes transmitted without error during the last data call ( 0 if no call was made ).<br>• Return only if not in a call and the previous call was made using DUN.<br>• 0xffffffffffffffff - Invalid data. |
| *lastCallRXOK-BytesCnt* | • Number of bytes transmitted without error during the last data call ( 0 if no call was made ).<br>• Return only if not in a call and the previous call was made using DUN.<br>• 0xffffffffffffffff - Invalid data. |
| *mdmCall-DurationActive* | • Duration that the call is active in milliseconds<br>• If the modem connection status is connected, this represents the active duration of the current DUN call<br>• If the modem connection status is disconnected, this represents the duration of the last DUN call since the device was powered up (0 if no call has been made or if the last call was not DUN)<br>• 0xffffffffffffffff - Invalid data. |
| *lastCallData-BearerTech* | • Last Call Data Bearer Technology<br>• Returned only if not in a call and when the previous call was made using DUN<br>– 0x01 - cdma2000 1X<br>– 0x02 - cdma2000 HRPD (1xEV-DO)<br>– 0x03 - GSM<br>– 0x04 - UMTS<br>– 0x05 - cdma200 HRPD (1xEV-DO Rev A)<br>– 0x06 - EDGE<br>– 0x07 - HSDPA and WCDMA<br>– 0x08 - WCDMA and HSUPA<br>– 0x09 - HSDPA and HSUPA<br>– 0x0A - LTE<br>– 0x0B - cdma2000 EHRPD<br>– 0x0C - HSDPA+ and WCDMA<br>– 0x0D - HSDPA+ and HSUPA<br>– 0x0E - DC_HSDPA+ and WCDMA<br>– 0x0F - DC_HSDPA+ and HSUPA<br>– 0x10 - HSDPA+ and 64QAM<br>– 0x11 - HSDPA+, 64QAM and HSUPA<br>– 0x12 - TDSCDMA<br>– 0x13 - TDSCDMA and HSDPA<br>– 0xFF - Unknown |

### 8.938.2 Field Documentation

#### 8.938.2.1 uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason

#### 8.938.2.2 dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate

#### 8.938.2.3 connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus

#### 8.938.2.4 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech

#### 8.938.2.5 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus

#### 8.938.2.6 uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech

#### 8.938.2.7 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt

#### 8.938.2.8 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt

#### 8.938.2.9 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive

#### 8.938.2.10 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount

#### 8.938.2.11 uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount

## 8.939 unpack_wds_SLQSGetProfileSettings_t Struct Reference

### Data Fields

- UnPackGetProfileSettingOut * pProfileSettings
- uint8_t ProfileType
- uint16_t Tlvresult

### 8.939.1 Detailed Description

This structure contains the profile setting information of parameter pOutput for API unpack_wds_SLQSGetProfileSettings

**Parameters**

| | |
|---|---|
| *pProfileSettings* | - Profile Settings |
| *ProfileType* | - Profile Type<br>• 0 - Profile 3GPP<br>• 1 - Profile 3GPP2 |
| *Tlvresult* | - unpack Tlv Result. |

### 8.939.2 Field Documentation

#### 8.939.2.1 UnPackGetProfileSettingOut* unpack_wds_SLQSGetProfileSettings_t::pProfileSettings

#### 8.939.2.2 uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType

#### 8.939.2.3 uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult

## 8.940 unpack_wds_SLQSGetRuntimeSettings_t Struct Reference

### Data Fields

- uint32_t IPv4
- uint8_t ProfileName [128]
- uint32_t PDPType
- uint8_t APNName [128]

- uint32_t PrimaryDNSV4
- uint32_t SecondaryDNSV4
- LibPackUMTSQoS UMTSGrantedQoS
- struct wds_GPRSQoS GPRSGrantedQoS
- uint8_t Username [128]
- uint32_t Authentication
- struct wds_ProfileIdentifier ProfileID
- uint32_t GWAddressV4
- uint32_t SubnetMaskV4
- uint8_t PCSCFAddrPCO
- struct wds_PCSCFIPv4ServerAddressList ServerAddrList
- struct wds_PCSCFFQDNAddressList PCSCFFQDNAddrList
- uint16_t PrimaryDNSV6 [8]
- uint16_t SecondaryDNSV6 [8]
- uint32_t Mtu
- struct wds_DomainNameList DomainList
- uint8_t IPFamilyPreference
- uint8_t IMCNflag
- uint16_t Technology
- struct wds_IPV6AddressInfo IPV6AddrInfo
- struct wds_IPV6GWAddressInfo IPV6GWAddrInfo

### 8.940.1 Detailed Description

This structure contains unpack get runtime settings information.

**Parameters**

| | |
|---|---|
| *IPv4* | ipv4 address<br>• 0xff - Invalid data. |
| *ProfileName* | profile name<br>• Invalid data when index 0 of ProfileName is equal 0xff. |
| *PDPType* | PDP type<br>• 0xffffffff - Invalid data. |
| *APNName* | APN name<br>• Invalid data when index 0 of APNName is equal 0xff. |
| *PrimaryDNSV4* | primary dns IPV4<br>• 0xffffffff - Invalid data. |
| *SecondaryDNS-V4* | secondary dns IPV4<br>• 0xffffffff - Invalid data. |
| *UMTSGranted-QoS* | UMTS Granted Qos |
| *GPRSGranted-QoS* | GPRS Granted QoS |
| *Username* | username for authentication process<br>• Invalid data when index 0 of APNName is equal 0xff. |

| | |
|---|---|
| *Authentication* | authentication for authentication process<br>• 0xffffffff - Invalid data. |
| *ProfielID* | profile ID |
| *GWAddressV4* | Gateway IPv4<br>• 0xffffffff - Invalid data. |
| *SubnetMaskV4* | Subnet mask IPV4<br>• 0xffffffff - Invalid data. |
| *PCSCFAddrPCO* | PCSCF address PCO<br>• 0xff - Invalid data. |
| *ServerAddrList* | PCSCF server address list IPV4 |
| *PCSCFFQDNAddrList* | PCSCF FQDN address list IPV4 |
| *PrimaryDNSV6* | Primary DNS IPV6<br>• invaild when all array data are equal 0xffff |
| *SecondryDNSV6* | Secondry DNS IPV6<br>• invaild when all array data are equal 0xffff |
| *Mtu* | actual (runtime) Maximum Transfer Unit<br>• 0xffffffff - Invalid data. |
| *DomainList* | domain list |
| *IPFamilyPreference* | ip family preference, it could be IPV4 or IPV6<br>• 0xff - Invalid data. |
| *IMCNflag* | IM control flag, value: TRUE or FALSE<br>• 0xff - Invalid data. |
| *Technology* | technology on which current packet data session is in progress<br>• Values:<br>– 32767 - CDMA<br>– 32764 - UMTS<br>– 30592 - EPC<br>– 30584 - modem link local<br>– 0xffff - Invalid data. |
| *IPV6AddrInfo* | address information IPV6 |
| *IPV6GWAddrInfo* | gateway address information IPV6 |

## 8.940.2 Field Documentation

### 8.940.2.1 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::APNName[128]

### 8.940.2.2 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Authentication

### 8.940.2.3 struct wds_DomainNameList unpack_wds_SLQSGetRuntimeSettings_t::DomainList

#### 8.940.2.4 struct wds_GPRSQoS unpack_wds_SLQSGetRuntimeSettings_t::GPRSGrantedQoS

#### 8.940.2.5 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4

#### 8.940.2.6 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag

#### 8.940.2.7 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference

#### 8.940.2.8 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4

#### 8.940.2.9 struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPV6AddrInfo

#### 8.940.2.10 struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPV6GWAddrInfo

#### 8.940.2.11 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu

#### 8.940.2.12 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO

#### 8.940.2.13 struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList

#### 8.940.2.14 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPType

#### 8.940.2.15 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4

#### 8.940.2.16 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]

#### 8.940.2.17 struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID

#### 8.940.2.18 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]

#### 8.940.2.19 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4

#### 8.940.2.20 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]

#### 8.940.2.21 struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList

#### 8.940.2.22 uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4

#### 8.940.2.23 uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology

#### 8.940.2.24 LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS

#### 8.940.2.25 uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]

## 8.941 unpack_wds_SLQSModifyProfile_t Struct Reference

### Data Fields

- uint16_t * pExtErrorCode

### 8.941.1 Detailed Description

This structure contains out parameters for unpack_wds_SLQSModifyProfile

**Parameters**

| | |
|---|---|
| *pExtErrorCode* | • The extended error code received from DS Profile subsystem of type eWDS_ERR_PROFILE_-REG_xxx.<br>• Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERN-AL is returned by device.<br>• See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description.<br>• NULL pointer - Invalid data. |

### 8.941.2 Field Documentation

#### 8.941.2.1 uint16_t* unpack_wds_SLQSModifyProfile_t::pExtErrorCode

## 8.942 unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference

### Data Fields

- uint16_t Tlvresult

### 8.942.1 Detailed Description

This structure contains unpack Set IP family preference information

**Parameters**

| | |
|---|---|
| *Tlvresult* | unpack result |

### 8.942.2 Field Documentation

#### 8.942.2.1 uint16_t unpack_wds_SLQSSetIPFamilyPreference_t::Tlvresult

## 8.943 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

### 8.943.1 Detailed Description

This structure contains unpack set packet service status callback information.

**Parameters**

| | |
|---|---|
| *conn_status* | connection status.<br>• Current link status. Values:<br>– 1 - DISCONNECTED<br>– 2 - CONNECTED<br>– 3 - SUSPENDED<br>– 4 - AUTHENTICATING<br>– 0xff Invalid data. |
| *reconfigReqd* | Indicates whether the network interface on the host needs to be reconfigured.<br>• Values:<br>– 0 - No need to reconfigure.<br>– 1 - Reconfiguration required.<br>– 0xff - Invalid data. |
| *sessionEnd-Reason* | Call End Reason<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason<br>• 0xffff - invalid data. |
| *verboseSessn-EndReasonType* | Verbose call end reason type<br>• Call end reason type. Values:<br>– 0 - Unspecified<br>– 1 - Mobile IP<br>– 2 - Internal<br>– 3 - Call Manager defined<br>– 6 - 3GPP Specification defined<br>– 7 - PPP<br>– 8 - EHRPD<br>– 9 - IPv6<br>– 0xffff - invalid data. |
| *verboseSessn-EndReason* | Reason the call ended (verbose)<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason<br>• 0xffff - invalid data. |
| *ipFamily* | IP family of the packet data connection.<br>• Values<br>– 4 - IPv4<br>– 6 - IPv6<br>– 0xff - Invalid data. |
| *techName* | Technology name of the packet data connection.<br>• Values<br>– 32767 - CDMA<br>– 32764 - UMTS<br>– 30592 - EPC<br>– 30590 - EMBMS<br>– 30584 - Modem Link Local EPC is a logical interface to support LTE/eHRPD handoff. It is returned if the device supports IP session continuity. Modem Link Local is an interface for transferring data between entities on the AP and modem.<br>– 0xffff - Invalid data. |

| | |
|---|---|
| *bearerID* | • bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection.<br>• Valid Values - 0 to 16<br>• 0xFF - Invalid value. |

### 8.943.2 Field Documentation

#### 8.943.2.1 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID

#### 8.943.2.2 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status

#### 8.943.2.3 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily

#### 8.943.2.4 uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd

#### 8.943.2.5 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason

#### 8.943.2.6 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName

#### 8.943.2.7 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason

#### 8.943.2.8 uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType

## 8.944 unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference

### Data Fields

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

### 8.944.1 Detailed Description

This structure contains unpack set WDS event callback information.

**Parameters**

| | |
|---|---|
| *xferStatAvail* | transfer statistic available |
| *tx_bytes* | • Number of bytes transmitted without error |
| *rx_bytes* | • Number of bytes received without error |
| *tx_pkts* | • Number of packets transmitted without error |
| *rx_pkts* | • Number of packets received without error. |
| *mipstatAvail* | Mobile IP status available |
| *mipStatus* | • Status of the last MIP call (or attempt).<br>• Values<br>  – 0x00 - Success |
| *dBTechAvail* | Data Bearer technology available |
| *dBTechnology* | • Data Bearer technology<br>  – Values<br>  * 0x01 - cdma2000 ® 1X<br>  * 0x02 - cdma2000 ® HRPD (1xEV-DO)<br>  * 0x03 - GSM<br>  * 0x04 - UMTS<br>  * 0x05 - cdma2000 ® HRPD (1xEV-DO RevA)<br>  * 0x06 - EDGE<br>  * 0x07 - HSDPA and WCDMA<br>  * 0x08 - WCDMA and HSUPA<br>  * 0x09 - HSDPA and HSUPA<br>  * 0x0A - LTE<br>  * 0x0B - cdma2000 ® EHRPD<br>  * 0x0C - HSDPA+ and WCDMA<br>  * 0x0D - HSDPA+ and HSUPA<br>  * 0x0E - DC_HSDPA+ and WCDMA<br>  * 0x0F - DC_HSDAP+ and HSUPA<br>  * 0x10 - HSDPA+ and 64QAM<br>  * 0x11 - HSDPA+, 64QAM and HSUPA<br>  * 0x12 - TDSCDMA<br>  * 0x13 - TDSCDMA and HSDPA<br>  * 0x14 - TDSCDMA and HSUPA<br>  * -1 - Unknown |
| *dormancyStat-Avail* | Dormancy status available |

| | |
|---|---|
| *dormancyStatus* | • Dormancy status<br>• Values<br>– 1 - Traffic channel dormant<br>– 2 - Traffic channel active |
| *currDBTechAvail* | • Current Data Bearer technology available |
| *ratMask* | • RAT mask to indicate type of technology<br>• Values<br>– 0x00 - DONT_CARE<br>– 0x8000 - NULL_BEARER<br>• CDMA RAT mask<br>– 0x01 - CDMA_1X<br>– 0x02 - EVDO_REV0<br>– 0x05 - HRPD<br>– 0x0B - EHRPD<br>• UMTS RAT mask<br>– 0x03 - GPRS<br>– 0x04 - WCDMA<br>– 0x06 - EDGE<br>– 0x07 - HSDPA and WCDMA<br>– 0x08 - WCDMA and HSUPA<br>– 0x09 - HSDPA and HSUPA<br>– 0x0A - LTE<br>– 0x0C - HSDPA+ and WCDMA<br>– 0x0D - HSDPA+ and HSUPA<br>– 0x0E - DC_HSDPA+ and WCDMA<br>– 0x0F - DC_HSDPA+ and HSUPA |
| *soMask* | • SO Mask |
| *dataSysStatAvail* | • Data System Status available |
| *prefNetwork* | • preferred network<br>• Values<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *currNWInfo* | • Current Network Info<br>• see wds_currNetworkInfo for more info |
| *dBtechExtAvail* | Data bearer technology extended available |

| | |
|---|---|
| *dBtechnologyExt* | • Data bearer technology Extended.<br>• Values<br>– WDS_BEARER_TECH_NETWORK_3GPP (0) - 3GPP<br>– WDS_BEARER_TECH_NETWORK_3GPP2 (1) - 3GPP2 |
| *dBTechExtRat-Value* | • Data bearer technology Extended RAT Value<br>• Values<br>– WDS_BEARER_TECH_RAT_EX_NULL_BEARER (0x00) - NULL bearer<br>– WDS_BEARER_TECH_RAT_EX_3GPP_WCDMA (0x01) - 3GPP WCDMA<br>– WDS_BEARER_TECH_RAT_EX_3GPP_GERAN (0x02) - 3GPP GERAN<br>– WDS_BEARER_TECH_RAT_EX_3GPP_LTE (0x03) - 3GPP LTE<br>– WDS_BEARER_TECH_RAT_EX_3GPP_TDSCDMA (0x04) - 3GPP TDSCDMA<br>– WDS_BEARER_TECH_RAT_EX_3GPP_WLAN (0x05) - 3GPP WLAN<br>– WDS_BEARER_TECH_RAT_EX_3GPP_MAX (0x64) - 3GPP maximum<br>– WDS_BEARER_TECH_RAT_EX_3GPP2_1X (0x65) - 3GPP2 1X<br>– WDS_BEARER_TECH_RAT_EX_3GPP2_HRPD (0x66) - 3GPP2 HRPD<br>– WDS_BEARER_TECH_RAT_EX_3GPP2_EHRPD (0x67) - 3GPP2 EHRPD<br>– WDS_BEARER_TECH_RAT_EX_3GPP2_WLAN (0x68) - 3GPP2 WLAN<br>– WDS_BEARER_TECH_RAT_EX_3GPP2_MAX (0xC8) - 3GPP2 maximum |

| | |
|---|---|
| *dBTechExtSo-Mask* | • Data bearer technology Extended SO Mask<br>• Service Option (SO) mask to indicate the service option or type of application.An SO mask value of zero indicates that this field is ignored.<br>• Values<br>– 0x00 - SO mask unspecified<br>– 3GPP SO mask<br>* 0x01 - WCDMA<br>* 0x02 - HSDPA<br>* 0x04 - HSUPA<br>* 0x08 - HSDPAPLUS<br>* 0x10 - DC HSDPAPLUS<br>* 0x20 - 64 QAM<br>* 0x40 - HSPA<br>* 0x80 - GPRS<br>* 0x100 - EDGE<br>* 0x200 - GSM<br>* 0x400 - S2B<br>* 0x800 - LTE limited service<br>* 0x1000 - LTE FDD<br>* 0x2000 - LTE TDD<br>– 3GPP2 SO mask<br>* 0x01000000 - 1X IS95<br>* 0x02000000 - 1X IS2000<br>* 0x04000000 - 1X IS2000 REL A<br>* 0x08000000 - HDR REV0 DPA<br>* 0x10000000 - HDR REVA DPA<br>* 0x20000000 - HDR REVB DPA<br>* 0x40000000 - HDR REVA MPA<br>* 0x80000000 - HDR REVB MPA<br>* 0x100000000 - HDR REVA EMPA<br>* 0x200000000 - HDR REVB EMPA<br>* 0x400000000 - HDR REVB MMPA<br>* 0x800000000 - HDR EVDO FMC |

## 8.944.2 Field Documentation

### 8.944.2.1 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail

### 8.944.2.2 wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]

### 8.944.2.3 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail

### 8.944.2.4 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail

### 8.944.2.5 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechExtAvail

### 8.944.2.6 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtRatValue

#### 8.944.2.7 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechExtSoMask

#### 8.944.2.8 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology

#### 8.944.2.9 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBtechnologyExt

#### 8.944.2.10 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail

#### 8.944.2.11 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus

#### 8.944.2.12 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail

#### 8.944.2.13 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus

#### 8.944.2.14 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen

#### 8.944.2.15 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork

#### 8.944.2.16 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask

#### 8.944.2.17 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes

#### 8.944.2.18 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts

#### 8.944.2.19 uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask

#### 8.944.2.20 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes

#### 8.944.2.21 uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts

#### 8.944.2.22 uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail

## 8.945 unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference

### Data Fields

- wdsDhcpv4HwConfig * pHwConfig
- wdsDhcpv4OptionList * pRequestOptionList

### 8.945.1 Detailed Description

WDS SWI DHCPv4 Config Structure

**Parameters**

| | |
|---|---|
| *pHWConfig* | • pointer to HW Config structure<br>• See wdsDhcpv4HwConfig<br>• NULL pointer - Invalid data. |
| *pRequestOption-List* | • pointer to Option List structure to be sent in DHCP request<br>• See wdsDhcpv4OptionList<br>• NULL pointer - Invalid data. |

### 8.945.2 Field Documentation

#### 8.945.2.1 wdsDhcpv4HwConfig* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig

#### 8.945.2.2 wdsDhcpv4OptionList* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList

## 8.946 unpack_wds_SLQSSGetLoopback_t Struct Reference

### Data Fields

- uint8_t ByteLoopbackMode
- uint8_t ByteLoopbackMultiplier

### 8.946.1 Detailed Description

This structure contains unpack Get loopback information.

**Parameters**

| | |
|---|---|
| *ByteLoopback-Mode* | • Loopback Mode.<br>– 0 - Disable<br>– 1 - Enable<br>– 0xff - Invalid data. |
| *ByteLoopback-Multiplier* | • Loopback multiplier. Number of downlink bytes to send for each uplink byte.<br>• 0xff - Invalid data. |

### 8.946.2 Field Documentation

#### 8.946.2.1 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode

#### 8.946.2.2 uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier

## 8.947 unpack_wds_SLQSStartDataSession_t Struct Reference

### Data Fields

- uint32_t * psid
- uint32_t * pFailureReason
- uint32_t * pVerboseFailReasonType
- uint32_t * pVerboseFailureReason

### 8.947.1 Detailed Description

This structure contains unpack Start Data Session Information.

**Parameters**

| | |
|---|---|
| *psid* | • Assigned session ID when starting a data session |
| *pFailureReason* | • Reason data session failed to be established<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason<br>• NULL pointer - Invalid data. |
| *pVerboseFail-ReasonType* | • Parameter describing type of verbose failure reason<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason Type<br>• NULL pointer - Invalid data. |
| *pVerboseFailure-Reason* | • Verbose reason explaining why call failed. Depends on verbFailReasonType parameter<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason<br>• NULL pointer - Invalid data. |

### 8.947.2 Field Documentation

#### 8.947.2.1 uint32_t* unpack_wds_SLQSStartDataSession_t::pFailureReason

#### 8.947.2.2 uint32_t* unpack_wds_SLQSStartDataSession_t::psid

#### 8.947.2.3 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailReasonType

#### 8.947.2.4 uint32_t* unpack_wds_SLQSStartDataSession_t::pVerboseFailureReason

## 8.948 unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference

### Data Fields

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

### 8.948.1 Detailed Description

This structure contains the unpack parameters retrieved by the API unpack_wds_SLQSWdsSwiPDPRuntime-Settings

**Parameters**

| | |
|---|---|
| *contextId* | (optional)<br>• Context Identifier<br>– 0xFF - Not Available |
| *bearerId* | (optional)<br>• Bearer Identity<br>• An EPS bearer identity uniquely identifies an EPS bearer for one UE accessing via E-UTRAN. The EPS Bearer Identity is allocated by the MME.<br>– 0xFF - Not Available |
| *apnName* | (optional)<br>• APN name associated with the context id<br>– NULL terminated by default. |
| *ipv4Address* | (optional)<br>• IPv4 Address<br>– 0xFFFF - Not Available |
| *ipv4GWAddress* | (optional)<br>• IPv4 Gateway Address<br>– 0xFFFF - Not Available |
| *prDNSIPv4-Address* | (optional)<br>• Primary DNS IPv4 Address<br>– 0xFFFF - Not Available |
| *seDNSIPv4-Address* | (optional)<br>• Secondary DNS IPv4 Address<br>– 0xFFFF - Not Available |
| *ipv6Address* | (optional)<br>• IPv6 Address<br>• See ipv6AddressInfo for more information |
| *ipv6GWAddress* | (optional)<br>• IPv6 Gateway Address<br>• See ipv6AddressInfo for more information |
| *prDNSIPv6-Address* | (optional)<br>• Primary IPv6 DNS Address(in network byte order)<br>• This is an 8-element array of 16-bit numbers, each of which is in big-endian format<br>• 0xffffffff - Invalid data. |
| *seDNSIPv6-Address* | (optional)<br>• Secondary IPv6 DNS Address(in network byte order)<br>• This is an 8-element array of 16-bit numbers, each of which is in big-endian format<br>• 0xffffffff - Invalid data. |

| | |
|---|---|
| *prPCSCFIPv4-Address* | (optional)<br>• Primary PCSCF IPv4 Address<br>• 0xffffffff - Invalid data. |
| *sePCSCFIPv4-Address* | (optional)<br>• Secondary PCSCF IPv4 Address<br>• 0xffffffff - Invalid data. |
| *prPCSCFIPv6-Address* | (optional)<br>• Primary PCSCF IPv6 Address<br>• This is an 8-element array of 16-bit numbers, each of which is in big-endian format<br>• 0xffff - Invalid data. |
| *sePCSCFIPv6-Address* | (optional)<br>• Secondary PCSCF IPv6 Address<br>• This is an 8-element array of 16-bit numbers, each of which is in big-endian format<br>• 0xffff - Invalid data. |

### 8.948.2 Field Documentation

#### 8.948.2.1 int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]

#### 8.948.2.2 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId

#### 8.948.2.3 uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId

#### 8.948.2.4 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address

#### 8.948.2.5 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress

#### 8.948.2.6 ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address

#### 8.948.2.7 ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress

#### 8.948.2.8 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address

#### 8.948.2.9 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]

#### 8.948.2.10 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address

#### 8.948.2.11 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]

#### 8.948.2.12 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address

#### 8.948.2.13 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]

#### 8.948.2.14 uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address

#### 8.948.2.15 uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]

## 8.949 UnPackGetProfileSettingOut Struct Reference

### Data Fields

- UnpackQmiProfileInfo curProfile
- uint16_t * pExtErrCode

### 8.949.1 Detailed Description

This structure contains the profile settings output

**Parameters**

| | |
|---|---|
| *curProfile* | • Structure containing details of the current profile<br>• See UnpackQmiProfileInfo |
| *pExtErrCode* | • pointer to a 2 byte extended error code<br>• Error code will only will be present if error code eQCWWAN_ERR_QMI_EXTENDED_INTERNAL is returned by device.<br>• See qm_wds_ds_profile_extended_err_codes enum in qmerrno.h for received error description. |

### 8.949.2 Field Documentation

#### 8.949.2.1 UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile

#### 8.949.2.2 uint16_t* UnPackGetProfileSettingOut::pExtErrCode

## 8.950 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo Struct Reference

### Data Fields

- uint8_t bBinaryType
- uint8_t bState
- uint8_t bUserInputRequest
- uint16_t wUserInputTimeout
- uint32_t ulPkgDownloadSize
- uint32_t ulPkgDownloadComplete
- uint16_t wUpdateCompeteStatus
- uint8_t bSerity
- uint16_t wVersionLength
- uint8_t szVersion [128]
- uint16_t wNameLength
- uint8_t szName [128]
- uint16_t wDescriptionLength
- uint8_t szDescription [1024]
- uint8_t TlvPresent

### 8.950.1 Detailed Description

This structure contains unpack event report binary update session information parameters.

**Parameters**

| | |
|---|---|
| *bBinaryType* | : Type<br>• 1 - Firmware.<br>• 2 - User App.<br>• 3 - Legato Framework. |
| *bStat* | : State<br>• 0x01 - No binary update available.<br>• 0x02 - Query binary Download.<br>• 0x03 - Binary Downloading.<br>• 0x04 - Binary downloaded.<br>• 0x05 -Query Binary Update.<br>• 0x06 - Binary updating.<br>• 0x07 - Binary updated. |
| *bUserInput-Request* | Bit mask of available user inputs.<br>• 0x00 - No user input required. Informational indication.<br>• 0x01 - Accept.<br>• 0x02 - Reject. |
| *wUserInput-Timeout* | Timeout for user input in minutes. A value of 0 means no time-out. |
| *ulPkgDownload-Size* | The size (in bytes) of the update package |
| *ulPkgDownload-Complete* | The number of bytes being downloaded. For downloading state, this value shall be $>$ 0 and incremented toward the pkg_dload_size. For other states, the value shall be 0 as it is meaningless. |
| *wUpdate-CompeteStatus* | Result code.This field should be looked at only when the AVMS session is complete. |
| *bSerity* | Serity.<br>• 0x01 - Mandatory.<br>• 0x02 - Optional. |
| *wVersionLength* | Length of FW Version string in bytes. |
| *szVersion* | FW Version string in ASCII. |
| *wNameLength* | Length Package Name string in bytes. |
| *szName* | Package Name in UCS2. |
| *wDescription-Length* | Length of description in bytes. |
| *szDescription* | Description of Update Package in USC2. |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

## 8.950.2 Field Documentation

### 8.950.2.1 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bBinaryType

### 8.950.2.2 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bSerity

### 8.950.2.3 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bState

### 8.950.2.4 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::bUserInputRequest

### 8.950.2.5 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szDescription[1024]

#### 8.950.2.6 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szName[128]

#### 8.950.2.7 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szVersion[128]

#### 8.950.2.8 uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::TlvPresent

#### 8.950.2.9 uint32_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::ulPkgDownloadComplete

#### 8.950.2.10 uint32_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::ulPkgDownloadSize

#### 8.950.2.11 uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wDescriptionLength

#### 8.950.2.12 uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wNameLength

#### 8.950.2.13 uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wUpdateCompeteStatus

#### 8.950.2.14 uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wUserInputTimeout

#### 8.950.2.15 uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wVersionLength

## 8.951 UnpackSwiAvmsEventReportConfig Struct Reference

### Data Fields

- uint8_t bState
- uint8_t bUserInputRequest
- uint16_t wUserInputTimeout
- uint16_t wAlertMsgLength
- uint8_t szAlertMsg [200]
- uint8_t TlvPresent

### 8.951.1 Detailed Description

This structure contains unpack event report configure parameters.

**Parameters**

| | |
|---|---|
| *bState* | State.<br>• 0x01 - AVMS Read Request.<br>• 0x02 - AVMS Change Request.<br>• 0x03 - AVMS Config Complete. |
| *bUserInput-Request* | Bit mask of available user inputs.<br>• 0x00 - No user input required. Informational indication.<br>• 0x01 - Accept.<br>• 0x02 - Reject. |
| *wUserInput-Timeout* | Timeout for user input in seconds. A value of 0 means no time-out |
| *wAlertMsg-Length* | Length of Alert message string in bytes. |
| *szAlertMsg* | Alert message in UCS2. |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.951.2 Field Documentation

#### 8.951.2.1 uint8_t UnpackSwiAvmsEventReportConfig::bState

#### 8.951.2.2 uint8_t UnpackSwiAvmsEventReportConfig::bUserInputRequest

#### 8.951.2.3 uint8_t UnpackSwiAvmsEventReportConfig::szAlertMsg[200]

#### 8.951.2.4 uint8_t UnpackSwiAvmsEventReportConfig::TlvPresent

#### 8.951.2.5 uint16_t UnpackSwiAvmsEventReportConfig::wAlertMsgLength

#### 8.951.2.6 uint16_t UnpackSwiAvmsEventReportConfig::wUserInputTimeout

## 8.952 UnpackSwiAvmsEventReportConnectionRequest Struct Reference

### Data Fields

- uint8_t bUserInputRequest
- uint16_t wUserInputTimeout
- uint8_t TlvPresent

### 8.952.1 Detailed Description

This structure contains unpack event report connection request parameters.

**Parameters**

| | |
|---|---|
| *bUserInput-Request* | Bit mask of available user inputs.<br>• 0x00 - No user input required. Informational indication.<br>• 0x01 - Accept.<br>• 0x02 - Reject. |
| *bUserInput-Request* | Timeout for user input in minutes. A value of 0 means no time-out. |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.952.2 Field Documentation

#### 8.952.2.1 uint8_t UnpackSwiAvmsEventReportConnectionRequest::bUserInputRequest

#### 8.952.2.2 uint8_t UnpackSwiAvmsEventReportConnectionRequest::TlvPresent

#### 8.952.2.3 uint16_t UnpackSwiAvmsEventReportConnectionRequest::wUserInputTimeout

## 8.953 UnpackSwiAvmsEventReportDataSessionStatus Struct Reference

### Data Fields

- uint8_t bType
- uint16_t wErrorCode
- uint8_t TlvPresent

### 8.953.1 Detailed Description

This structure contains unpack evnet report data session status parameters.

**Parameters**

| | |
|---|---|
| *bType* | Notification type.<br>• 0: Data session closed.<br>• 1: Data session activated.<br>• 2: Register made. -3: Data session error. |
| *wErrorCode* | LWM2M Session error code.<br>• 0x0000: none |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.953.2 Field Documentation

#### 8.953.2.1 uint8_t UnpackSwiAvmsEventReportDataSessionStatus::bType

#### 8.953.2.2 uint8_t UnpackSwiAvmsEventReportDataSessionStatus::TlvPresent

#### 8.953.2.3 uint16_t UnpackSwiAvmsEventReportDataSessionStatus::wErrorCode

## 8.954 UnpackSwiAvmsEventReportHTTPStatus Struct Reference

### Data Fields

- uint16_t wHTTPStatus
- uint8_t TlvPresent

### 8.954.1 Detailed Description

This structure contains unpack event report HTTP status parameters.

**Parameters**

| | |
|---|---|
| *wHTTPStatus* | See RFC 7231.<br>• 100 - Continue<br>• 101 - Switching Protocols<br>• 200 - OK<br>• 201 - Created<br>• 202 - Accepted<br>• 203 - Non-Authoritative Information<br>• 204 - No Content<br>• 205 - Reset Content<br>• 206 - Partial Content<br>• 300 - Multiple Choices<br>• 301 - Moved Permanently<br>• 302 - Found<br>• 303 - See Other<br>• 304 - Not Modified<br>• 305 - Use Proxy<br>• 307 - Temporary Redirect<br>• 400 - Bad Request<br>• 401 - Unauthorized<br>• 402 - Payment Required<br>• 403 - Forbidden<br>• 404 - Not Found<br>• 405 - Method Not Allowed<br>• 406 - Not Acceptable<br>• 407 - Proxy Authentication Required<br>• 408 - Request Timeout<br>• 409 - Conflict<br>• 410 - Gone<br>• 411 - Length Required<br>• 412 - Precondition Failed<br>• 413 - Payload Too Large<br>• 414 - URI Too Long<br>• 415 - Unsupported Media Type<br>• 416 - Range Not Satisfiable<br>• 417 - Expectation Failed<br>• 426 - Upgrade Required<br>• 500 - Internal Server Error<br>• 501 - Not Implemented<br>• 502 - Bad Gateway<br>• 503 - Service Unavailable<br>• 504 - Gateway Timeout<br>• 505 - HTTP Version Not Supported |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

## 8.954.2 Field Documentation

### 8.954.2.1 uint8_t UnpackSwiAvmsEventReportHTTPStatus::TlvPresent

#### 8.954.2.2 uint16_t UnpackSwiAvmsEventReportHTTPStatus::wHTTPStatus

## 8.955 UnpackSwiAvmsEventReportNotification Struct Reference

### Data Fields

- uint8_t bNotification
- uint16_t wSessionStatus
- uint8_t TlvPresent

### 8.955.1 Detailed Description

This structure contains unpack event report notification parameters.

**Parameters**

| | |
|---|---|
| *bNotification* | Notification.<br>• 0x14 - Module starts sending data to server.<br>• 0x15 - Authentication with the server.<br>• 0x16 - session with the server is ended. |
| *wSessionStatus* | This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications.<br>• 0x0000: Successful: Session succeeded<br>• 0x0001: Break: Session succeeded<br>• 0x0002: Large Object Handled: Session succeeded<br>• 0x0003: No status: Session succeeded<br>• 0x0004: No more commands: Session succeeded<br>• 0x0005: User cancel: Session cancelled<br>• 0x0100-0x01FF: General errors<br>• 0x0200-0x02FF: Syncml errors<br>• 0x0300-0x03FF: Authentication errors<br>• 0x0400-0x04FF: Protocol errors<br>• 0x0500-0x05FF: Tree errors (DM Only)<br>• 0x0600-0x06FF: Not applicable<br>• 0x0700-0x07FF: Trigger errors<br>• 0x0800-0x08FF: FUMO errors<br>• 0x0900-0x09FF: Communication errors<br>• 0x0A00-0x0AFF: Parsing errors<br>• 0x0B00-0x0CFF: Not applicable<br>• 0x7F00-0x7F12:Insignia errors<br>• 0x7F13: Illegal text: Text received contains illegal characters<br>• 0x7F14: Download failure: Failed to download FOTA image<br>• 0x7F15: Empty session: Session ran successfully, but no information wasa updated<br>• 0x7F16: Factory reset successful: Factory reset succeeded<br>• 0x7F17: Factory reset fail: Factory reset failed |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.955.2 Field Documentation

#### 8.955.2.1 uint8_t UnpackSwiAvmsEventReportNotification::bNotification

#### 8.955.2.2 uint8_t UnpackSwiAvmsEventReportNotification::TlvPresent

#### 8.955.2.3 uint16_t UnpackSwiAvmsEventReportNotification::wSessionStatus

## 8.956 UnpackSwiAvmsEventReportPackageID Struct Reference

### Data Fields

- uint8_t bPackageID
- uint8_t TlvPresent

### 8.956.1 Detailed Description

This structure contains unpack event report package ID parameters.

**Parameters**

| | |
|---|---|
| *bPackageID* | Package ID of the application binary that this AVMS_EVENT_ID notification is for. |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.956.2 Field Documentation

#### 8.956.2.1 uint8_t UnpackSwiAvmsEventReportPackageID::bPackageID

#### 8.956.2.2 uint8_t UnpackSwiAvmsEventReportPackageID::TlvPresent

## 8.957 UnpackSwiAvmsEventReportRegStatus Struct Reference

### Data Fields

- uint8_t bRegStatus
- uint8_t TlvPresent

### 8.957.1 Detailed Description

This structure contains unpack event report registration status parameters.

**Parameters**

| | |
|---|---|
| *bRegStatus* | LWM2M Registration status.<br>• 0: Need Bootstrap.<br>• 1: Bootstrap made.<br>• 2: Register made. -3: Update made. |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.957.2 Field Documentation

#### 8.957.2.1 uint8_t UnpackSwiAvmsEventReportRegStatus::bRegStatus

#### 8.957.2.2 uint8_t UnpackSwiAvmsEventReportRegStatus::TlvPresent

## 8.958 UnpackSwiAvmsEventReportSessionType Struct Reference

### Data Fields

- uint8_t bType
- uint8_t TlvPresent

### 8.958.1 Detailed Description

This structure contains unpack event report session type parameters.

**Parameters**

| | |
|---|---|
| *bType* | Session Type.<br>• 0: Bootstrap session.<br>• 1. DM session |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.958.2 Field Documentation

#### 8.958.2.1 uint8_t UnpackSwiAvmsEventReportSessionType::bType

#### 8.958.2.2 uint8_t UnpackSwiAvmsEventReportSessionType::TlvPresent

## 8.959 UnpackSwiAvmsEventReportWAMSParamChange Struct Reference

### Data Fields

- uint16_t wWamsChangeMask
- uint8_t TlvPresent

### 8.959.1 Detailed Description

This structure contains unpack event report WAMS parameter change parameters.

**Parameters**

| | |
|---|---|
| *wWamsChange-Mask* | Mask of WAMS parameters changed.<br>• By default set to 0xFF for all changes.<br>• 0x01 - device_login<br>• 0x02 - device_MD5_key<br>• 0x04 - server_login<br>• 0x08 - server_MD5_key<br>• 0x10 - server_URL<br>• 0x20 - Nonce<br>• 0x40 - Application key |
| *TlvPresent* | Boolean indicating the presence of the TLV in the QMI response |

### 8.959.2 Field Documentation

#### 8.959.2.1 uint8_t UnpackSwiAvmsEventReportWAMSParamChange::TlvPresent

#### 8.959.2.2 uint16_t UnpackSwiAvmsEventReportWAMSParamChange::wWamsChangeMask

## 8.960 unpackWdsProfileParam Union Reference

### Data Fields

- LibpackProfile3GPP SlqsProfile3GPP
- LibpackProfile3GPP2 SlqsProfile3GPP2

### 8.960.1 Detailed Description

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

**Parameters**

| | |
|---|---|
| *SlqsProfile3GPP* | 3GPP profile<br>• See LibpackProfile3GPP |
| *SlqsProfile3GP-P2* | 3GPP2 profile<br>• See LibpackProfile3GPP2 |

### 8.960.2 Field Documentation

#### 8.960.2.1 LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP

#### 8.960.2.2 LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2

## 8.961 voice_airTimer Struct Reference

### Data Fields

- uint8_t namID
- uint32_t airTimerValue

### 8.961.1 Detailed Description

This structure contains information about the Air Timer.

**Parameters**

| | |
|---|---|
| *namID* | • Index of the NAM(Number Assignment Module) to be configured.<br>• Range 0 to 3.<br>• Some modems support only 1 or 2 NAMs.<br>• 0xFF,if not available. |

| | |
|---|---|
| *airTimerValue* | • Time in minutes.<br>• Cumulative air time is slammed.<br>• 0xFFFFFFFF,if not available. |

### 8.961.2 Field Documentation

#### 8.961.2.1 uint32_t voice_airTimer::airTimerValue

#### 8.961.2.2 uint8_t voice_airTimer::namID

## 8.962 voice_allCallsAlphaIDInfo Struct Reference

### Data Fields

- uint8_t callID
- voice_alphaIDInfo AlphaIDInfo

### 8.962.1 Detailed Description

This structure contains information for Alpha Identifier for All Calls

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *AlphaIDInfo* | • See voice_alphaIDInfo for more information. |

### 8.962.2 Field Documentation

#### 8.962.2.1 voice_alphaIDInfo voice_allCallsAlphaIDInfo::AlphaIDInfo

#### 8.962.2.2 uint8_t voice_allCallsAlphaIDInfo::callID

## 8.963 voice_allCallsDiagInfo Struct Reference

### Data Fields

- uint8_t callID
- voice_diagInfo DiagInfo

### 8.963.1 Detailed Description

This structure contains Diagnostic Information for All Calls

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *DiagInfo* | • See voice_diagInfo for more information. |

### 8.963.2 Field Documentation

#### 8.963.2.1 uint8_t voice_allCallsDiagInfo::callID

#### 8.963.2.2 voice_diagInfo voice_allCallsDiagInfo::DiagInfo

## 8.964 voice_allCallsUUSInfo Struct Reference

### Data Fields

- uint8_t callID
- voice_UUSInfo uusInfo

### 8.964.1 Detailed Description

This structure contains information for User to User Signaling Service for All Calls.

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *uusInfo* | • See voice_UUSInfo for more information. |

### 8.964.2 Field Documentation

#### 8.964.2.1 uint8_t voice_allCallsUUSInfo::callID

#### 8.964.2.2 voice_UUSInfo voice_allCallsUUSInfo::uusInfo

## 8.965 voice_alphaIDInfo Struct Reference

### Data Fields

- uint8_t alphaDcs
- uint8_t alphaLen
- uint8_t alphaText [255]

### 8.965.1 Detailed Description

This structure contains information about the Alpha Identifier.

**Parameters**

| | |
|---|---|
| *alphaDcs* | • Alpha coding scheme<br>– 0x01 - GSM Default_Char<br>– 0x02 - UCS2<br>– 0xFF - Not Available |
| *alphaLen* | • Number of sets of the following elements:<br>– pAlpha_text<br>• If zero(0) then no further information exists. |
| *alphaText[MAX_-VOICE_DESCR-IPTION_LENGT-H]* | • Data encoded as per the alpha_dcs |

### 8.965.2 Field Documentation

#### 8.965.2.1 uint8_t voice_alphaIDInfo::alphaDcs

#### 8.965.2.2 uint8_t voice_alphaIDInfo::alphaLen

#### 8.965.2.3 uint8_t voice_alphaIDInfo::alphaText[255]

## 8.966 voice_arrAlertingPattern Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint32_t alertingPattern [20]

### 8.966.1 Detailed Description

This structure contains an array of Alerting Pattern.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of callID, alertingPattern that follow.<br>• If zero(0) then no further information exists. |
| *callID[VOICE_-MAX_NO_OF_-CALLS]* | • Array of Unique call identifier for the call. |
| *alertingPattern[-VOICE_MAX_N-O_OF_CALLS]* | • Array of Alerting pattern.<br>– 0x00 - QMI_VOICE_ALERTING_PATTERN_1 - Pattern 1<br>– 0x01 - QMI_VOICE_ALERTING_PATTERN_2 - Pattern 2<br>– 0x02 - QMI_VOICE_ALERTING_PATTERN_3 - Pattern 3<br>– 0x04 - QMI_VOICE_ALERTING_PATTERN_5 - Pattern 5<br>– 0x05 - QMI_VOICE_ALERTING_PATTERN_6 - Pattern 6<br>– 0x06 - QMI_VOICE_ALERTING_PATTERN_7 - Pattern 7<br>– 0x07 - QMI_VOICE_ALERTING_PATTERN_8 - Pattern 8<br>– 0x08 - QMI_VOICE_ALERTING_PATTERN_9 - Pattern 9 |

### 8.966.2 Field Documentation

#### 8.966.2.1 uint32_t voice_arrAlertingPattern::alertingPattern[20]

#### 8.966.2.2 uint8_t voice_arrAlertingPattern::callID[20]

#### 8.966.2.3 uint8_t voice_arrAlertingPattern::numInstances

## 8.967 voice_arrAlertingType Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint8_t AlertingType [20]

### 8.967.1 Detailed Description

This structure contains an array of Alerting Type.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of callID, AlertingType that follow.<br>• If zero(0) then no further information exists. |
| *callID[VOICE_-MAX_NO_OF_-CALLS]* | • Array of Unique call identifier for the call. |

| | |
|---|---|
| *AlertingType[VOICE_MAX_NO_OF_CALLS]* | • Array of Alerting type.<br>– 0x00 - ALERTING_LOCAL - Local<br>– 0x01 - ALERTING_REMOTE - Remote |

### 8.967.2 Field Documentation

#### 8.967.2.1 uint8_t voice_arrAlertingType::AlertingType[20]

#### 8.967.2.2 uint8_t voice_arrAlertingType::callID[20]

#### 8.967.2.3 uint8_t voice_arrAlertingType::numInstances

## 8.968 voice_arrAlphaID Struct Reference

### Data Fields

- uint8_t numInstances
- voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]

### 8.968.1 Detailed Description

This structure contains an array of Alpha ID Info

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of allCallsAlphaIDInfo that follow.<br>• If zero(0) then no further information exists. |
| *allCallsAlphaIDInfo[VOICE_MAX_NO_OF_CALLS]* | • Array of allCallsAlphaIDInfo.<br>• See voice_allCallsAlphaIDInfo for more information. |

### 8.968.2 Field Documentation

#### 8.968.2.1 voice_allCallsAlphaIDInfo voice_arrAlphaID::allCallsAlphaIDInfoArr[20]

#### 8.968.2.2 uint8_t voice_arrAlphaID::numInstances

## 8.969 voice_arrCalledPartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_peerNumberInfo CalledPartyNum [20]

### 8.969.1 Detailed Description

This structure contains an array of Called Party Numbers consisting of information of all the numbers which have been called from the device.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of calledPartyNum that follow.<br>• If zero(0) then no further information exists. |
| *CalledPartyNum[VOICE_MAX_NO_OF_CALLS]* | • Array of CalledPartyNum.<br>• See voice_peerNumberInfo for more information. |

### 8.969.2 Field Documentation

#### 8.969.2.1 voice_peerNumberInfo voice_arrCalledPartyNum::CalledPartyNum[20]

#### 8.969.2.2 uint8_t voice_arrCalledPartyNum::numInstances

## 8.970 voice_arrCallEndReason Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t callEndReason [20]

### 8.970.1 Detailed Description

This structure contains an array of Call End Reasons.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of callID, callEndReason that follow.<br>• If zero(0) then no further information exists. |
| *callID[VOICE_MAX_NO_OF_CALLS]* | • Array of Unique call identifier for the call. |
| *callEndReason[VOICE_MAX_NO_OF_CALLS]* | • Array of Call End Reason .<br>• See Table9 qaGobiApiTableVoiceCallEndReasons.h for a list of valid voice-related call end reasons |

### 8.970.2 Field Documentation

#### 8.970.2.1 uint16_t voice_arrCallEndReason::callEndReason[20]

#### 8.970.2.2 uint8_t voice_arrCallEndReason::callID[20]

#### 8.970.2.3 uint8_t voice_arrCallEndReason::numInstances

## 8.971 voice_arrCallInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_getAllCallInformation getAllCallInfo [20]

### 8.971.1 Detailed Description

This structure contains an array of Call Info

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of getAllCallInfo that follow.<br>• If zero(0) then no further information exists. |
| *getAllCallInfo[VOICE_MAX_NO_OF_CALLS]* | • Array of CallInfo.<br>• See voice_getAllCallInformation for more information. |

### 8.971.2 Field Documentation

#### 8.971.2.1 voice_getAllCallInformation voice_arrCallInfo::getAllCallInfo[20]

#### 8.971.2.2 uint8_t voice_arrCallInfo::numInstances

## 8.972 voice_arrConnectPartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_peerNumberInfo ConnectedPartyNum [20]

### 8.972.1 Detailed Description

This structure contains an array of Connected Party Numbers consisting of information regarding all the devices connected.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of ConnectedPartyNum that follow.<br>• If zero(0) then no further information exists. |
| *ConnectedPartyNum[VOICE_MAX_NO_OF_CALLS]* | • Array of ConnectedPartyNum.<br>• See voice_peerNumberInfo for more information. |

### 8.972.2 Field Documentation

#### 8.972.2.1 voice_peerNumberInfo voice_arrConnectPartyNum::ConnectedPartyNum[20]

#### 8.972.2.2 uint8_t voice_arrConnectPartyNum::numInstances

## 8.973 voice_arrDiagInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_allCallsDiagInfo DiagInfo [20]

### 8.973.1 Detailed Description

This structure contains an array of Diagnostic Information.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of DiagInfo that follow.<br>• If zero(0) then no further information exists. |
| *DiagInfo[VOICE_MAX_NO_OF_CALLS]* | • Array of DiagInfo.<br>• See voice_allCallsDiagInfo for more information. |

### 8.973.2 Field Documentation

#### 8.973.2.1 voice_allCallsDiagInfo voice_arrDiagInfo::DiagInfo[20]

#### 8.973.2.2 uint8_t voice_arrDiagInfo::numInstances

## 8.974 voice_arrRedirPartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_peerNumberInfo RedirPartyNum [20]

### 8.974.1 Detailed Description

This structure contains an array of Redirecting Party Numbers consisting of information of all the numbers which have been redirected from the device.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of redirPartyNum that follow.<br>• If zero(0) then no further information exists. |

| | |
|---|---|
| *RedirPartyNum[-VOICE_MAX_N-O_OF_CALLS]* | • Array of RedirPartyNum.<br>• See voice_peerNumberInfo for more information. |

### 8.974.2 Field Documentation

#### 8.974.2.1 uint8_t voice_arrRedirPartyNum::numInstances

#### 8.974.2.2 voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]

## 8.975 voice_arrRemotePartyName Struct Reference

### Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyName GetAllCallRmtPtyName [20]

### 8.975.1 Detailed Description

This structure contains an array of Remote Party Names

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of remotePartyName that follow.<br>• If zero(0) then no further information exists. |
| *GetAllCallRmt-PtyName[MAX_-NO_OF_CALLS]* | • Array of remotePartyName.<br>• See voice_getAllCallRmtPtyName for more information. |

### 8.975.2 Field Documentation

#### 8.975.2.1 voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]

#### 8.975.2.2 uint8_t voice_arrRemotePartyName::numInstances

## 8.976 voice_arrRemotePartyNum Struct Reference

### Data Fields

- uint8_t numInstances
- voice_getAllCallRmtPtyNum RmtPtyNum [20]

### 8.976.1 Detailed Description

This structure contains an array of Remote Party Numbers

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of remotePartyNum that follow.<br>• If zero(0) then no further information exists. |
| *RmtPtyNum[VOICE_MAX_NO_OF_CALLS]* | • Array of remotePartyNum.<br>• See voice_getAllCallRmtPtyNum for more information. |

### 8.976.2 Field Documentation

#### 8.976.2.1 uint8_t voice_arrRemotePartyNum::numInstances

#### 8.976.2.2 voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]

## 8.977 voice_arrSvcOption Struct Reference

### Data Fields

- uint8_t numInstances
- uint8_t callID [20]
- uint16_t srvOption [20]

### 8.977.1 Detailed Description

This structure contains array an of Servicing option.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of callID, srvOption that follow.<br>• If zero(0) then no further information exists. |
| *callID[VOICE_MAX_NO_OF_CALLS]* | • Array of Unique call identifier for the call. |
| *srvOption[VOICE_MAX_NO_OF_CALLS]* | • Array of Service option.<br>• See Table9 qaGobiApiTableServiceOptions.h for standard service option number assignments. |

### 8.977.2 Field Documentation

#### 8.977.2.1 uint8_t voice_arrSvcOption::callID[20]

#### 8.977.2.2 uint8_t voice_arrSvcOption::numInstances

#### 8.977.2.3 uint16_t voice_arrSvcOption::srvOption[20]

## 8.978 voice_arrUUSInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_allCallsUUSInfo AllCallsUUSInfo [20]

### 8.978.1 Detailed Description

This structure contains an array of User to User Signaling Service Information

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of allCallsUUSInfo that follow.<br>• If zero(0) then no further information exists. |
| *AllCallsUUSInfo[VOICE_MAX_NO_OF_CALLS]* | • Array of allCallsUUSInfo.<br>• See voice_allCallsUUSInfo for more information. |

### 8.978.2 Field Documentation

#### 8.978.2.1 voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]

#### 8.978.2.2 uint8_t voice_arrUUSInfo::numInstances

## 8.979 voice_burstDTMFInfo Struct Reference

### Data Fields

- uint8_t * pCallID
- uint8_t digitCnt
- uint8_t pDigitBuff [255]

### 8.979.1 Detailed Description

This structure contains Voice Burst DTMF Information

**Parameters**

| | |
|---|---|
| *pCallID* | • Call ID associated with call on which the DTMF information has to be sent.<br>A burst DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.<br>• This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.<br>• If the call ID value received is 0, no value has been returned by the device<br>• NULL pointer - Invalid data. |
| *digitCnt* | • Length of DTMF digit buffer which follows |
| *pDigitBuff[MAX_VOICE_DESCRIPTION_LENGTH]* | • DTMF digit buffer in ASCII, NULL terminated |

### 8.979.2 Field Documentation

#### 8.979.2.1 uint8_t voice_burstDTMFInfo::digitCnt

#### 8.979.2.2 uint8_t* voice_burstDTMFInfo::pCallID

#### 8.979.2.3 uint8_t voice_burstDTMFInfo::pDigitBuff[255]

## 8.980 voice_calledPartyInfo Struct Reference

### Data Fields

- uint8_t PI
- uint8_t SI
- uint8_t numType
- uint8_t numPlan
- uint8_t numLen
- uint8_t number [255]

### 8.980.1 Detailed Description

This structure contains Called party Number Information

**Parameters**

| | |
|---|---|
| *PI* | • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values. |
| *SI* | • Number of sets of following elements<br>– Caller Id |
| *SI* | • Number screening indicator.<br>• Values:<br>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened<br>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification<br>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification<br>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network |

| | |
|---|---|
| *numType* | • Number type.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension |
| *numPlan* | • Number plan.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension |
| *numLen* | • Provides the length of number which follow. |
| *number[255]* | • number of numLen length, NULL terminated. |

### 8.980.2 Field Documentation

#### 8.980.2.1 uint8_t voice_calledPartyInfo::number[255]

#### 8.980.2.2 uint8_t voice_calledPartyInfo::numLen

#### 8.980.2.3 uint8_t voice_calledPartyInfo::numPlan

#### 8.980.2.4 uint8_t voice_calledPartyInfo::numType

#### 8.980.2.5 uint8_t voice_calledPartyInfo::PI

#### 8.980.2.6 uint8_t voice_calledPartyInfo::SI

## 8.981 voice_calledPartySubAdd Struct Reference

### Data Fields

- uint8_t extBit

- uint8_t subAddrType
- uint8_t oddEvenInd
- uint8_t subAddrLen
- uint8_t subAddr [255]

### 8.981.1 Detailed Description

This structure contains information about the Called Sub Party Addresses.

**Parameters**

| | |
|---|---|
| *extBit* | • Extension bit. |
| *subAddrType* | • Subaddress type.<br>– 0x00 - NSAP<br>– 0x01 - USER |
| *oddEvenInd* | • Even/odd indicator.<br>– 0x00 - Even number of address signals<br>– 0x01 - Odd number of address signals |
| *subAddrLen* | • Number of sets of the following elements:<br>– SubAddress |
| *subAddr[MAX_VOICE_DESCRIPTION_LENGTH]* | • Array of the SubAddress in BCD number format. |

### 8.981.2 Field Documentation

#### 8.981.2.1 uint8_t voice_calledPartySubAdd::extBit

#### 8.981.2.2 uint8_t voice_calledPartySubAdd::oddEvenInd

#### 8.981.2.3 uint8_t voice_calledPartySubAdd::subAddr[255]

#### 8.981.2.4 uint8_t voice_calledPartySubAdd::subAddrLen

#### 8.981.2.5 uint8_t voice_calledPartySubAdd::subAddrType

## 8.982 voice_callerIDInfo Struct Reference

### Data Fields

- uint8_t PI
- uint8_t callerIDLen
- uint8_t callerID [255]

### 8.982.1 Detailed Description

This structure contains Caller ID Information

**Parameters**

| | |
|---|---|
| *PI* | • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values. |
| *callerIDLen* | • Number of sets of following elements<br>– Caller Id |
| *pCallerID* | • Caller ID in ASCII string. |

### 8.982.2 Field Documentation

#### 8.982.2.1 uint8_t voice_callerIDInfo::callerID[255]

#### 8.982.2.2 uint8_t voice_callerIDInfo::callerIDLen

#### 8.982.2.3 uint8_t voice_callerIDInfo::PI

## 8.983 voice_callFwdTypeAndPlan Struct Reference

### Data Fields

- uint8_t numberType
- uint8_t numberPlan

### 8.983.1 Detailed Description

This structure contains Supplementary Service request parameters related to different features and their activation, deactivation, registration and erasure (applicable only for 3GPP)

**Parameters**

| | |
|---|---|
| *numberType* | • Call forwarding number type<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION Reserved extension |

| | |
|---|---|
| *numberPlan* | • Call forwarding number plan<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION Reserved extension |

### 8.983.2 Field Documentation

#### 8.983.2.1 uint8_t voice_callFwdTypeAndPlan::numberPlan

#### 8.983.2.2 uint8_t voice_callFwdTypeAndPlan::numberType

## 8.984 voice_callFWExtInfo Struct Reference

### Data Fields

- uint8_t SvcStatus
- uint8_t SvcClass
- uint8_t noReplyTimer
- uint8_t PI
- uint8_t SI
- uint8_t numType
- uint8_t numPlan
- uint8_t numLen
- uint8_t number [255]

### 8.984.1 Detailed Description

This structure contains information for Get Call Forwarding Extended Information.

**Parameters**

| | |
|---|---|
| *SvcStatus* | • Service status. Values:<br>– 0x00 - SERVICE_STATUS_INACTIVE - Inactive<br>– 0x01 - SERVICE_STATUS_ACTIVE - Active |
| *SvcClass* | • Service Class is a combination (sum) of information class constants<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes. |
| *noReplyTimer* | • No reply timer value in seconds<br>• A value of 0 indicates that no_reply_timer is ignored. |

| | |
|---|---|
| *PI* | • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values. |
| *SI* | • Number screening indicator.<br>• Values:<br>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened<br>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification<br>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification<br>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network |
| *numType* | • Number type.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension |
| *numPlan* | • Number plan.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension |
| *numLen* | • Provides the length of number which follow. |
| *number[255]* | • number of numLen length, NULL terminated. |

## 8.984.2 Field Documentation

### 8.984.2.1 uint8_t voice_callFWExtInfo::noReplyTimer

### 8.984.2.2 uint8_t voice_callFWExtInfo::number[255]

#### 8.984.2.3 uint8_t voice_callFWExtInfo::numLen

#### 8.984.2.4 uint8_t voice_callFWExtInfo::numPlan

#### 8.984.2.5 uint8_t voice_callFWExtInfo::numType

#### 8.984.2.6 uint8_t voice_callFWExtInfo::PI

#### 8.984.2.7 uint8_t voice_callFWExtInfo::SI

#### 8.984.2.8 uint8_t voice_callFWExtInfo::SvcClass

#### 8.984.2.9 uint8_t voice_callFWExtInfo::SvcStatus

## 8.985 voice_callFWInfo Struct Reference

### Data Fields

- uint8_t SvcStatus
- uint8_t SvcClass
- uint8_t numLen
- uint8_t number [255]
- uint8_t noReplyTimer

### 8.985.1 Detailed Description

This structure contains information for Get Call Forwarding Information.

**Parameters**

| | |
|---|---|
| *SvcStatus* | • Service status. Values:<br>– 0x00 - SERVICE_STATUS_INACTIVE - Inactive<br>– 0x01 - SERVICE_STATUS_ACTIVE - Active |
| *SvcClass* | • Service Class is a combination (sum) of information class constants<br>• See qaGobiApiTableSupServiceInfoClasses.h for service classes. |
| *numLen* | • Provides the length of number which follow. |
| *number[255]* | • number of numLen length, NULL terminated. |
| *noReplyTimer* | • No reply timer value in seconds<br>• A value of 0 indicates that noReplyTimer is ignored. |

### 8.985.2 Field Documentation

#### 8.985.2.1 uint8_t voice_callFWInfo::noReplyTimer

#### 8.985.2.2 uint8_t voice_callFWInfo::number[255]

#### 8.985.2.3 uint8_t voice_callFWInfo::numLen

#### 8.985.2.4 uint8_t voice_callFWInfo::SvcClass

#### 8.985.2.5 uint8_t voice_callFWInfo::SvcStatus

## 8.986 voice_callInfo Struct Reference

### Data Fields

- uint8_t callID
- uint8_t callState
- uint8_t callType
- uint8_t direction
- uint8_t mode

### 8.986.1 Detailed Description

This structure contains Information about call state changes. For example, when an incoming call is received, this structure is populated and indicate the incoming call information. When this incoming call is answered, the call status changes from INCOMING to CONVERSATION, which means a change in the call information and this structure is populated again with the changes and notified to/retrived by the user.

**Parameters**

| | |
|---|---|
| *callID* | • Call identifier for the call queried for information.<br>• If zero(0) then invalid. |
| *callState* | • Call state.<br>– 0x01 - CALL_STATE_ORIGINATION - Origination<br>– 0x02 - CALL_STATE_INCOMING - Incoming<br>– 0x03 - CALL_STATE_CONVERSATION - Conversation<br>– 0x04 - CALL_STATE_CC_IN_PROGRESS - Call is originating but waiting for call control to complete<br>– 0x05 - CALL_STATE_ALERTING - Alerting<br>– 0x06 - CALL_STATE_HOLD - Hold<br>– 0x07 - CALL_STATE_WAITING - Waiting<br>– 0x08 - CALL_STATE_DISCONNECTING - Disconnecting<br>– 0x09 - CALL_STATE_END - End<br>– 0x0A - CALL_STATE_SETUP - MT call is in Setup state in 3GPP<br>– 0xFF - Not Available |

| | |
|---|---|
| *callType* | • Call type.<br>– 0x00 - CALL_TYPE_VOICE - Voice<br>– 0x02 - CALL_TYPE_VOICE_IP - Voice over IP<br>– 0x06 - CALL_TYPE_OTAPA - OTAPA<br>– 0x07 - CALL_TYPE_STD_OTASP - Standard OTASP<br>– 0x08 - CALL_TYPE_NON_STD_OTASP - Nonstandard OTASP<br>– 0x09 - CALL_TYPE_EMERGENCY - Emergency<br>– 0xFF - Not Available |
| *direction* | • Direction.<br>– 0x01 - CALL_DIRECTION_MO - MO call<br>– 0x02 - CALL_DIRECTION_MT - MT call<br>– 0xFF - Not Available |
| *mode* | • Mode.<br>• If the mode field is "0x01 - CDMA",the optional Service Option, Voice Privacy, and OTASP Status (only for OTASP calls) TLVs are included in the response.<br>– 0x01 - CALL_MODE_CDMA - CDMA<br>– 0x02 - CALL_MODE_GSM - GSM<br>– 0x03 - CALL_MODE_UMTS - UMTS<br>– 0x04 - CALL_MODE_LTE - LTE<br>– 0x05 - CALL_MODE_TDS - TD-SCDMA<br>– 0xFF - Not Available |

### 8.986.2 Field Documentation

#### 8.986.2.1 uint8_t voice_callInfo::callID

#### 8.986.2.2 uint8_t voice_callInfo::callState

#### 8.986.2.3 uint8_t voice_callInfo::callType

#### 8.986.2.4 uint8_t voice_callInfo::direction

#### 8.986.2.5 uint8_t voice_callInfo::mode

## 8.987 voice_callingPartyInfo Struct Reference

### Data Fields

- uint8_t PI
- uint8_t SI
- uint8_t numType
- uint8_t numPlan
- uint8_t numLen
- uint8_t number [255]

### 8.987.1 Detailed Description

This structure contains Calling party Number Information

**Parameters**

| | |
|---|---|
| *PI* | • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values. |
| *SI* | • Number of sets of following elements<br>– Caller Id |
| *SI* | • Number screening indicator.<br>• Values:<br>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened<br>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification<br>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification<br>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network |
| *numType* | • Number type.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension |
| *numPlan* | • Number plan.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension |
| *numLen* | • Provides the length of number which follow. |

| | |
|---|---|
| *number[255]* | • number of numLen length, NULL terminated. |

### 8.987.2 Field Documentation

#### 8.987.2.1 uint8_t voice_callingPartyInfo::number[255]

#### 8.987.2.2 uint8_t voice_callingPartyInfo::numLen

#### 8.987.2.3 uint8_t voice_callingPartyInfo::numPlan

#### 8.987.2.4 uint8_t voice_callingPartyInfo::numType

#### 8.987.2.5 uint8_t voice_callingPartyInfo::PI

#### 8.987.2.6 uint8_t voice_callingPartyInfo::SI

## 8.988 voice_ccSUPSType Struct Reference

### Data Fields

- uint8_t svcType
- uint8_t reason

### 8.988.1 Detailed Description

This structure contains information about the Call Control Supplementary Service Types

**Parameters**

| | |
|---|---|
| *svcType* | • Service type.<br>– 0x01 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ACTIVATE - Activate<br>– 0x02 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_DEACTIVATE - Deactivate<br>– 0x03 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER - Register<br>– 0x04 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_ERASE - Erase<br>– 0x05 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_INTERROGATE - Interrogate<br>– 0x06 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_REGISTER_PASSWORD - Register password<br>– 0x07 - VOICE_CC_SUPS_RESULT_SERVICE_TYPE_USSD - USSD<br>– 0xFF - Not Available |
| *reason* | • Call control supplementary service result reason<br>• Values:<br>– See qaGobiApiTableCallControlReturnReasons.h for return reasons. |

### 8.988.2 Field Documentation

#### 8.988.2.1 uint8_t voice_ccSUPSType::reason

#### 8.988.2.2 uint8_t voice_ccSUPSType::svcType

## 8.989 voice_CLIPResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.989.1 Detailed Description

This structure contains information about the Calling Line Identification Presentation (CLIP) supplementary service responses.

**Parameters**

| | |
|---|---|
| *ActiveStatus* | • Active status.<br>• Values:<br>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive<br>– 0x01 - ACTIVE_STATUS_ACTIVE - Active<br>– 0xFF - Not Available |
| *ProvisionStatus* | • Provisioned status.<br>• Values:<br>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned<br>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned<br>– 0xFF - Not Available |

### 8.989.2 Field Documentation

#### 8.989.2.1 uint8_t voice_CLIPResp::ActiveStatus

#### 8.989.2.2 uint8_t voice_CLIPResp::ProvisionStatus

## 8.990 voice_CLIRResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.990.1 Detailed Description

This structure contains information about the Calling Line Identification Restriction (CLIR) supplementary service responses.

**Parameters**

| | |
|---|---|
| *ActiveStatus* | • Active status.<br>• Values:<br>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive<br>– 0x01 - ACTIVE_STATUS_ACTIVE - Active<br>– 0xFF - Not Available |
| *ProvisionStatus* | • Provisioned status.<br>• Values:<br>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned<br>– 0x01 - PROVISION_STATUS_PROVISIONED_PERMANENT - Permanently provisioned<br>– 0x02 - PROVISION_STATUS_PRESENTATION_RESTRICTED - Restricted presentation<br>– 0x03 - PROVISION_STATUS_PRESENTATION_ALLOWED - Allowed presentation<br>– 0xFF - Not Available |

### 8.990.2 Field Documentation

#### 8.990.2.1 uint8_t voice_CLIRResp::ActiveStatus

#### 8.990.2.2 uint8_t voice_CLIRResp::ProvisionStatus

## 8.991 voice_CNAPResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.991.1 Detailed Description

This structure contains information about the Calling Name Presentation (CNAP) supplementary service responses.

**Parameters**

| | |
|---|---|
| *ActiveStatus* | • Active status.<br>• Values:<br>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive<br>– 0x01 - ACTIVE_STATUS_ACTIVE - Active<br>– 0xFF - Not Available |
| *ProvisionStatus* | • Provisioned status.<br>• Values:<br>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned<br>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned<br>– 0xFF - Not Available |

### 8.991.2 Field Documentation

#### 8.991.2.1 uint8_t voice_CNAPResp::ActiveStatus

#### 8.991.2.2 uint8_t voice_CNAPResp::ProvisionStatus

## 8.992 voice_COLPResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.992.1 Detailed Description

This structure contains information about the Connected Line Identification Presentation (COLP) supplementary service responses.

**Parameters**

| | |
|---|---|
| *ActiveStatus* | • Active status.<br>• Values:<br>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive<br>– 0x01 - ACTIVE_STATUS_ACTIVE - Active<br>– 0xFF - Not Available |
| *ProvisionStatus* | • Provisioned status.<br>• Values:<br>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned<br>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned<br>– 0xFF - Not Available |

### 8.992.2 Field Documentation

#### 8.992.2.1 uint8_t voice_COLPResp::ActiveStatus

#### 8.992.2.2 uint8_t voice_COLPResp::ProvisionStatus

## 8.993 voice_COLRResp Struct Reference

### Data Fields

- uint8_t ActiveStatus
- uint8_t ProvisionStatus

### 8.993.1 Detailed Description

This structure contains information about the Connected Line Identification Restriction (COLR) supplementary service responses.

**Parameters**

| | |
|---|---|
| *ActiveStatus* | • Active status.<br>• Values:<br>– 0x00 - ACTIVE_STATUS_INACTIVE - Inactive<br>– 0x01 - ACTIVE_STATUS_ACTIVE - Active<br>– 0xFF - Not Available |
| *ProvisionStatus* | • Provisioned status.<br>• Values:<br>– 0x00 - PROVISION_STATUS_NOT_PROVISIONED - Not provisioned<br>– 0x01 - PROVISION_STATUS_PROVISIONED - Provisioned<br>– 0xFF - Not Available |

### 8.993.2 Field Documentation

#### 8.993.2.1 uint8_t voice_COLRResp::ActiveStatus

#### 8.993.2.2 uint8_t voice_COLRResp::ProvisionStatus

## 8.994 voice_connectNumInfo Struct Reference

### Data Fields

- uint8_t numPresInd
- uint8_t screeningInd
- uint8_t numType
- uint8_t numPlan
- uint8_t callerIDLen
- uint8_t callerID [81]

### 8.994.1 Detailed Description

This structure contains information about the numbers connected to a device. It contains information such as number type, eg International or Local.

**Parameters**

| | |
|---|---|
| *numPresInd* | • Presentation indicator<br>– 0x00 - PRESENTATION_ALLOWED - Allowed presentation<br>– 0x01 - PRESENTATION_RESTRICTED - Restricted presentation<br>– 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation<br>– 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific)<br>– 0xFF - Not Available |

| | |
|---|---|
| *screeningInd* | • Screening indicator.<br>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened<br>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification<br>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification<br>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network<br>– 0xFF - Not Available |
| *numType* | • Number type.<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension<br>– 0xFF - Not Available |
| *numPlan* | • Number plan.<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension<br>– 0xFF - Not Available |
| *callerIDLen* | • Provides the length of caller ID which follow.<br>• If zero(0) then no further information exists. |
| *callerID[MAX_VOICE_CALL_NO_LEN]* | • callerID of numLen length, NULL terminated. |

## 8.994.2 Field Documentation

### 8.994.2.1 uint8_t voice_connectNumInfo::callerID[81]

### 8.994.2.2 uint8_t voice_connectNumInfo::callerIDLen

### 8.994.2.3 uint8_t voice_connectNumInfo::numPlan

#### 8.994.2.4 uint8_t voice_connectNumInfo::numPresInd

#### 8.994.2.5 uint8_t voice_connectNumInfo::numType

#### 8.994.2.6 uint8_t voice_connectNumInfo::screeningInd

## 8.995 voice_CUGInfo Struct Reference

### Data Fields

- uint16_t CUGIndex
- uint8_t SuppPrefCUG
- uint8_t SuppOA

### 8.995.1 Detailed Description

This structure contains Closed User Group Information

**Parameters**

| | |
|---|---|
| *CUGIndex* | • Range 0x00... 0x7FFF |
| *SuppPrefCUG* | • Suppress preferential CUG<br>– 0x00 - FALSE<br>– 0x01 - TRUE |
| *SuppOA* | • Suppress OA subscription option<br>– 0x00 - FALSE<br>– 0x01 - TRUE |

### 8.995.2 Field Documentation

#### 8.995.2.1 uint16_t voice_CUGInfo::CUGIndex

#### 8.995.2.2 uint8_t voice_CUGInfo::SuppOA

#### 8.995.2.3 uint8_t voice_CUGInfo::SuppPrefCUG

## 8.996 voice_curAMRConfig Struct Reference

### Data Fields

- uint8_t gsmAmrStat
- uint8_t wcdmaAmrStat

### 8.996.1 Detailed Description

This structure contains the Current Adaptive Multi Rate Configuration Information.

**Parameters**

| | |
|---|---|
| *gsmAmrStat* | • GSM AMR Status<br>– 0x00 - Disable<br>– 0x01 - Enable<br>– 0xFF - Not Available |
| *wcdmaAmrStat* | • WCDMA AMR Status<br>• One or a combination of the following bitmask values:<br>– Bit 0 - AMR codec advertised is not supported<br>– Bit 1 - Controls WCDMA AMR wideband<br>– Bit 2 - Controls GSM half rate AMR<br>– Bit 3 - Controls GSM AMR wideband<br>– Bit 4 - Controls GSM AMR narrowband<br>• 0xFF, if not available |

### 8.996.2 Field Documentation

#### 8.996.2.1 uint8_t voice_curAMRConfig::gsmAmrStat

#### 8.996.2.2 uint8_t voice_curAMRConfig::wcdmaAmrStat

## 8.997 voice_diagInfo Struct Reference

### Data Fields

- uint8_t diagInfoLen
- uint8_t diagnosticInfo [255]

### 8.997.1 Detailed Description

This structure contains Diagnostic Information

**Parameters**

| | |
|---|---|
| *diagInfoLen* | • Provides the length of information which follow.<br>• If zero(0) then no further information exists. |
| *diagnosticInfo[MAX_VOICE_DESCRIPTION_LENGTH]* | • Diagnostic information. |

### 8.997.2 Field Documentation

#### 8.997.2.1 uint8_t voice_diagInfo::diagInfoLen

#### 8.997.2.2 uint8_t voice_diagInfo::diagnosticInfo[255]

## 8.998 voice_DTMFInfo Struct Reference

### Data Fields

- uint8_t callID
- uint8_t DTMFEvent
- uint8_t digitCnt
- uint8_t digitBuff [255]

### 8.998.1 Detailed Description

This structure contains information about the DTMF (Dual Tone Multi-Frequency).

**Parameters**

| | |
|---|---|
| *callID* | • Call identifier for the current call. |
| *DTMFEvent* | • DTMF event<br>– 0x00 - DTMF_EVENT_REV_BURST - Sends a CDMA-burst DTMF<br>– 0x01 - DTMF_EVENT_REV_START_CONT - Starts a continuous DTMF tone<br>– 0x03 - DTMF_EVENT_REV_STOP_CONT - Stops a continuous DTMF tone<br>– 0x05 - DTMF_EVENT_FWD_BURST - Received a CDMA-burst DTMF message<br>– 0x06 - DTMF_EVENT_FWD_START_CONT - Received a start-continuous DTMF tone order<br>– 0x07 - DTMF_EVENT_FWD_STOP_CONT - Received a stop-continuous DTMF tone order |
| *digitCnt* | • Number of set of following element i.e. digitBuff. |
| *digitBuff[MAX_VOICE_DESCRIPTION_LENGTH]* | • DTMF digit buffer in ASCII string which is NULL terminated |

### 8.998.2 Field Documentation

#### 8.998.2.1 uint8_t voice_DTMFInfo::callID

#### 8.998.2.2 uint8_t voice_DTMFInfo::digitBuff[255]

#### 8.998.2.3 uint8_t voice_DTMFInfo::digitCnt

#### 8.998.2.4 uint8_t voice_DTMFInfo::DTMFEvent

## 8.999 voice_DTMFLengths Struct Reference

### Data Fields

- uint8_t DTMFPulseWidth
- uint8_t DTMFInterdigitInterval

### 8.999.1 Detailed Description

This structure contains Voice Burst DTMF pulse length information

**Parameters**

| | |
|---|---|
| *DTMFPulse-Width* | • DTMF pulse width. Values:<br>– 0x00 - DTMF_ONLENGTH_95MS - 95 ms<br>– 0x01 - DTMF_ONLENGTH_150MS - 150 ms<br>– 0x02 - DTMF_ONLENGTH_200MS - 200 ms<br>– 0x03 - DTMF_ONLENGTH_250MS - 250 ms<br>– 0x04 - DTMF_ONLENGTH_300MS - 300 ms<br>– 0x05 - DTMF_ONLENGTH_350MS - 350 ms<br>– 0x06 - DTMF_ONLENGTH_SMS SMS Tx special pulse width |
| *DTMFInterdigit-Interval* | • DTMF interdigit interval Values:<br>– 0x00 - DTMF_OFFLENGTH_60MS - 60 ms<br>– 0x01 - DTMF_OFFLENGTH_100MS - 100 ms<br>– 0x02 - DTMF_OFFLENGTH_150MS - 150 ms<br>– 0x03 - DTMF_OFFLENGTH_200MS - 200 ms |

### 8.999.2 Field Documentation

#### 8.999.2.1 uint8_t voice_DTMFLengths::DTMFInterdigitInterval

#### 8.999.2.2 uint8_t voice_DTMFLengths::DTMFPulseWidth

## 8.1000 voice_ECTNum Struct Reference

### Data Fields

- uint8_t ECTCallState
- uint8_t presentationInd
- uint8_t number [81]

### 8.1000.1 Detailed Description

Contains the parameters passed for Explicit Communication Transfer by the device.

**Parameters**

| | |
|---|---|
| *ECTCallState* | • ECT call state:<br>– 0x00 - ECT_CALL_STATE_NONE - None<br>– 0x01 - ECT_CALL_STATE_ALERTING - Alerting<br>– 0x02 - ECT_CALL_STATE_ACTIVE - Active |

| | |
|---|---|
| *presentationInd* | • Presentation indicator<br>– 0x00 - presentationAllowedAddress<br>– 0x01 - presentationRestricted<br>– 0x02 - numberNotAvailable<br>– 0x04 - presentationRestrictedAddress |
| *number* | • Number in ASCII characters terminated by NULL |

### 8.1000.2 Field Documentation

#### 8.1000.2.1 uint8_t voice_ECTNum::ECTCallState

#### 8.1000.2.2 uint8_t voice_ECTNum::number[81]

#### 8.1000.2.3 uint8_t voice_ECTNum::presentationInd

## 8.1001 voice_extDispRecInfo Struct Reference

### Data Fields

- uint8_t dispType
- uint8_t extDispInfoLen
- uint8_t extDispInfo [255]

### 8.1001.1 Detailed Description

This structure contains Line Control Information

**Parameters**

| | |
|---|---|
| *dispType* | • Values are per [S1, Table 3.7.5.16-1]. |
| *extDispInfoLen* | • Number of sets of the following elements:<br>– ext_display_info |
| *extDispInfo* | • Extended display information buffer containing the display record; refer to [S1, Section 3.7.5.16] for the format information of the buffer contents. |

### 8.1001.2 Field Documentation

#### 8.1001.2.1 uint8_t voice_extDispRecInfo::dispType

#### 8.1001.2.2 uint8_t voice_extDispRecInfo::extDispInfo[255]

#### 8.1001.2.3 uint8_t voice_extDispRecInfo::extDispInfoLen

## 8.1002 voice_getAllCallInformation Struct Reference

### Data Fields

- voice_callInfo CallInfo
- uint8_t isEmpty
- uint8_t ALS

### 8.1002.1 Detailed Description

This structure contains information related to call state change.

**Parameters**

| | |
|---|---|
| *CallInfo* | • See voice_callInfo for more information. |
| *isEmpty* | • Multiparty indicator.<br>– 0x00 - False<br>– 0x01 - True |
| *ALS* | • Alternate Line Service line indicator.<br>• Feature for supporting two different phone numbers on the same mobile device.<br>– 0x00 - ALS_LINE1 - Line 1 (default)<br>– 0x01 - ALS_LINE2 - Line 2 |

### 8.1002.2 Field Documentation

#### 8.1002.2.1 uint8_t voice_getAllCallInformation::ALS

#### 8.1002.2.2 voice_callInfo voice_getAllCallInformation::CallInfo

#### 8.1002.2.3 uint8_t voice_getAllCallInformation::isEmpty

## 8.1003 voice_getAllCallRmtPtyName Struct Reference

### Data Fields

- uint8_t callID
- voice_remotePartyName RemotePartyName

### 8.1003.1 Detailed Description

This structure contains information for All Call Remote Party Names

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *RemoteParty-Name* | • See voice_remotePartyName for more information. |

### 8.1003.2 Field Documentation

#### 8.1003.2.1 uint8_t voice_getAllCallRmtPtyName::callID

#### 8.1003.2.2 voice_remotePartyName voice_getAllCallRmtPtyName::RemotePartyName

## 8.1004 voice_getAllCallRmtPtyNum Struct Reference

### Data Fields

- uint8_t callID
- voice_remotePartyNum RemotePartyNum

### 8.1004.1 Detailed Description

This structure contains information for All Call Remote Party Numbers

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |
| *RemoteParty-Num* | • See voice_remotePartyNum for more information. |

### 8.1004.2 Field Documentation

#### 8.1004.2.1 uint8_t voice_getAllCallRmtPtyNum::callID

#### 8.1004.2.2 voice_remotePartyNum voice_getAllCallRmtPtyNum::RemotePartyNum

## 8.1005 voice_getCallFWExtInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_callFWExtInfo CallFWExtInfo [20]

### 8.1005.1 Detailed Description

This structure contains an array of Call Forwarded Extended Information.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of callFWExtInfo that follow.<br>• If zero(0) then no further information exists. |
| *CallFWExtInfo[VOICE_MAX_NO_OF_CALLS]* | • Array of CallFWExtInfo.<br>– See voice_callFWExtInfo for more information. |

### 8.1005.2 Field Documentation

#### 8.1005.2.1 voice_callFWExtInfo voice_getCallFWExtInfo::CallFWExtInfo[20]

#### 8.1005.2.2 uint8_t voice_getCallFWExtInfo::numInstances

## 8.1006 voice_getCallFWInfo Struct Reference

### Data Fields

- uint8_t numInstances
- voice_callFWInfo CallFWInfo [20]

### 8.1006.1 Detailed Description

This structure contains an array of Call Forwarded Information.

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of callFWInfo that follow.<br>• If zero(0) then no further information exists. |
| *CallFWInfo[VOICE_MAX_NO_OF_CALLS]* | • Array of callFWInfo.<br>– See voice_callFWInfo for more information. |

### 8.1006.2 Field Documentation

#### 8.1006.2.1 voice_callFWInfo voice_getCallFWInfo::CallFWInfo[20]

#### 8.1006.2.2 uint8_t voice_getCallFWInfo::numInstances

## 8.1007 voice_lineCtrlInfo Struct Reference

### Data Fields

- uint8_t polarityIncluded
- uint8_t toggleMode
- uint8_t revPolarity
- uint8_t pwrDenialTime

### 8.1007.1 Detailed Description

This structure contains Line Control Information

**Parameters**

| | |
|---|---|
| *polarityIncluded* | • Included Polarity; Boolean Value |
| *toggleMode* | • Toggle mode; Boolean Value |
| *revPolarity* | • Reverse Polarity; Boolean Value |
| *pwrDenialTime* | • Power denial time; refer to [S1, Section 3.7.5.15 Line Control] for valid values |

### 8.1007.2 Field Documentation

#### 8.1007.2.1 uint8_t voice_lineCtrlInfo::polarityIncluded

#### 8.1007.2.2 uint8_t voice_lineCtrlInfo::pwrDenialTime

#### 8.1007.2.3 uint8_t voice_lineCtrlInfo::revPolarity

#### 8.1007.2.4 uint8_t voice_lineCtrlInfo::toggleMode

## 8.1008 voice_newPwdData Struct Reference

### Data Fields

- uint8_t newPwd [4]
- uint8_t newPwdAgain [4]

### 8.1008.1 Detailed Description

This structure contains New Password Data.

**Parameters**

| | |
|---|---|
| *newPwd[BARRING_PASSWORD_LENGTH]* | • New password.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |
| *newPwdAgain[BARRING_PASSWORD_LENGTH]* | • New password again.<br>– Password consists of 4 ASCII digits.<br>– Range: 0000 to 9999. |

### 8.1008.2 Field Documentation

#### 8.1008.2.1 uint8_t voice_newPwdData::newPwd[4]

#### 8.1008.2.2 uint8_t voice_newPwdData::newPwdAgain[4]

## 8.1009 voice_NSSAudioCtrl Struct Reference

### Data Fields

- uint8_t upLink
- uint8_t downLink

### 8.1009.1 Detailed Description

This structure contains National Supplementary Services - Audio Control Information

**Parameters**

| | |
|---|---|
| *upLink* | • Values as per[ S24, 4.10 Reservation response]. |
| *downLink* | • Values as per[ S24, 4.10 Reservation response]. |

### 8.1009.2 Field Documentation

#### 8.1009.2.1 uint8_t voice_NSSAudioCtrl::downLink

#### 8.1009.2.2 uint8_t voice_NSSAudioCtrl::upLink

## 8.1010 voice_peerNumberInfo Struct Reference

### Data Fields

- uint8_t callID
- uint8_t numPI
- uint8_t numSI
- uint8_t numType
- uint8_t numPlan
- uint8_t numLen
- uint8_t number [81]

### 8.1010.1 Detailed Description

This structure contains information for Connected Peer Numbers.

**Parameters**

| | |
|---|---|
| *callID* | • Unique call identifier for the call. |

| | |
|---|---|
| *numPI* | • Number presentation indicator.<br>– 0x00 - PRESENTATION_ALLOWED - Allowed presentation<br>– 0x01 - PRESENTATION_RESTRICTED - Restricted presentation<br>– 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation<br>– 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific)<br>– 0xFF - Not Available |
| *numSI* | • Number screening indicator.<br>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened<br>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification<br>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification<br>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network |
| *numType* | • Number type.<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension |
| *numPlan* | • Number plan.<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension |
| *numLen* | • Provides the length of number which follow. |
| *number[MAX_VOICE_CALL_NO_LEN]* | • number of numLen length, NULL terminated. |

### 8.1010.2 Field Documentation

#### 8.1010.2.1 uint8_t voice_peerNumberInfo::callID

#### 8.1010.2.2 uint8_t voice_peerNumberInfo::number[81]

#### 8.1010.2.3 uint8_t voice_peerNumberInfo::numLen

#### 8.1010.2.4 uint8_t voice_peerNumberInfo::numPI

#### 8.1010.2.5 uint8_t voice_peerNumberInfo::numPlan

#### 8.1010.2.6 uint8_t voice_peerNumberInfo::numSI

#### 8.1010.2.7 uint8_t voice_peerNumberInfo::numType

## 8.1011 voice_prefVoiceSO Struct Reference

### Data Fields

- uint8_t namID
- uint8_t evrcCapability
- uint16_t homePageVoiceSO
- uint16_t homeOrigVoiceSO
- uint16_t roamOrigVoiceSO

### 8.1011.1 Detailed Description

This structure contains information about the Preferred Voice Service Options.

**Parameters**

| | |
|---|---|
| *namID* | • Index of the NAM(Number Assignment Module) to be configured.<br>• Range 0 to 3.<br>• Some modems support only 1 or 2 NAMs.<br>• 0xFF,if not available. |
| *evrcCapability* | • EVRC capability.<br>• Values:<br>– 0x00 - Disable<br>– 0x01 - Enable<br>– 0xFF - Not Available |

| | |
|---|---|
| *homePageVoiceSO* | • Home page voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.<br>• Values:<br>– 0x0000 - VOICE_SO_WILD - Any service option<br>– 0x0001 - VOICE_SO_IS_96A - IS-96A<br>– 0x0003 - VOICE_SO_EVRC - EVRC<br>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733<br>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder<br>– 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband<br>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband<br>– 0x8000 - VOICE_SO_13K - 13K<br>– 0x8001 - VOICE_SO_IS_96 - IS-96<br>– 0x8023 - VOICE_SO_WVRC - WVRC<br>– 0xFFFF - Not Available |
| *homeOrigVoiceSO* | • Home origination voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.<br>• Values:<br>– 0x0000 - VOICE_SO_WILD - Any service option<br>– 0x0001 - VOICE_SO_IS_96A - IS-96A<br>– 0x0003 - VOICE_SO_EVRC - EVRC<br>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733<br>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder<br>– 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband<br>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband<br>– 0x8000 - VOICE_SO_13K - 13K<br>– 0x8001 - VOICE_SO_IS_96 - IS-96<br>– 0x8023 - VOICE_SO_WVRC - WVRC<br>– 0xFFFF - Not Available |
| *roamOrigVoiceSO* | • Roaming origination voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.<br>• Values:<br>– 0x0000 - VOICE_SO_WILD - Any service option<br>– 0x0001 - VOICE_SO_IS_96A - IS-96A<br>– 0x0003 - VOICE_SO_EVRC - EVRC<br>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733<br>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder<br>– 0x0044 - VOICE_SO_4GV_NARR0W_BAND - 4GV narrowband<br>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband<br>– 0x8000 - VOICE_SO_13K - 13K<br>– 0x8001 - VOICE_SO_IS_96 - IS-96<br>– 0x8023 - VOICE_SO_WVRC - WVRC<br>– 0xFFFF - Not Available |

### 8.1011.2 Field Documentation

#### 8.1011.2.1 uint8_t voice_prefVoiceSO::evrcCapability

#### 8.1011.2.2 uint16_t voice_prefVoiceSO::homeOrigVoiceSO

#### 8.1011.2.3 uint16_t voice_prefVoiceSO::homePageVoiceSO

#### 8.1011.2.4 uint8_t voice_prefVoiceSO::namID

#### 8.1011.2.5 uint16_t voice_prefVoiceSO::roamOrigVoiceSO

## 8.1012 voice_redirNumInfo Struct Reference

### Data Fields

- uint8_t PI
- uint8_t SI
- uint8_t numType
- uint8_t numPlan
- uint8_t reason
- uint8_t numLen
- uint8_t number [255]

### 8.1012.1 Detailed Description

This structure contains Redirecting Number Information

**Parameters**

| | |
|---|---|
| *PI* | • Presentation indicator; refer to [S1, Table 2.7.4.4-1] for valid values. |
| *SI* | • Number of sets of following elements<br>– Caller Id |
| *SI* | • Number screening indicator.<br>• Values:<br>– 0x00 - QMI_VOICE_SI_USER_PROVIDED_NOT_SCREENED - Provided user is not screened<br>– 0x01 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_PASSED - Provided user passed verification<br>– 0x02 - QMI_VOICE_SI_USER_PROVIDED_VERIFIED_FAILED - Provided user failed verification<br>– 0x03 - QMI_VOICE_SI_NETWORK_PROVIDED - Provided network |

| | |
|---|---|
| *numType* | • Number type.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_TYPE_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_TYPE_INTERNATIONAL - International<br>– 0x02 - QMI_VOICE_NUM_TYPE_NATIONAL - National<br>– 0x03 - QMI_VOICE_NUM_TYPE_NETWORK_SPECIFIC - Network-specific<br>– 0x04 - QMI_VOICE_NUM_TYPE_SUBSCRIBER - Subscriber<br>– 0x05 - QMI_VOICE_NUM_TYPE_RESERVED - Reserved<br>– 0x06 - QMI_VOICE_NUM_TYPE_ABBREVIATED - Abbreviated<br>– 0x07 - QMI_VOICE_NUM_TYPE_RESERVED_EXTENSION - Reserved extension |
| *numPlan* | • Number plan.<br>• Values:<br>– 0x00 - QMI_VOICE_NUM_PLAN_UNKNOWN - Unknown<br>– 0x01 - QMI_VOICE_NUM_PLAN_ISDN - ISDN<br>– 0x03 - QMI_VOICE_NUM_PLAN_DATA - Data<br>– 0x04 - QMI_VOICE_NUM_PLAN_TELEX - Telex<br>– 0x08 - QMI_VOICE_NUM_PLAN_NATIONAL - National<br>– 0x09 - QMI_VOICE_NUM_PLAN_PRIVATE - Private<br>– 0x0B - QMI_VOICE_NUM_PLAN_RESERVED_CTS - Reserved cordless telephony system<br>– 0x0F - QMI_VOICE_NUM_PLAN_RESERVED_EXTENSION - Reserved extension |
| *reason* | -Redirecting reason; refer to [S1, Table 3.7.5.11-1] for valid values |
| *numLen* | • Provides the length of number which follow. |
| *number[255]* | • number of numLen length, NULL terminated. |

### 8.1012.2 Field Documentation

#### 8.1012.2.1 uint8_t voice_redirNumInfo::number[255]

#### 8.1012.2.2 uint8_t voice_redirNumInfo::numLen

#### 8.1012.2.3 uint8_t voice_redirNumInfo::numPlan

#### 8.1012.2.4 uint8_t voice_redirNumInfo::numType

#### 8.1012.2.5 uint8_t voice_redirNumInfo::PI

#### 8.1012.2.6 uint8_t voice_redirNumInfo::reason

#### 8.1012.2.7 uint8_t voice_redirNumInfo::SI

## 8.1013 voice_remotePartyName Struct Reference

### Data Fields

- uint8_t namePI
- uint8_t codingScheme
- uint8_t nameLen
- uint8_t callerName [255]

### 8.1013.1 Detailed Description

This structure contains information about the names that are dialed from the device or from which a call is received on the device.

**Parameters**

| | |
|---|---|
| *namePI* | • Name presentation indicator.<br>– 0x00 - PRESENTATION_NAME_PRESENTATION_ALLOWED - Allowed presentation<br>– 0x01 - PRESENTATION_NAME_PRESENTATION_RESTRICTED - Restricted presentation<br>– 0x02 - PRESENTATION_NAME_UNAVAILABLE - Unavailable presentation<br>– 0x03 - PRESENTATION_NAME_NAME_PRESENTATION_RESTRICTED - Restricted name presentation<br>– 0xFF - Not Available |
| *codingScheme* | • Refer to Table10 qaGobiApiTableCodingScheme.h for coding schemes<br>• 0xFF - Not Available |
| *nameLen* | • Provides the length of name which follow.<br>• If zero(0) then no further information exists. |
| *callerName[MAX_VOICE_DESCRIPTION_LENGTH]* | • Name in ASCII, NULL ending. |

### 8.1013.2 Field Documentation

#### 8.1013.2.1 uint8_t voice_remotePartyName::callerName[255]

#### 8.1013.2.2 uint8_t voice_remotePartyName::codingScheme

#### 8.1013.2.3 uint8_t voice_remotePartyName::nameLen

#### 8.1013.2.4 uint8_t voice_remotePartyName::namePI

## 8.1014 voice_remotePartyNum Struct Reference

### Data Fields

- uint8_t presentationInd
- uint8_t numLen
- uint8_t remPartyNumber [81]

### 8.1014.1 Detailed Description

This structure contains information about the numbers that are dialed from the device or from which a call is received on the device.

**Parameters**

| | |
|---|---|
| *presentationInd* | • Presentation indicator.<br>– 0x00 - PRESENTATION_ALLOWED - Allowed presentation<br>– 0x01 - PRESENTATION_RESTRICTED - Restricted presentation<br>– 0x02 - PRESENTATION_NUM_UNAVAILABLE - Unavailable presentation<br>– 0x04 - PRESENTATION_PAYPHONE - Payphone presentation (GSM/UMTS specific)<br>– 0xFF - Not Available |
| *numLen* | • Provides the length of number which follow.<br>• If zero(0) then no further information exists. |
| *remPartyNumber[MAX_CALL_NO_LEN]* | • Array of numbers in ASCII, NULL ending. |

### 8.1014.2 Field Documentation

#### 8.1014.2.1 uint8_t voice_remotePartyNum::numLen

#### 8.1014.2.2 uint8_t voice_remotePartyNum::presentationInd

#### 8.1014.2.3 uint8_t voice_remotePartyNum::remPartyNumber[81]

## 8.1015 voice_roamTimer Struct Reference

### Data Fields

- uint8_t namID
- uint32_t roamTimerValue

### 8.1015.1 Detailed Description

This structure contains information about the Roam Timer.

**Parameters**

| | |
|---|---|
| *namID* | • Index of the NAM(Number Assignment Module) to be configured.<br>• Range 0 to 3.<br>• Some modems support only 1 or 2 NAMs.<br>• 0xFF,if not available. |
| *roamTimerValue* | • Time in minutes.<br>• Cumulative air time is slammed.<br>• 0xFFFFFFFF,if not available. |

### 8.1015.2 Field Documentation

#### 8.1015.2.1 uint8_t voice_roamTimer::namID

#### 8.1015.2.2 uint32_t voice_roamTimer::roamTimerValue

## 8.1016 voice_signalInfo Struct Reference

### Data Fields

- uint8_t signalType
- uint8_t alertPitch
- uint8_t signal

### 8.1016.1 Detailed Description

This structure contains Signal Information

**Parameters**

| | |
|---|---|
| *signalType* | • Call identifier for the call. |
| *alertPitch* | • Signal Information |
| *signal* | • Caller ID Information |

### 8.1016.2 Field Documentation

#### 8.1016.2.1 uint8_t voice_signalInfo::alertPitch

#### 8.1016.2.2 uint8_t voice_signalInfo::signal

#### 8.1016.2.3 uint8_t voice_signalInfo::signalType

## 8.1017 voice_SUPSInfo Struct Reference

### Data Fields

- uint8_t svcType
- uint8_t isModByCC

### 8.1017.1 Detailed Description

This structure contains information about the Supplementary Services.

**Parameters**

| | |
|---|---|
| *svcType* | • Service type.<br>– 0x01 - SERVICE_TYPE_ACTIVATE - Activate<br>– 0x02 - SERVICE_TYPE_DEACTIVATE - Deactivate<br>– 0x03 - SERVICE_TYPE_REGISTER - Register<br>– 0x04 - SERVICE_TYPE_ERASE - Erase<br>– 0x05 - SERVICE_TYPE_INTERROGATE - Interrogate<br>– 0x06 - SERVICE_TYPE_REGISTER_PASSWORD - Register password<br>– 0x07 - SERVICE_TYPE_USSD - USSD |
| *isModByCC* | • Indicates whether the supplementary service data is modified by the card (SIM/USIM) as part of the call control:<br>– 0 - False<br>– 1 - True |

### 8.1017.2 Field Documentation

#### 8.1017.2.1 uint8_t voice_SUPSInfo::isModByCC

#### 8.1017.2.2 uint8_t voice_SUPSInfo::svcType

## 8.1018 voice_USSDNotificationNetworkInfo Struct Reference

### Data Fields

- uint8_t tlvPresent
- struct voice_USSInfo networkInfo

### 8.1018.1 Detailed Description

Structure for storing the USS info present in USSDNotification callback.

**Parameters**

| | |
|---|---|
| *tlvPresent* | Values:<br>• 0 - Not present<br>• 1 - Tlv is present |

| | |
|---|---|
| *NetworkInfo* | • USS Data from Network (See structure voice_USSInfo)<br>• NULL pointer - Invalid data. |

### 8.1018.2 Field Documentation

#### 8.1018.2.1 struct voice_USSInfo voice_USSDNotificationNetworkInfo::networkInfo

#### 8.1018.2.2 uint8_t voice_USSDNotificationNetworkInfo::tlvPresent

## 8.1019 voice_USSInfo Struct Reference

### Data Fields

- uint8_t ussDCS
- uint8_t ussLen
- uint8_t ussData [182]

### 8.1019.1 Detailed Description

This structure contains USS Information

**Parameters**

| | |
|---|---|
| *ussDCS* | • 1 - ASCII coding scheme<br>• 2 - 8-BIT coding scheme<br>• 3 - UCS2 |
| *ussLen* | • Range 1 to 182 |
| *ussData* | • Data encoded as per the DCS |

### 8.1019.2 Field Documentation

#### 8.1019.2.1 uint8_t voice_USSInfo::ussData[182]

#### 8.1019.2.2 uint8_t voice_USSInfo::ussDCS

#### 8.1019.2.3 uint8_t voice_USSInfo::ussLen

## 8.1020 voice_UUSInfo Struct Reference

### Data Fields

- uint8_t UUSType
- uint8_t UUSDcs

- uint8_t UUSDatalen
- uint8_t UUSData [255]

### 8.1020.1 Detailed Description

This structure contains User to User Signaling Service Information.

**Parameters**

| | |
|---|---|
| *UUSType* | • UUS type values are:<br>– 0x00 - UUS_DATA<br>– 0x01 - UUS_TYPE1_IMPLICIT<br>– 0x02 - UUS_TYPE1_REQUIRED<br>– 0x03 - UUS_TYPE1_NOT_REQUIRED<br>– 0x04 - UUS_TYPE2_REQUIRED<br>– 0x05 - UUS_TYPE2_NOT_REQUIRED<br>– 0x06 - UUS_TYPE3_REQUIRED<br>– 0x07 - UUS_TYPE3_NOT_REQUIRED<br>– 0xFF - Not Available |
| *UUSDcs* | • UUS data coding scheme values are:<br>– 0x01 - UUS_DCS_USP<br>– 0x02 - UUS_DCS_OHLP<br>– 0x03 - UUS_DCS_X244<br>– 0x04 - UUS_DCS_SMCF<br>– 0x05 - UUS_DCS_IA5<br>– 0x06 - UUS_DCS_RV12RD<br>– 0x07 - UUS_DCS_Q931UNCCM<br>– 0xFF - Not Available |
| *UUSDatalen* | • Number of sets of the following elements.<br>– UUSData<br>• If zero(0) then no further information exists. |
| *UUSData[MAX_VOICE_DESCRIPTION_LENGTH]* | • UUS data encoded as per coding scheme |

### 8.1020.2 Field Documentation

#### 8.1020.2.1 uint8_t voice_UUSInfo::UUSData[255]

#### 8.1020.2.2 uint8_t voice_UUSInfo::UUSDatalen

#### 8.1020.2.3 uint8_t voice_UUSInfo::UUSDcs

#### 8.1020.2.4 uint8_t voice_UUSInfo::UUSType

## 8.1021 wds_channelRate Struct Reference

### Data Fields

- uint32_t CurrChanTxRate
- uint32_t CurrChanRxRate

### 8.1021.1 Detailed Description

This structure contains Channel Rate

**Parameters**

| | |
|---|---|
| *CurrChanTxRate* | • Max channel Tx rate in bits per second |
| *CurrChanRx-Rate* | • Max channel Rx rate in bits per second |

### 8.1021.2 Field Documentation

#### 8.1021.2.1 uint32_t wds_channelRate::CurrChanRxRate

#### 8.1021.2.2 uint32_t wds_channelRate::CurrChanTxRate

## 8.1022 wds_ChannelRateTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- wds_channelRate ChannelRate

### 8.1022.1 Detailed Description

This structure stores information about channel rate TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *ChannelRate* | • see wds_channelRate for more info |

### 8.1022.2 Field Documentation

#### 8.1022.2.1 wds_channelRate wds_ChannelRateTlv::ChannelRate

#### 8.1022.2.2 uint8_t wds_ChannelRateTlv::TlvPresent

## 8.1023 wds_ConnStatusTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t MDMConnStatus

### 8.1023.1 Detailed Description

This structure stores information about modem connection status TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *MDMConn-Status* | -Modem connecton status<br>• 0x01 - disconnected<br>• 0x02 - connected |

### 8.1023.2 Field Documentation

#### 8.1023.2.1 uint8_t wds_ConnStatusTlv::MDMConnStatus

#### 8.1023.2.2 uint8_t wds_ConnStatusTlv::TlvPresent

## 8.1024 wds_currNetworkInfo Struct Reference

### Data Fields

- uint8_t NetworkType
- uint32_t RATMask
- uint32_t SOMask

### 8.1024.1 Detailed Description

Network information structure

**Parameters**

| | |
|---|---|
| *NetworkType* | • Values:<br>– 0 - 3GPP<br>– 1 - 3GPP2 |
| *RATMask* | • RAT Mask |
| *SOMask* | • SO Mask |

### 8.1024.2 Field Documentation

#### 8.1024.2.1 uint8_t wds_currNetworkInfo::NetworkType

#### 8.1024.2.2 uint32_t wds_currNetworkInfo::RATMask

#### 8.1024.2.3 uint32_t wds_currNetworkInfo::SOMask

## 8.1025 wds_DataBearTechTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t DataBearerTech

### 8.1025.1 Detailed Description

This structure hold parameters about data bearer technology TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *DataBearerTech* | • Data bear technology<br>– 0x01 - cdma2000 1X<br>– 0x02 - cdma2000 HRPD (1xEV-DO)<br>– 0x03 - GSM<br>– 0x04 - UMTS<br>– 0x05 - cdma200 HRPD ( 1xEV-DO RevA)<br>– 0x06 - EDGE<br>– 0x07 - HSDPA and WCDMA<br>– 0x08 - WCDMA and HSUPA<br>– 0x09 - HSDPA and HSUPA<br>– 0x0A - LTE<br>– 0x0B - cdma2000 EHRPD<br>– 0x0C - HSDPA+ and WCDMA<br>– 0x0D - HSDPA+ and HSUPA<br>– 0x0E - DC_HSDPA+ and WCDMA<br>– 0x0F - DC_HSDPA+ and HSUPA<br>– 0x10 - HSDPA+ and 64QAM<br>– 0x11 - HSDPA+, 64QAM and HSUPA<br>– 0x12 - TDSCDMA<br>– 0x13 - TDSCDMA and HSDPA<br>– 0xFF - Unknown |

### 8.1025.2 Field Documentation

#### 8.1025.2.1 uint8_t wds_DataBearTechTlv::DataBearerTech

#### 8.1025.2.2 uint8_t wds_DataBearTechTlv::TlvPresent

## 8.1026 wds_DataULongLongTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t ullData

### 8.1026.1 Detailed Description

This structure contains Data unsigned long long TLV data.

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *ullData* | unsigend long long data in 64bit |

### 8.1026.2 Field Documentation

#### 8.1026.2.1 uint8_t wds_DataULongLongTlv::TlvPresent

#### 8.1026.2.2 uint64_t wds_DataULongLongTlv::ullData

## 8.1027 wds_DataULongTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t ulData

### 8.1027.1 Detailed Description

This structure contains Data unsigned long TLV data.

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *ulData* | unsigend long data in 32bit |

### 8.1027.2 Field Documentation

#### 8.1027.2.1 uint8_t wds_DataULongTlv::TlvPresent

#### 8.1027.2.2 uint32_t wds_DataULongTlv::ulData

## 8.1028 wds_DHCPLeaseOptTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t numOpt

- wds_DHCPOpt optList [30]
- uint8_t optListData [2048]

### 8.1028.1 Detailed Description

DHCP lease option information

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *numOpt* | number of sets of wds_DHCPOpt |
| *optList* | option list |
| *optListData* | option list data |

### 8.1028.2 Field Documentation

#### 8.1028.2.1 uint8_t wds_DHCPLeaseOptTlv::numOpt

#### 8.1028.2.2 wds_DHCPOpt wds_DHCPLeaseOptTlv::optList[30]

#### 8.1028.2.3 uint8_t wds_DHCPLeaseOptTlv::optListData[2048]

#### 8.1028.2.4 uint8_t wds_DHCPLeaseOptTlv::TlvPresent

## 8.1029 wds_DHCPLeaseStateTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t leaseState

### 8.1029.1 Detailed Description

DHCP lease state information

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *leaseState* | lease state<br>• 0 - active, newly acquired<br>• 1 - active, renewed<br>• 2 - active, renewing<br>• 3 - active, rebinding<br>• 4 - inactive, expired<br>• 5 - inactive, renew refused<br>• 6 - inactive, rebind refused<br>• 7 - inactive, other |

### 8.1029.2 Field Documentation

#### 8.1029.2.1 uint8_t wds_DHCPLeaseStateTlv::leaseState

#### 8.1029.2.2 uint8_t wds_DHCPLeaseStateTlv::TlvPresent

## 8.1030 wds_DHCPOpt Struct Reference

### Data Fields

- uint8_t optCode
- uint8_t optValLen
- uint8_t * pOptVal

### 8.1030.1 Detailed Description

DHCP option code information

**Parameters**

| | |
|---|---|
| *optCode* | option code<br>• values:<br>– 0 - 255 |
| *optValLen* | length of option code<br>• values:<br>– 0 - 255 |
| *pOptVal* | option value |

### 8.1030.2 Field Documentation

#### 8.1030.2.1 uint8_t wds_DHCPOpt::optCode

#### 8.1030.2.2 uint8_t wds_DHCPOpt::optValLen

#### 8.1030.2.3 uint8_t* wds_DHCPOpt::pOptVal

## 8.1031 wds_DHCPProfileIdTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t profileType
- uint8_t profileId

### 8.1031.1 Detailed Description

DHCP profile id TLV information

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *profileType* | identifying the type of the profile<br>• 0 - 3GPP |

| | |
|---|---|
| *profileId* | index identifying the profile<br>• 1-24 valid for 3GPP profile type (9x30 and onwards) |

### 8.1031.2 Field Documentation

#### 8.1031.2.1 uint8_t wds_DHCPProfileIdTlv::profileId

#### 8.1031.2.2 uint8_t wds_DHCPProfileIdTlv::profileType

#### 8.1031.2.3 uint8_t wds_DHCPProfileIdTlv::TlvPresent

## 8.1032 wds_DHCPv4HWConfig Struct Reference

### Data Fields

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

### 8.1032.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

**Parameters**

| | |
|---|---|
| *hwType* | • HW Type 1 - Ethernet 20 - Serial |
| *chaddrlen* | • chaddrlen |
| *chaddr* | • chaddr. Max size 16 bytes |

### 8.1032.2 Field Documentation

#### 8.1032.2.1 uint8_t wds_DHCPv4HWConfig::chaddr[16]

#### 8.1032.2.2 uint8_t wds_DHCPv4HWConfig::chaddrLen

#### 8.1032.2.3 uint8_t wds_DHCPv4HWConfig::hwType

## 8.1033 wds_DHCPv4Option Struct Reference

### Data Fields

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

### 8.1033.1 Detailed Description

WDS SWI DHCPv4 Option Structure

**Parameters**

| | |
|---|---|
| *optCode* | • Option code<br>– 0 - 255 |
| *optValLen* | • Option value length<br>– 0 - 255 |
| *optVal* | • Option value |

### 8.1033.2 Field Documentation

#### 8.1033.2.1 uint8_t wds_DHCPv4Option::optCode

#### 8.1033.2.2 uint8_t wds_DHCPv4Option::optVal[255]

#### 8.1033.2.3 uint8_t wds_DHCPv4Option::optValLen

## 8.1034 wds_DHCPv4OptionList Struct Reference

### Data Fields

- uint8_t numOpt
- wds_DHCPv4Option * pOptList

### 8.1034.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

**Parameters**

| | |
|---|---|
| *numOpt* | • number of options<br>– 0 - 255 |
| *pOptList* | • pointer to list of DHCP Options |

### 8.1034.2 Field Documentation

#### 8.1034.2.1 uint8_t wds_DHCPv4OptionList::numOpt

#### 8.1034.2.2 wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList

## 8.1035 wds_DHCPv4ProfileId Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileId

### 8.1035.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

**Parameters**

| | |
|---|---|
| *profileType* | • 0 for 3GPP |
| *profileId* | • 1 to 24 for 3GPP profile |

### 8.1035.2 Field Documentation

#### 8.1035.2.1 uint8_t wds_DHCPv4ProfileId::profileId

#### 8.1035.2.2 uint8_t wds_DHCPv4ProfileId::profileType

## 8.1036 wds_Domain Struct Reference

### Data Fields

- uint16_t domainLen
- uint8_t domainName [256]

### 8.1036.1 Detailed Description

This structure contains the DomainName Information

**Parameters**

| | |
|---|---|
| *domainLen* | • length of the recieved Domain name |
| *domainName* | • Domain name(Max 256 characters) |

### 8.1036.2 Field Documentation

#### 8.1036.2.1 uint16_t wds_Domain::domainLen

#### 8.1036.2.2 uint8_t wds_Domain::domainName[256]

## 8.1037 wds_DomainNameList Struct Reference

### Data Fields

- uint8_t numInstances
- struct wds_Domain domain [10]

### 8.1037.1 Detailed Description

This structure contains the DomainNameList Information

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of Domain name receieved |
| *domain* | • Domain name information(Max 10 Domain names) |

**Note**

Data invalid when numInstances equal 0xff.

### 8.1037.2 Field Documentation

#### 8.1037.2.1 struct wds_Domain wds_DomainNameList::domain[10]

#### 8.1037.2.2 uint8_t wds_DomainNameList::numInstances

## 8.1038 wds_DormStatTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint8_t DormancyStat

### 8.1038.1 Detailed Description

This structure stores information about dormancy status TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *DormancyStat* | -Dormancy status<br>• 0x01 - Traffic channel dormant<br>• 0x02 - Traffic channel active |

### 8.1038.2 Field Documentation

#### 8.1038.2.1 uint8_t wds_DormStatTlv::DormancyStat

#### 8.1038.2.2 uint8_t wds_DormStatTlv::TlvPresent

## 8.1039 wds_GPRSQoS Struct Reference

### Data Fields

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

### 8.1039.1 Detailed Description

This structure contains the GPRS Quality Of Service Information

**Parameters**

| | |
|---|---|
| *precedence-Class* | • Precedence class |
| *delayClass* | • Delay class |
| *reliabilityClass* | • Reliability class |
| *peak-Throughput-Class* | • Peak throughput class |
| *mean-Throughput-Class* | • Mean throughput class |

**Note**

Data invalid when all parameters are equal to 0xffffffff.

### 8.1039.2 Field Documentation

#### 8.1039.2.1 uint32_t wds_GPRSQoS::delayClass

#### 8.1039.2.2 uint32_t wds_GPRSQoS::meanThroughputClass

#### 8.1039.2.3 uint32_t wds_GPRSQoS::peakThroughputClass

#### 8.1039.2.4 uint32_t wds_GPRSQoS::precedenceClass

#### 8.1039.2.5 uint32_t wds_GPRSQoS::reliabilityClass

## 8.1040 wds_IPv4AdTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint32_t IPv4Addr

### 8.1040.1 Detailed Description

IPv4 address TLV information

**Parameters**

| | |
|---|---|
| *TlvPresent* | a flag indicates whether TLV presented or not |
| *IPv4Addr* | IPv4 address |

### 8.1040.2 Field Documentation

#### 8.1040.2.1 uint32_t wds_IPv4AdTlv::IPv4Addr

#### 8.1040.2.2 uint8_t wds_IPv4AdTlv::TlvPresent

## 8.1041 wds_IPV6AddressInfo Struct Reference

### Data Fields

- uint8_t IPV6PrefixLen
- uint16_t IPAddressV6 [8]

### 8.1041.1 Detailed Description

This structure contains the IPV6 Address Information

**Parameters**

| | |
|---|---|
| *IPV6PrefixLen* | • Length of the received IPv6 address in no. of bits; can take value between 0 and 128<br>– 0xFF - Not Available |
| *IPAddressV6* | • IPv6 address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format. |

### 8.1041.2 Field Documentation

#### 8.1041.2.1 uint16_t wds_IPV6AddressInfo::IPAddressV6[8]

#### 8.1041.2.2 uint8_t wds_IPV6AddressInfo::IPV6PrefixLen

## 8.1042 wds_IPV6GWAddressInfo Struct Reference

### Data Fields

- uint8_t gwV6PrefixLen

- uint16_t gwAddressV6 [8]

### 8.1042.1 Detailed Description

This structure contains the IPV6 Gateway Address Information

**Parameters**

| | |
|---|---|
| *gwV6PrefixLen* | • Length of the received IPv6 Gateway address in no. of bits; can take value between 0 and 128 |
| *IPAddressV6* | • IPv6 Gateway address(in network byte order); This is an 8-element array of 16 bit numbers, each of which is in big endian format. |

**Note**

Data invalid when IPV6PrefixLen equal 0xff.

### 8.1042.2 Field Documentation

#### 8.1042.2.1 uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]

#### 8.1042.2.2 uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen

## 8.1043 wds_LastMdmCallEndRsnTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint16_t CallEndReason

### 8.1043.1 Detailed Description

This structure stores information about modem last call end reason TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *CallEndReason* | -Modem call end reason<br>• See qaGobiApiTableCallEndReasons.h for Call End Reason |

### 8.1043.2 Field Documentation

#### 8.1043.2.1 uint16_t wds_LastMdmCallEndRsnTlv::CallEndReason

#### 8.1043.2.2 uint8_t wds_LastMdmCallEndRsnTlv::TlvPresent

## 8.1044 wds_PCSCFFQDNAddress Struct Reference

### Data Fields

- uint16_t fqdnLen
- uint8_t fqdnAddr [256]

### 8.1044.1 Detailed Description

This structure contains the PCSCFFQDNAddress Information

**Parameters**

| | |
|---|---|
| *fqdnLen* | • length of the received FQDN address |
| *fqdnAddr* | • FQDN address(Max 256 characters) |

### 8.1044.2 Field Documentation

#### 8.1044.2.1 uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]

#### 8.1044.2.2 uint16_t wds_PCSCFFQDNAddress::fqdnLen

## 8.1045 wds_PCSCFFQDNAddressList Struct Reference

### Data Fields

- uint8_t numInstances
- struct wds_PCSCFFQDNAddress pcsfFQDNAddress [10]

### 8.1045.1 Detailed Description

This structure contains the PCSCFFQDNAddressList Information

**Parameters**

| | |
|---|---|
| *numInstances* | • Number of FQDN addresses received |
| *pcsfFQDN-Address* | • FQDN address information(Max 10 addresses) |

**Note**

Data invalid when numInstances equal 0xff.

### 8.1045.2 Field Documentation

#### 8.1045.2.1 uint8_t wds_PCSCFFQDNAddressList::numInstances

#### 8.1045.2.2 struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]

## 8.1046 wds_PCSCFIPv4ServerAddressList Struct Reference

### Data Fields

- uint8_t numInstances
- uint32_t pscsfIPv4Addr [64]

### 8.1046.1 Detailed Description

This structure contains the PCSCFIPv4ServerAddressList Information

**Parameters**

| | |
|---|---|
| *numInstances* | • number of address following |
| *pscsfIPv4Addr* | • P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each) |

**Note**

Data invalid when numInstances equal 0xff and pscsfIPv4Addr equal to 0xffffffff.

### 8.1046.2 Field Documentation

#### 8.1046.2.1 uint8_t wds_PCSCFIPv4ServerAddressList::numInstances

#### 8.1046.2.2 uint32_t wds_PCSCFIPv4ServerAddressList::pscsfIPv4Addr[64]

## 8.1047 wds_ProfileIdentifier Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileIndex

### 8.1047.1 Detailed Description

This structure contains the Profile Identifier Information

**Parameters**

| | |
|---|---|
| *profileType* | • Identifies the type of profile 0x00 = 3GPP |
| *profileIndex* | • Index of profile whose settings were loaded prior to session parameter negotiation for the current call. If this TLV is not present, data call parameters are based on device default settings for each parameter |

**Note**

Data invalid when all parameters are equal to 0xff.

### 8.1047.2 Field Documentation

#### 8.1047.2.1 uint8_t wds_ProfileIdentifier::profileIndex

#### 8.1047.2.2 uint8_t wds_ProfileIdentifier::profileType

## 8.1048 wds_profileInfo Union Reference

### Data Fields

- LibPackprofile_3GPP SlqsProfile3GPP
- LibPackprofile_3GPP2 SlqsProfile3GPP2

### 8.1048.1 Detailed Description

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

**Parameters**

| | |
|---|---|
| *SlqsProfile3GPP* | 3GPP profile See LibPackprofile_3GPP |
| *SlqsProfile3GP-P2* | 3GPP2 profile See LibPackprofile_3GPP2 |

### 8.1048.2 Field Documentation

#### 8.1048.2.1 LibPackprofile_3GPP wds_profileInfo::SlqsProfile3GPP

#### 8.1048.2.2 LibPackprofile_3GPP2 wds_profileInfo::SlqsProfile3GPP2

## 8.1049 wds_RXBytesOKTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t RxByteOKCnt

### 8.1049.1 Detailed Description

This structure stores information about Rx Bytes OK TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *RxByteOKCnt* | -Number of bytes received without error |

### 8.1049.2 Field Documentation

#### 8.1049.2.1 uint64_t wds_RXBytesOKTlv::RxByteOKCnt

#### 8.1049.2.2 uint8_t wds_RXBytesOKTlv::TlvPresent

## 8.1050 wds_transferStatInd Struct Reference

### Data Fields

- uint8_t StatsPeriod
- uint32_t StatsMask

### 8.1050.1 Detailed Description

This structure contains the transfer statistic indication information

**Parameters**

| | |
|---|---|
| *StatsPeriod* | Field Period between transfer statistic reports.<br>• 0 - Do not report.<br>• 1 - Other - Period between reports (in seconds). |
| *StatsMask* | requested statistic bit mask.<br>• 0x00000040 - Tx bytes OK<br>• 0x00000080 - Rx bytes OK |

### 8.1050.2 Field Documentation

#### 8.1050.2.1 uint32_t wds_transferStatInd::StatsMask

#### 8.1050.2.2 uint8_t wds_transferStatInd::StatsPeriod

## 8.1051 wds_TrStatInd Struct Reference

### Data Fields

- uint8_t statsPeriod
- uint32_t statsMask

### 8.1051.1 Detailed Description

This structure contains the information about the Transfer Statistics Indicator parameters.

**Parameters**

| | |
|---|---|
| *statsPeriod* | • Period between transfer statistics reports.<br>– 0 - Do not report<br>– Other - Period between reports (seconds) |

| | |
|---|---|
| *statsMask* | • Requested statistic bit mask.<br>– 0x00000001 - Tx packets OK<br>– 0x00000002 - Rx packets OK<br>– 0x00000004 - Tx packet errors<br>– 0x00000008 - Rx packet errors<br>– 0x00000010 - Tx overflows<br>– 0x00000020 - Rx overflows<br>– 0x00000040 - Tx bytes OK<br>– 0x00000080 - Rx bytes OK<br>• Each bit set causes the corresponding optional information to be sent in SLQSWdsEventReport-CallBack.<br>• All unlisted bits are reserved for future use and must be set to zero. |

### 8.1051.2 Field Documentation

#### 8.1051.2.1 uint32_t wds_TrStatInd::statsMask

#### 8.1051.2.2 uint8_t wds_TrStatInd::statsPeriod

## 8.1052 wds_TXBytesOKTlv Struct Reference

### Data Fields

- uint8_t TlvPresent
- uint64_t TxByteOKCnt

### 8.1052.1 Detailed Description

This structure stores information about Tx Bytes OK TLV.

**Parameters**

| | |
|---|---|
| *TlvPresent* | • value is 1 if this TLV is present in indication, otherwise 0 |
| *TxByteOKCnt* | • Number of bytes transmitted without error |

### 8.1052.2 Field Documentation

#### 8.1052.2.1 uint8_t wds_TXBytesOKTlv::TlvPresent

#### 8.1052.2.2 uint64_t wds_TXBytesOKTlv::TxByteOKCnt

## 8.1053 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

### 8.1053.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

**Parameters**

| | |
|---|---|
| *trafficClass* | • 0x00 - Subscribed<br>• 0x01 - Conversational<br>• 0x02 - Streaming<br>• 0x03 - Interactive<br>• 0x04 - Background |
| *maxUplinkBitrate* | • Maximum uplink bit rate in bits/sec |
| *maxDownlink-Bitrate* | • Maximum downlink bit rate in bits/sec |
| *grntUplinkBitrate* | • Guaranteed uplink bit rate in bits/sec |
| *grntDownlink-Bitrate* | • Guaranteed downlink bit rate in bits/sec |
| *qosDelivery-Order* | - Qos delivery order<br>• 0x00 - Subscribe<br>• 0x01 - Delivery order on<br>• 0x02 - Delivery order off |
| *maxSDUSize* | • Maximum SDU size |

| | |
|---|---|
| *sduErrorRatio* | - SDU error ratio<br>• Target value for fraction of SDUs lost or detected as erroneous.<br>• 0x00 - Subscribe<br>• 0x01 - 1∗10^(-2)<br>• 0x02 - 7∗10^(-3)<br>• 0x03 - 1∗10^(-3)<br>• 0x04 - 1∗10^(-4)<br>• 0x05 - 1∗10^(-5)<br>• 0x06 - 1∗10^(-6)<br>• 0x07 - 1∗10^(-1) |
| *resBerRatio* | - Residual bit error ratio<br>• Target value for undetected bit error ratio in in the delivered SDUs.<br>• 0x00 - Subscribe<br>• 0x01 - 5∗10^(-2)<br>• 0x02 - 1∗10^(-2)<br>• 0x03 - 5∗10^(-3)<br>• 0x04 - 4∗10^(-3)<br>• 0x05 - 1∗10^(-3)<br>• 0x06 - 1∗10^(-4)<br>• 0x07 - 1∗10^(-5)<br>• 0x08 - 1∗10^(-6)<br>• 0x09 - 1∗10^(-8) |
| *deliveryErrSDU* | - delivery of erroneous SDUs<br>• Indicates whether SDUs detected as erroneous shall be delivered or not.<br>• 0x00 - Subscribe<br>• 0x01 - 5∗10^(-2)<br>• 0x02 - 1∗10^(-2)<br>• 0x03 - 5∗10^(-3)<br>• 0x04 - 4∗10^(-3)<br>• 0x05 - 1∗10^(-3)<br>• 0x06 - 1∗10^(-4)<br>• 0x07 - 1∗10^(-5)<br>• 0x08 - 1∗10^(-6)<br>• 0x09 - 1∗10^(-8) |
| *transferDelay* | - Transfer delay (ms)<br>• Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds. |
| *trafficPriority* | - Transfer handling priority<br>• Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers. |

### 8.1053.2 Field Documentation

#### 8.1053.2.1 uint8_t wds_UMTSMinQoS::deliveryErrSDU

#### 8.1053.2.2 uint32_t wds_UMTSMinQoS::grntDownlinkBitrate

#### 8.1053.2.3 uint32_t wds_UMTSMinQoS::grntUplinkBitrate

#### 8.1053.2.4 uint32_t wds_UMTSMinQoS::maxDownlinkBitrate

#### 8.1053.2.5 uint32_t wds_UMTSMinQoS::maxSDUSize

#### 8.1053.2.6 uint32_t wds_UMTSMinQoS::maxUplinkBitrate

#### 8.1053.2.7 uint8_t wds_UMTSMinQoS::qosDeliveryOrder

#### 8.1053.2.8 uint8_t wds_UMTSMinQoS::resBerRatio

#### 8.1053.2.9 uint8_t wds_UMTSMinQoS::sduErrorRatio

#### 8.1053.2.10 uint8_t wds_UMTSMinQoS::trafficClass

#### 8.1053.2.11 uint32_t wds_UMTSMinQoS::trafficPriority

#### 8.1053.2.12 uint32_t wds_UMTSMinQoS::transferDelay

## 8.1054 wdsDhcpv4HwConfig Struct Reference

### Data Fields

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

### 8.1054.1 Detailed Description

Structure contain DHCP V4 Hardware Configuration.

**Parameters**

| | |
|---|---|
| *hwType* | DHCP HW Type, examples:<br>• 0 - Ethernet<br>• 20 - Serial |
| *chaddrLen* | Length of chaddr field, examples:<br>• 6 for Ethernet MAC address |
| *chaddr* | Client hardware address |

### 8.1054.2 Field Documentation

#### 8.1054.2.1 uint8_t wdsDhcpv4HwConfig::chaddr[16]

#### 8.1054.2.2 uint8_t wdsDhcpv4HwConfig::chaddrLen

#### 8.1054.2.3 uint8_t wdsDhcpv4HwConfig::hwType

## 8.1055 wdsDhcpv4Option Struct Reference

### Data Fields

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

### 8.1055.1 Detailed Description

This structure contains DHCP V4 Option information.

**Parameters**

| | |
|---|---|
| *optCode* | Option code<br>• 0 - 255 |
| *optValLen* | Option value length<br>• 0 - 255 |
| *optVal* | Option Value |

### 8.1055.2 Field Documentation

#### 8.1055.2.1 uint8_t wdsDhcpv4Option::optCode

#### 8.1055.2.2 uint8_t wdsDhcpv4Option::optVal[255]

#### 8.1055.2.3 uint8_t wdsDhcpv4Option::optValLen

## 8.1056 wdsDhcpv4OptionList Struct Reference

### Data Fields

- uint8_t numOpt
- wdsDhcpv4Option * pOptList

### 8.1056.1 Detailed Description

Structure contain DHCP V4 Option List

**Parameters**

| | |
|---|---|
| *numOpt* | number of options<br>• 0 - 255 |
| *pOptList* | pointer to list of DHCP Options<br>• See wdsDhcpv4Option |

### 8.1056.2 Field Documentation

#### 8.1056.2.1 uint8_t wdsDhcpv4OptionList::numOpt

#### 8.1056.2.2 wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList

## 8.1057 wdsDhcpv4ProfileId Struct Reference

### Data Fields

- uint8_t profileType
- uint8_t profileId

### 8.1057.1 Detailed Description

This structure contains DHCP v4 ProfielID Information.

**Parameters**

| | |
|---|---|
| *profileType* | profile type<br>• 0 - 3GPP |
| *profileId* | profile index<br>• index identifiying the profile 1-24 valid for 3GPP profile type (EM74xx and onwards) |

### 8.1057.2 Field Documentation

#### 8.1057.2.1 uint8_t wdsDhcpv4ProfileId::profileId

#### 8.1057.2.2 uint8_t wdsDhcpv4ProfileId::profileType

# Chapter 9

# File Documentation

## 9.1 apdoxypages.c File Reference

Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages.

### Namespaces

- Tables

### 9.1.1 Detailed Description

Contains the module declaration for the Doxygen output. Also contains the content of the main page and related pages.

## 9.2 audio.h File Reference

### Data Structures

- struct pack_audio_SLQSGetAudioProfile_t
- struct unpack_audio_SLQSGetAudioProfile_t
- struct pack_audio_SLQSSetAudioProfile_t
- struct pack_audio_SLQSGetAudioPathConfig_t
- struct audio_TXPCMIIRFltr
- struct audio_RXPCMIIRFltr
- struct audio_RXAGCList
- struct audio_RXAVCList
- struct audio_TXAGCList
- struct unpack_audio_SLQSGetAudioPathConfig_t
- struct pack_audio_SLQSSetAudioPathConfig_t
- struct pack_audio_SLQSGetAudioVolTLBConfig_t
- struct unpack_audio_SLQSGetAudioVolTLBConfig_t
- struct pack_audio_SLQSSetAudioVolTLBConfig_t
- struct unpack_audio_SLQSSetAudioVolTLBConfig_t

## Functions

- int pack_audio_SLQSGetAudioProfile (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_audio_SLQSGetAudioProfile_t ∗pReqParam)
- int unpack_audio_SLQSGetAudioProfile (uint8_t ∗pResp, uint16_t respLen, unpack_audio_SLQSGetAudioProfile_t ∗pOutput)
- int pack_audio_SLQSSetAudioProfile (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_audio_SLQSSetAudioProfile_t ∗pReqParam)
- int unpack_audio_SLQSSetAudioProfile (uint8_t ∗pResp, uint16_t respLen)
- int pack_audio_SLQSGetAudioPathConfig (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_audio_SLQSGetAudioPathConfig_t ∗pReqParam)
- int unpack_audio_SLQSGetAudioPathConfig (uint8_t ∗pResp, uint16_t respLen, unpack_audio_SLQSGetAudioPathConfig_t ∗pOutput)
- int pack_audio_SLQSSetAudioPathConfig (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_audio_SLQSSetAudioPathConfig_t ∗pReqParam)
- int unpack_audio_SLQSSetAudioPathConfig (uint8_t ∗pResp, uint16_t respLen)
- int pack_audio_SLQSGetAudioVolTLBConfig (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_audio_SLQSGetAudioVolTLBConfig_t ∗pReqParam)
- int unpack_audio_SLQSGetAudioVolTLBConfig (uint8_t ∗pResp, uint16_t respLen, unpack_audio_SLQSGetAudioVolTLBConfig_t ∗pOutput)
- int pack_audio_SLQSSetAudioVolTLBConfig (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_audio_SLQSSetAudioVolTLBConfig_t ∗pReqParam)
- int unpack_audio_SLQSSetAudioVolTLBConfig (uint8_t ∗pResp, uint16_t respLen, unpack_audio_SLQSSetAudioVolTLBConfig_t ∗pOutput)

### 9.2.1 Function Documentation

#### 9.2.1.1 int pack_audio_SLQSGetAudioPathConfig ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen*, pack_audio_SLQSGetAudioPathConfig_t ∗ *pReqParam* )

Gets the audio path configuration parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: SL9090

#### 9.2.1.2 int pack_audio_SLQSGetAudioProfile ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen*, pack_audio_SLQSGetAudioProfile_t ∗ *pReqParam* )

Gets get the profile content of the requested audio generator. pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: SL9090

#### 9.2.1.3 int pack_audio_SLQSGetAudioVolTLBConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_audio_SLQSGetAudioVolTLBConfig_t * *pReqParam* )

Gets the audio VolTLB configuration parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: SL9090

#### 9.2.1.4 int pack_audio_SLQSSetAudioPathConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_audio_SLQSSetAudioPathConfig_t * *pReqParam* )

sets the audio path configuration parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: SL9090

#### 9.2.1.5 int pack_audio_SLQSSetAudioProfile ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_audio_SLQSSetAudioProfile_t * *pReqParam* )

sets an audio profile pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: SL9090

#### 9.2.1.6 int pack_audio_SLQSSetAudioVoITLBConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_audio_SLQSSetAudioVoITLBConfig_t * *pReqParam* )

Sets the audio VoITLB configuration parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: SL9090

#### 9.2.1.7 int unpack_audio_SLQSGetAudioPathConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_audio_SLQSGetAudioPathConfig_t * *pOutput* )

Gets the audio path configuration parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.2.1.8 int unpack_audio_SLQSGetAudioProfile ( uint8_t * *pResp*, uint16_t *respLen*, unpack_audio_SLQSGetAudioProfile_t * *pOutput* )

Gets get the profile content of the requested audio generator unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.2.1.9 int unpack_audio_SLQSGetAudioVolTLBConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_audio_SLQSGetAudioVolTLBConfig_t * *pOutput* )

Gets the audio VolTLB configuration parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.2.1.10 int unpack_audio_SLQSSetAudioPathConfig ( uint8_t * *pResp*, uint16_t *respLen* )

sets the audio path configuration parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.2.1.11 int unpack_audio_SLQSSetAudioProfile ( uint8_t * *pResp*, uint16_t *respLen* )

sets an audio profile unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.2.1.12 int unpack_audio_SLQSSetAudioVoITLBConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_audio_SLQSSetAudioVoITLBConfig_t * *pOutput* )

Sets the audio VoITLB configuration parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

#### Returns

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

#### See Also

See qmerrno.h for eQCWWAN_xxx error values

## 9.3 cat.h File Reference

### Data Structures

- struct pack_cat_SetCATEventCallback_t
- struct unpack_cat_SetCATEventCallback_t
- struct cat_EventIDDataTlv
- struct cat_AlPhaIdentifierTlv
- struct cat_EventListTlv
- struct cat_RefreshTlv
- struct cat_EndProactiveSessionTlv
- union cat_currentCatEvent
- struct cat_commonEventTlv
- struct unpack_cat_SetCatEventCallback_ind_t
- struct pack_cat_CATSendEnvelopeCommand_t
- struct pack_cat_CATSendTerminalResponse_t

### Macros

- #define CAT_EVENT_DATA_MAX_LENGTH 255
- #define CAN_COMMON_EVENT_TLV_NUMBER 11

### Typedefs

- typedef unpack_result_t unpack_cat_CATSendEnvelopeCommand_t
- typedef unpack_result_t unpack_cat_CATSendTerminalResponse_t

### Functions

- int pack_cat_SetCATEventCallback (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_cat_SetCATEventCallback_t ∗reqArg)
- int unpack_cat_SetCATEventCallback (uint8_t ∗pResp, uint16_t respLen, unpack_cat_SetCATEventCallback_t ∗pOutput)
- int unpack_cat_SetCatEventCallback_ind (uint8_t ∗pResp, uint16_t respLen, unpack_cat_SetCatEventCallback_ind_t ∗pOutput)
- int pack_cat_CATSendEnvelopeCommand (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_cat_CATSendEnvelopeCommand_t ∗reqArg)
- int unpack_cat_CATSendEnvelopeCommand (uint8_t ∗pResp, uint16_t respLen, unpack_cat_CATSendEnvelopeCommand_t ∗pOutput)
- int pack_cat_CATSendTerminalResponse (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_cat_CATSendTerminalResponse_t ∗reqArg)
- int unpack_cat_CATSendTerminalResponse (uint8_t ∗pResp, uint16_t respLen, unpack_cat_CATSendTerminalResponse_t ∗pOutput)

### 9.3.1 Macro Definition Documentation

#### 9.3.1.1 #define CAN_COMMON_EVENT_TLV_NUMBER 11

#### 9.3.1.2 #define CAT_EVENT_DATA_MAX_LENGTH 255

### 9.3.2 Typedef Documentation

#### 9.3.2.1 typedef unpack_result_t unpack_cat_CATSendEnvelopeCommand_t

#### 9.3.2.2 typedef unpack_result_t unpack_cat_CATSendTerminalResponse_t

### 9.3.3 Function Documentation

#### 9.3.3.1 int pack_cat_CATSendEnvelopeCommand ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_cat_CATSendEnvelopeCommand_t * *reqArg* )

Sends the envelope command to the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.3.3.2 int pack_cat_CATSendTerminalResponse ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_cat_CATSendTerminalResponse_t * *reqArg* )

Sends the terminal response to the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter<br>• |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.3.3.3 int pack_cat_SetCATEventCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_cat_SetCATEventCallback_t * *reqArg* )

Enables/disables the CAT event callback pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.3.3.4 int unpack_cat_CATSendEnvelopeCommand ( uint8_t * *pResp*, uint16_t *respLen*, unpack_cat_CATSendEnvelopeCommand_t * *pOutput* )

Sends the envelope command to the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.3.3.5 int unpack_cat_CATSendTerminalResponse ( uint8_t * *pResp*, uint16_t *respLen*, unpack_cat_CATSendTerminalResponse_t * *pOutput* )

Sends the terminal response to the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.3.3.6 int unpack_cat_SetCATEventCallback ( uint8_t * *pResp*, uint16_t *respLen*, unpack_cat_SetCATEventCallback_t * *pOutput* )

Enables/disables the CAT event callback unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.3.3.7 int unpack_cat_SetCatEventCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_cat_SetCatEvent-Callback_ind_t * *pOutput* )

CAT event indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.4 common.h File Reference

### Data Structures

- struct pack_qmi_t
- struct unpack_qmi_t
- struct unpack_result_t

### Macros

- #define SDU_HDR_LEN (3)

- #define MINREQBKLEN (2048)
- #define MSGID_AND_LEN (4)
- #define MSGID_DONT_CARE (0xffff)
- #define UNUSEDPARAM(x) (void)x
- #define DEAULT_LOC_TIMEOUT_IN_SEC 2
- #define SDK_VALIDATE_INPUT_PACK_PARAM(pCtx, pBuf, pLen)
- #define SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID(pCtx, pBuf, pLen)
- #define SDK_VALIDATE_INPUT_UNPACK_PARAM(pResp, respLen, pOutput)

### Typedefs

- typedef void(∗ logger )(uint8_t lvl, const char ∗buff)

### Enumerations

- enum eLOG_LEVEL {
  eLOG_INFO,
  eLOG_DEBUG,
  eLOG_WARN,
  eLOG_FATAL }
- enum eTimeout {
  eTIMEOUT_2_S = 2000,
  eTIMEOUT_5_S = 5000,
  eTIMEOUT_8_S = 8000,
  eTIMEOUT_10_S = 10000,
  eTIMEOUT_20_S = 20000,
  eTIMEOUT_30_S = 30000,
  eTIMEOUT_60_S = 60000,
  eTIMEOUT_300_S = 300000,
  eTIMEOUT_DEFAULT = eTIMEOUT_8_S }
- enum eQMI_SVC {
  eCTL,
  eWDS,
  eDMS,
  eNAS =3,
  eQOS,
  eSMS =5,
  ePDS =0x06,
  eVOICE =0x09,
  eCAT =0x0A,
  eUIM =0x0B,
  eLOC =0x10,
  eSAR =0x11,
  eIMS =0x12,
  eTMD =0x18,
  eIMSA =0x21,
  eRMS =225,
  eSWIOMA =240,
  eAUDIO =241,
  eSWIM2MCMD =243,
  eSWIM2MCMD_AVC2 =0xfd,
  eSWIOMAEXT =244,
  eSWIAUDIO = 245,
  eSWILOC =246,
  eSWIDMS =254 }

- enum msgtype {
  eREQ =0,
  eRSP =2,
  eIND =4 }

## Functions

- uint16_t helper_get_xid (uint8_t ∗qmi_resp)
- uint16_t helper_get_error_code (uint8_t ∗qmi_resp)
- char ∗ helper_get_error_reason (uint16_t retVal)
- const char ∗ helper_get_resp_ctx (uint8_t svc, uint8_t ∗pbuf, uint32_t len, unpack_qmi_t ∗pCtx)
- const char ∗ helper_get_req_str (uint8_t svc, uint8_t ∗req, uint32_t len)
- unsigned unpack_result_code_only (uint8_t ∗pMdmResp)
- int helper_set_log_func (logger func)
- void liteqmi_log (uint8_t lvl, const char ∗fmt,...)
- int helper_set_log_lvl (uint8_t lvl)
- void fill_sdu_hdr (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf)
- void fill_pack_ctx (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, uint8_t svc, int timeout)
- char ∗ get_version ()
- char ∗ liteqmi_GetVersion ()
- void liteqmi_helper_decode7bitAsciiEncString (uint8_t ∗encoded_str, uint8_t encoded_len, uint8_t ∗decoded_str)

## Variables

- logger glog
- uint8_t gloglvl

### 9.4.1 Macro Definition Documentation

#### 9.4.1.1 #define DEAULT_LOC_TIMEOUT_IN_SEC 2

#### 9.4.1.2 #define MINREQBKLEN (2048)

#### 9.4.1.3 #define MSGID_AND_LEN (4)

#### 9.4.1.4 #define MSGID_DONT_CARE (0xffff)

#### 9.4.1.5 #define SDK_VALIDATE_INPUT_PACK_PARAM( *pCtx, pBuf, pLen* )

**Value:**

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

#### 9.4.1.6 #define SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID( *pCtx, pBuf, pLen* )

**Value:**

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
        {\
        liteqmi_log(eLOG_DEBUG,"[  pack] %s parameter NULL\n",__func__); \
        return eQCWWAN_ERR_INVALID_ARG; \
        } \
    if (pCtx->xid == 0) \
       return eQCWWAN_ERR_INVALID_XID; \
    pBuf[0] = eREQ; \
    pBuf[1] = pCtx->xid & 0xff; \
    pBuf[2] = pCtx->xid >> 8;
```

#### 9.4.1.7 #define SDK_VALIDATE_INPUT_UNPACK_PARAM( *pResp, respLen, pOutput* )

**Value:**

```
if ((pResp == NULL) || (respLen == 0) || (pOutput == NULL) ) \
        {\
        liteqmi_log(eLOG_DEBUG,"[ unpack] %s parameter NULL or invalid\n",
      __func__); \
        return eQCWWAN_ERR_INVALID_ARG; \
        }
```

#### 9.4.1.8 #define SDU_HDR_LEN (3)

#### 9.4.1.9 #define UNUSEDPARAM( *x* ) (void)x

### 9.4.2 Typedef Documentation

#### 9.4.2.1 typedef void(∗ logger)(uint8_t lvl, const char ∗buff)

### 9.4.3 Enumeration Type Documentation

#### 9.4.3.1 enum eLOG_LEVEL

log levels

**Enumerator**

***eLOG_INFO***

***eLOG_DEBUG***

***eLOG_WARN***

***eLOG_FATAL***

#### 9.4.3.2 enum eQMI_SVC

qmi service

**Enumerator**

***eCTL***

***eWDS***

***eDMS***

***eNAS***

***eQOS***

***eSMS***

***ePDS***

***eVOICE***

***eCAT***

***eUIM***

***eLOC***

***eSAR***

***eIMS***

***eTMD***

***eIMSA***

***eRMS***

***eSWIOMA***

***eAUDIO***

***eSWIM2MCMD***

***eSWIM2MCMD_AVC2***

***eSWIOMAEXT***

***eSWIAUDIO***

***eSWILOC***

***eSWIDMS***

#### 9.4.3.3 enum eTimeout

eTimeout

**Enumerator**

***eTIMEOUT_2_S***

***eTIMEOUT_5_S***

***eTIMEOUT_8_S***

***eTIMEOUT_10_S***

***eTIMEOUT_20_S***

***eTIMEOUT_30_S***

***eTIMEOUT_60_S***

***eTIMEOUT_300_S***

***eTIMEOUT_DEFAULT***

#### 9.4.3.4 enum msgtype

qmi message type

**Enumerator**

***eREQ***

***eRSP***

***eIND***

### 9.4.4 Function Documentation

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

#### 9.4.4.2 void fill_sdu_hdr ( pack_qmi_t * *pCtx,* uint8_t * *pReqBuf* )

#### 9.4.4.3 char* get_version ( )

**Returns**

version string

#### 9.4.4.4 uint16_t helper_get_error_code ( uint8_t * *qmi_resp* )

#### 9.4.4.5 char* helper_get_error_reason ( uint16_t *retVal* )

#### 9.4.4.6 const char* helper_get_req_str ( uint8_t *svc,* uint8_t * *req,* uint32_t *len* )

extract msgid string from modem req

**Parameters**

| | | |
|---|---|---|
| in | *svc* | qmi service |
| in | *req* | qmi request |
| in | *len* | request length |

**Returns**

qmi message string

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

extract msgid/xid/type from modem reply

**Parameters**

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

**Returns**

qmi message string

#### 9.4.4.8 uint16_t helper_get_xid ( uint8_t * *qmi_resp* )

#### 9.4.4.9 int helper_set_log_func ( logger *func* )

set log function

#### 9.4.4.10 int helper_set_log_lvl ( uint8_t *lvl* )

set log level

#### 9.4.4.11 char* liteqmi_GetVersion ( )

**Returns**

version string

#### 9.4.4.12 void liteqmi_helper_decode7bitAsciiEncString ( uint8_t * *encoded_str*, uint8_t *encoded_len*, uint8_t * *decoded_str* )

decode 7bit encoded string

**Parameters**

| | | |
|---|---|---|
| in | *encoded_str* | 7 bit encoded ASCII |
| in | *encoded_len* | encoded length |
| out | *decoded_str* | decoded ASCII string |

#### 9.4.4.13 void liteqmi_log ( uint8_t *lvl*, const char * *fmt*, ... )

#### 9.4.4.14 unsigned unpack_result_code_only ( uint8_t * *pMdmResp* )

common handler for unpacking response with TLV type 0x02 only

### 9.4.5 Variable Documentation

#### 9.4.5.1 logger glog

#### 9.4.5.2 uint8_t gloglvl

## 9.5 dms.h File Reference

### Data Structures

- struct unpack_dms_GetModelID_t
- struct unpack_dms_GetIMSI_t
- struct unpack_dms_GetFirmwareInfo_t
- struct unpack_dms_GetPower_t
- struct unpack_dms_GetSerialNumbers_t
- struct unpack_dms_GetHardwareRevision_t
- struct unpack_dms_SLQSGetBandCapability_t
- struct 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
- struct unpack_dms_SLQSDmsSwiGetPCInfo_t
- struct dms_PSMEnableStateTlv
- struct dms_PSMDurationThresholdTlv
- struct dms_PSMDurationDueToOOSTlv
- struct dms_PSMRandomizationWindowTlv
- struct dms_PSMActiveTimerTlv
- struct dms_PSMPeriodicUpdateTimerTlv
- struct dms_PSMEarlyWakeupTimeTlv
- struct unpack_dms_SLQSGetPowerSaveModeConfig_t
- struct pack_dms_SLQSSetPowerSaveModeConfig_t
- struct unpack_dms_SLQSSetPowerSaveModeConfig_t
- struct pack_dms_SetIndicationRegister_t
- struct unpack_dms_SetIndicationRegister_t
- struct dms_PSMEnableStateIndTlv
- struct dms_PSMActiveTimerIndTlv
- struct dms_PSMPeriodicUpdateTimerIndTlv
- struct unpack_dms_PSMCfgChange_ind_t
- struct pack_dms_SwiSetEventReport_t
- struct unpack_dms_SwiSetEventReport_t
- struct dms_TemperatureTlv
- struct dms_VoltageTlv
- struct unpack_dms_SwiEventReportCallBack_ind_t

## Macros

- #define DMS_UINT8_MAX_STRING_SZ 255
- #define DMS_MAX_CUST_ID_LEN 64
- #define DMS_MAX_CUST_VALUE_LEN 8
- #define DMS_IMGDETAILS_LEN 16
- #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255
- #define DMS_MAX_FWUPDATE_REF_STR_SZ 15
- #define DMS_VALID_FSN_LEN 14
- #define DMS_MAX_RADIO_IFCS_SIZE 64
- #define DMS_SLQSFWINFO_MODELID_SZ 20
- #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85
- #define DMS_SLQSFWINFO_APPVERSION_SZ 85

- #define DMS_SLQSFWINFO_SKU_SZ 15
- #define DMS_SLQSFWINFO_PACKAGEID_SZ 85
- #define DMS_SLQSFWINFO_CARRIER_SZ 20
- #define DMS_SLQSFWINFO_PRIVERSION_SZ 16
- #define DMS_SLQSFWINFO_CUR_CARR_NAME 17
- #define DMS_SLQSFWINFO_CUR_CARR_REV 13
- #define MAX_BUILD_ID_LEN 255
- #define UNIQUE_ID_LEN 16
- #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160
- #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20
- #define SPC_SIZE 6
- #define CK_MAX_SIZE 8
- #define ACT_CODE_MAX_SIZE 81
- #define ERI_DATA_MAX_SIZE 1024
- #define MEID_MAX_SIZE 8
- #define DMS_PM_ONLINE 0x00 /∗ Online ∗/
- #define DMS_PM_LOW 0x01 /∗ Low Power ∗/
- #define DMS_PM_FACTORY 0x02 /∗ Factory Test Mode ∗/
- #define DMS_PM_OFFLINE 0x03 /∗ Offline ∗/
- #define DMS_PM_RESET 0x04 /∗ Reset ∗/
- #define DMS_PM_SHUT_DOWN 0x05 /∗ Shut Down ∗/
- #define DMS_PM_PERSISTENT_LOW 0x06 /∗ Persistent Low Power ∗/
- #define DMS_SET_REPORT_ENABLE 1
- #define DMS_SET_REPORT_DISABLE 0
- #define DMS_SWI_SET_IND_ENABLE 1
- #define DMS_SWI_SET_IND_DISABLE 0
- #define DMS_SET_REG_IND_ENABLE 1
- #define DMS_SET_REG_IND_DISABLE 0
- #define DMS_SET_REG_IND_NO_CHANGE 2

## Functions

- int pack_dms_GetIMSI (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetIMSI (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetIMSI_t ∗pOutput)
- int pack_dms_GetModelID (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetModelID (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetModelID_t ∗pOutput)
- int pack_dms_GetFirmwareInfo (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetFirmwareInfo (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetFirmwareInfo_t ∗pOutput)
- int pack_dms_GetPower (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetPower (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetPower_t ∗pOutput)
- int pack_dms_GetSerialNumbers (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetSerialNumbers (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetSerialNumbers_t ∗pOutput)
- int pack_dms_GetHardwareRevision (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_GetHardwareRevision (uint8_t ∗pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t ∗pOutput)
- int pack_dms_SLQSGetBandCapability (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, void ∗reqArg)
- int unpack_dms_SLQSGetBandCapability (uint8_t ∗pResp, uint16_t respLen, unpack_dms_SLQSGetBandCapability_t ∗pOutput)
- int 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_SLQSGetERIFile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSGetERIFile (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetERIFile_t *pOutput)
- int pack_dms_SLQSUIMGetState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSUIMGetState (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSUIMGetState_t *pOutput)
- int pack_dms_SLQSSwiGetCrashInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiGetCrashInfo_t *pReq)
- int unpack_dms_SLQSSwiGetCrashInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetCrashInfo_t *pOutput)
- int pack_dms_SLQSSwiGetHostDevInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiGetHostDevInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetHostDevInfo_t *pOutput)

- int pack_dms_SLQSSwiSetHostDevInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiSetHostDevInfo_t *pReq)
- int unpack_dms_SLQSSwiSetHostDevInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiSetHostDevInfo_t *pOutput)
- int pack_dms_SLQSSwiGetOSInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiGetOSInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetOSInfo_t *pOutput)
- int pack_dms_SLQSSwiSetOSInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSwiSetOSInfo_t *pReq)
- int unpack_dms_SLQSSwiSetOSInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiSetOSInfo_t *pOutput)
- int pack_dms_SLQSSwiGetSerialNoExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_dms_SLQSSwiGetSerialNoExt (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSwiGetSerialNoExt_t *pOutput)
- int pack_dms_SLQSDmsSwiGetPCInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_SLQSDmsSwiGetPCInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetPCInfo_t *pOutput)
- int pack_dms_SLQSGetPowerSaveModeConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)
- int unpack_dms_SLQSGetPowerSaveModeConfig (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetPowerSaveModeConfig_t *pOutput)
- int pack_dms_SLQSSetPowerSaveModeConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSSetPowerSaveModeConfig_t *reqArg)
- int unpack_dms_SLQSSetPowerSaveModeConfig (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSSetPowerSaveModeConfig_t *pOutput)
- int pack_dms_SetIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetIndicationRegister_t *reqArg)
- int unpack_dms_SetIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_dms_SetIndicationRegister_t *pOutput)
- int unpack_dms_PSMCfgChange_ind (uint8_t *pResp, uint16_t respLen, unpack_dms_PSMCfgChange_ind_t *pOutput)
- int pack_dms_SwiSetEventReport (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SwiSetEventReport_t *reqArg)
- int unpack_dms_SwiSetEventReport (uint8_t *pResp, uint16_t respLen, unpack_dms_SwiSetEventReport_t *pOutput)
- int unpack_dms_SwiEventReportCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_dms_SwiEventReportCallBack_ind_t *pOutput)

## 9.5.1 Macro Definition Documentation

### 9.5.1.1 #define ACT_CODE_MAX_SIZE 81

### 9.5.1.2 #define CK_MAX_SIZE 8

### 9.5.1.3 #define DMS_IMGDETAILS_LEN 16

### 9.5.1.4 #define DMS_MAX_CUST_ID_LEN 64

### 9.5.1.5 #define DMS_MAX_CUST_VALUE_LEN 8

### 9.5.1.6 #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255

### 9.5.1.7 #define DMS_MAX_FWUPDATE_REF_STR_SZ 15

### 9.5.1.8 #define DMS_MAX_RADIO_IFCS_SIZE 64

9.5.1.9 #define DMS_PM_FACTORY 0x02 /∗ Factory Test Mode ∗/

9.5.1.10 #define DMS_PM_LOW 0x01 /∗ Low Power ∗/

9.5.1.11 #define DMS_PM_OFFLINE 0x03 /∗ Offline ∗/

9.5.1.12 #define DMS_PM_ONLINE 0x00 /∗ Online ∗/

9.5.1.13 #define DMS_PM_PERSISTENT_LOW 0x06 /∗ Persistent Low Power ∗/

9.5.1.14 #define DMS_PM_RESET 0x04 /∗ Reset ∗/

9.5.1.15 #define DMS_PM_SHUT_DOWN 0x05 /∗ Shut Down ∗/

9.5.1.16 #define DMS_SET_REG_IND_DISABLE 0

9.5.1.17 #define DMS_SET_REG_IND_ENABLE 1

9.5.1.18 #define DMS_SET_REG_IND_NO_CHANGE 2

9.5.1.19 #define DMS_SET_REPORT_DISABLE 0

9.5.1.20 #define DMS_SET_REPORT_ENABLE 1

9.5.1.21 #define DMS_SLQSFWINFO_APPVERSION_SZ 85

9.5.1.22 #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85

9.5.1.23 #define DMS_SLQSFWINFO_CARRIER_SZ 20

9.5.1.24 #define DMS_SLQSFWINFO_CUR_CARR_NAME 17

9.5.1.25 #define DMS_SLQSFWINFO_CUR_CARR_REV 13

9.5.1.26 #define DMS_SLQSFWINFO_MODELID_SZ 20

9.5.1.27 #define DMS_SLQSFWINFO_PACKAGEID_SZ 85

9.5.1.28 #define DMS_SLQSFWINFO_PRIVERSION_SZ 16

9.5.1.29 #define DMS_SLQSFWINFO_SKU_SZ 15

9.5.1.30 #define DMS_SWI_SET_IND_DISABLE 0

9.5.1.31 #define DMS_SWI_SET_IND_ENABLE 1

9.5.1.32 #define DMS_UINT8_MAX_STRING_SZ 255

9.5.1.33 #define DMS_VALID_FSN_LEN 14

9.5.1.34 #define ERI_DATA_MAX_SIZE 1024

9.5.1.35 #define MAX_BUILD_ID_LEN 255

9.5.1.36 #define MEID_MAX_SIZE 8

#### 9.5.1.37 #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160

#### 9.5.1.38 #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

#### 9.5.1.39 #define SPC_SIZE 6

#### 9.5.1.40 #define UNIQUE_ID_LEN 16

### 9.5.2 Function Documentation

#### 9.5.2.1 int pack_dms_ActivateAutomatic ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_ActivateAutomatic_t * *pReq* )

Requests the device to perform automatic service activation pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.2 int pack_dms_GetActivationState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get Activation State pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

This is only supported for 3GPP2 devices.

#### 9.5.2.3 int pack_dms_GetBandCapability ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Band Capability pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.4 int pack_dms_GetCrashAction ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Crash Action pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.5 int pack_dms_GetCustFeature ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

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

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.6 int pack_dms_GetCustFeaturesV2 ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_GetCustFeaturesV2_t * *reqArg* )

Get Custom Feature pack This function queries the modem for a list of supported features. This function is for firmware version 2.0 and newer. Currently supported Customization features:

- GPIOSARENABLE
- GPSSEL
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.7 int pack_dms_GetDeviceCap ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Device Capabilities pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.8 int pack_dms_GetDeviceCapabilities ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get device capability pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.9 int pack_dms_GetDeviceHardwareRev ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Hardware Revision of the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.10 int pack_dms_GetDeviceMfr ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Device Manufacture Name pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.11 int pack_dms_GetDeviceSerialNumbers ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Device Serial Number pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Get firmware info pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Get Firmware Revision pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Get Firmware Revisions pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.15 int pack_dms_GetFSN ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get FSN pack. This API get the Factory Sequence Number of the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.16 int pack_dms_GetHardwareRevision ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get hardware revision of the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.17 int pack_dms_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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.18 int pack_dms_GetManufacturer ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

To get device manufacturer information pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Get model id pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Get Network Time pack Returns the current time of the device based on the value supported by the network.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

To get operating mode offline reason pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.22 int pack_dms_GetPower ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get operating mode of the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.23 int pack_dms_GetPRLVersion ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get PRL Versions pack. Return version of the active Preferred Roaming List (PRL) in use by the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.24 int pack_dms_GetSerialNumbers ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get serial numbers pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.25 int pack_dms_GetUSBComp ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get USB Comp pack This API queries the modem's USB interface configuration and supported configuration parameters.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.26 int pack_dms_GetVoiceNumber ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )

Get Voice Number pack. Returns the voice number in use by the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.27 int pack_dms_ResetToFactoryDefaults ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ResetToFactoryDefaults_t * pReq )

Resets to default factory settings of the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.28 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,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.29 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,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.30 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_SetCustFeatures-V2() instead for EM74xx/MC74xx.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Set Cust Features pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.32 int pack_dms_SetEventReport ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetEventReport_t * *reqArg* )

Set Event Report pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.33 int pack_dms_SetFirmwarePreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Set Firmware Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.34 int pack_dms_SetIndicationRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetIndicationRegister_t * *reqArg* )

Set Indication register pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.35 int pack_dms_SetPower ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SetPower_t * *reqArg* )

Set Power pack Sets the operating mode of the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Set USB Comp pack This API is used to change the modem's USB interface configuration thus allowing a device to have multiple USB compositions. Devices will, by default, be configured to support a minimal set of interfaces to reduce end user modem installation time. Developers and some customers, however, require access to a custom set of interfaces. A reset is required for any change in the USB composition to take effect.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.37 int pack_dms_SLQSDmsSwiGetPCInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get device power control status information pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.38 int pack_dms_SLQSDmsSwiGetResetInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

To get reset info pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.39 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

support EM/MC74xx onwards

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.40 int pack_dms_SLQSGetBandCapability ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get band capability of the device pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.41 int pack_dms_SLQSGetERIFile ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

To set UIM Get ERI file pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.42 int pack_dms_SLQSGetPowerSaveModeConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Power Save Mode (PSM) configuration parameter pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter(NULL) |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.43 int pack_dms_SLQSSetPowerSaveModeConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SLQSSetPowerSaveModeConfig_t * *reqArg* )

Sets the Power Save Mode (PSM) configuration parameters pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.44 int pack_dms_SLQSSwiClearDyingGaspStatistics ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Clear Dying GASP Statistics pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.45 int pack_dms_SLQSSwiGetCrashInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiGetCrashInfo_t * pReq )

To Get the Crash Information from the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

Get Dying GASP Config pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.47 int pack_dms_SLQSSwiGetDyingGaspStatistics ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Get Dying GASP Statistics pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.48 int pack_dms_SLQSSwiGetFirmwareCurr ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Get currently active image pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.49 int pack_dms_SLQSSwiGetFwUpdateStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

To get Firmware Update status pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.50 int pack_dms_SLQSSwiGetHostDevInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

To get host dev information pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.51 int pack_dms_SLQSSwiGetOSInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

To Get Host OS Info pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.52 int pack_dms_SLQSSwiGetSerialNoExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

To get serial number extension pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.53 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.54 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.55 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.56 int pack_dms_SLQSUIMGetState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Returns the UIM state pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9-X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API unpack_uim_GetCardStatus/ unpack_uim-_GetCardStatus for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.57 int pack_dms_SwiSetEventReport ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_SwiSetEventReport_t * *reqArg* )

Sets the DMS swi event report parameters pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.58 int pack_dms_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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.59 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.60 int pack_dms_UIMGetICCID ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_UIMGetICCID_t * *reqArg* )

Packs the UIMGetICCID response message to a user-provided response structure. This API is deprecated on MC73xx/EM73xx modules. Since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use pack_uim_ReadTransparent()(EF ID: 3F00 2FE2) instead for new firmware versions and new modules.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.61 int pack_dms_UIMGetPINStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

To set UIM Get PIN Status pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.62 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.63 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.64 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.65 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.66 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.67 int pack_dms_ValidateSPC ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_dms_ResetToFactoryDefaults_t * *pReq* )

Requests the device to perform automatic service activation pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *pReq* | requeset parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.68 int unpack_dms_ActivateAutomatic ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_ResetToFactoryDefaults_t * *pOutput* )

Requests the device to perform automatic service activation unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.69 int unpack_dms_GetActivationState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetActivationState_t * *pOutput* )

Get Activation State unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

This is only supported for 3GPP2 devices.

#### 9.5.2.70 int unpack_dms_GetBandCapability ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetBandCapability_t * *pOutput* )

Get Band Capabilities unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.71 int unpack_dms_GetCrashAction ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetCrashAction_t * *pOutput* )

Get Crash Action unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.72 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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.73 int unpack_dms_GetCustFeaturesV2 ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetCustFeaturesV2_t** * *pOutput* )

Get Custom Feature unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.74 int unpack_dms_GetDeviceCap ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetDeviceCap_t** * *pOutput* )

Get Device Capabilities unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.75 int unpack_dms_GetDeviceCapabilities ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetDeviceCapabilities_t** * *pOutput* )

Get device capability unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.76 int unpack_dms_GetDeviceHardwareRev ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceHardwareRev_t * *pOutput* )

Get Hardware Revision of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.77 int unpack_dms_GetDeviceMfr ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceMfr_t * *pOutput* )

Get Device Manufacture Name unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.78 int unpack_dms_GetDeviceSerialNumbers ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_GetDeviceSerialNumbers_t * *pOutput* )

Get Device Serial Number unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.79 int unpack_dms_GetFirmwareInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetFirmwareInfo_t** * *pOutput* )

Get firmware info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.80 int unpack_dms_GetFirmwareRevision ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetFirmwareRevision_t** * *pOutput* )

Get Firmware Revision unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.81 int unpack_dms_GetFirmwareRevisions ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetFirmwareRevisions_t** * *pOutput* )

Get Firmware Revisions unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.82 int unpack_dms_GetFSN ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetFSN_t** * *pOutput* )

Get FSN unpack This API get the Factory Sequence Number of the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.83 int unpack_dms_GetHardwareRevision ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetHardwareRevision_t** * *pOutput* )

Get hardware revision of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.84 int unpack_dms_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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

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

To get device manufacturer information unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.86 int unpack_dms_GetModelID ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetModelID_t** * *pOutput* )

Get model id unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.87 int unpack_dms_GetNetworkTime ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetNetworkTime_t** * *pOutput* )

Get Network Time unpack. Returns the current time of the device based on the value supported by the network.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.88 int unpack_dms_GetOfflineReason ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetOfflineReason_t** * *pOutput* )

To get operating mode offline reason unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.89 int unpack_dms_GetPower ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetPower_t** * *pOutput* )

Get operating mode of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.90 int unpack_dms_GetPRLVersion ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetPRLVersion_t** * *pOutput* )

Get PRL Versions unpack Return version of the active Preferred Roaming List (PRL) in use by the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.91 int unpack_dms_GetSerialNumbers ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetSerialNumbers_t** * *pOutput* )

Get serial numbers unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.92 int unpack_dms_GetUSBComp ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetUSBComp_t** * *pOutput* )

Get USB Comp unpack This API queries the modem's USB interface configuration and supported configuration parameters.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.93 int unpack_dms_GetVoiceNumber ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_GetVoiceNumber_t** * *pOutput* )

Get Voice Number unpack Returns the voice number in use by the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.94 int unpack_dms_PSMCfgChange_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_PSMCfgChange_ind_t** * *pOutput* )

PSM Configuration change Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.95 int unpack_dms_ResetToFactoryDefaults ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_ResetToFactory-Defaults_t** * *pOutput* )

Resets to default factory settings of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.96 int unpack_dms_SetActivationStatusCallback ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SetActivationStatusCallback_t** * *pOutput* )

Set Activation status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.97 int unpack_dms_SetCrashAction ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SetCrashAction_t** * *pOutput* )

Set Crash Action unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response. Not used |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.98 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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.99 int unpack_dms_SetCustFeaturesV2 ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SetCustFeaturesV2_t** * *pOutput* )

Set Cust features unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.100 int unpack_dms_SetEventReport ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SetEventReport_t** * *pOutput* )

Set Event Report unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.101 int unpack_dms_SetEventReport_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SetEventReport_ind_t** * *pOutput* )

Event Report Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.102 int unpack_dms_SetFirmwarePreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetFirmwarePreference_t * *pOutput* )

Set Firmware Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.103 int unpack_dms_SetIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetIndication-Register_t * *pOutput* )

Set Indication register unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.104 int unpack_dms_SetPower ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SetPower_t * *pOutput* )

Set Power unpack. Sets the operating mode of the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.105 int unpack_dms_SetUSBComp ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SetUSBComp_t** * *pOutput* )

Set USB Comp unpack This API is used to change the modem's USB interface configuration thus allowing a device to have multiple USB compositions. Devices will, by default, be configured to support a minimal set of interfaces to reduce end user modem installation time. Developers and some customers, however, require access to a custom set of interfaces. A reset is required for any change in the USB composition to take effect.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.106 int unpack_dms_SLQSDmsSwiGetPCInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSDmsSwiGetPCInfo_t** * *pOutput* )

Get device power control status information unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.107 int unpack_dms_SLQSDmsSwiGetResetInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSDmsSwiGetResetInfo_t** * *pOutput* )

To get reset info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.108 int unpack_dms_SLQSDmsSwiGetResetInfo_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t** * *pOutput* )

DMS reset info Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

support EM/MC74xx onwards

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.109 int unpack_dms_SLQSDmsSwiIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSDmsSwiIndicationRegister_t** * *pOutput* )

Set the registration state for different indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

support EM/MC74xx onwards

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.110 int unpack_dms_SLQSGetBandCapability ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSGetBandCapability_t** * *pOutput* )

Get band capability of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.111 int unpack_dms_SLQSGetERIFile ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSGetERIFile_t** * *pOutput* )

To get UIM Get ERI file unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.112 int unpack_dms_SLQSGetPowerSaveModeConfig ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSGetPowerSaveModeConfig_t** * *pOutput* )

Get Power Save Mode (PSM) configuration parameter unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.113 int unpack_dms_SLQSSetPowerSaveModeConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSetPowerSaveModeConfig_t * *pOutput* )

Sets the Power Save Mode (PSM) configuration parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.114 int unpack_dms_SLQSSwiClearDyingGaspStatistics ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiClearDyingGaspStatistics_t * *pOutput* )

Clear Dying GASP Statistics unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.115 int unpack_dms_SLQSSwiGetCrashInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetCrashInfo_t * *pOutput* )

To Get the Crash Information from the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.116 int unpack_dms_SLQSSwiGetDyingGaspCfg ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspCfg_t * *pOutput* )

Get Dying GASP Config unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.117 int unpack_dms_SLQSSwiGetDyingGaspStatistics ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetDyingGaspStatistics_t * *pOutput* )

Get Dying GASP Statistics unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.118 int unpack_dms_SLQSSwiGetFirmwareCurr ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetFirmwareCurr_t * *pOutput* )

Get currently active image unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.119 int unpack_dms_SLQSSwiGetFwUpdateStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetFwUpdateStatus_t * *pOutput* )

To get Firmware Update status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.120 int unpack_dms_SLQSSwiGetHostDevInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetHostDevInfo_t * *pOutput* )

To get host dev information unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.121 int unpack_dms_SLQSSwiGetOSInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetOSInfo_t * *pOutput* )

To get Host OS Info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.122 int unpack_dms_SLQSSwiGetSerialNoExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiGetSerialNoExt_t * *pOutput* )

To get serial number extension unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.123 int unpack_dms_SLQSSwiSetDyingGaspCfg ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiSetDyingGaspCfg_t * *pOutput* )

Set Dying GASP Config unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.124 int unpack_dms_SLQSSwiSetHostDevInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_SLQSSwiSetHostDevInfo_t * *pOutput* )

To set Host Dev Info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.125 int unpack_dms_SLQSSwiSetOSInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSSwiSetOSInfo_t** * *pOutput* )

To set Host OS Info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.126 int unpack_dms_SLQSUIMGetState ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SLQSUIMGetState_t** * *pOutput* )

Returns the UIM state unpack. This API is deprecated on MC73xx/EM73xx modules since firmware version SW-I9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API unpack_uim_GetCardStatus / unpack_-uim_GetCardStatus for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.127 int unpack_dms_SwiEventReportCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SwiEventReportCallBack_ind_t** * *pOutput* )

unpack for DMS swi event report indication

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.128 int unpack_dms_SwiSetEventReport ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_SwiSetEventReport_t** * *pOutput* )

Sets the swi event report parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.129 int unpack_dms_UIMChangePIN ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_UIMSetPINProtection_t** * *pOutput* )

To get UIM change PIN unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.130 int unpack_dms_UIMGetControlKeyStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMGetControlKeyStatus_t * *pOutput* )

To get UIM Get ControlKeyStatus unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.131 int unpack_dms_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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.132 int unpack_dms_UIMGetPINStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMGetPINStatus_t * *pOutput* )

To get UIM Get PIN Status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.133 int unpack_dms_UIMSetControlKeyProtection ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetControlKeyProtection_t * *pOutput* )

To get UIM Set Control Key Protection unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.134 int unpack_dms_UIMSetPINProtection ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMSetPINProtection_t * *pOutput* )

To get UIM Set PIN protection unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.135 int unpack_dms_UIMUnblockControlKey ( uint8_t * *pResp*, uint16_t *respLen*, unpack_dms_UIMUnblockControlKey_t * *pOutput* )

To get UIM Unblock Control Key unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.136 int unpack_dms_UIMUnblockPIN ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_UIMSetPINProtection_t** * *pOutput* )

To get UIM unblock PIN unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.137 int unpack_dms_UIMVerifyPIN ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_UIMSetPINProtection_t** * *pOutput* )

To get UIM verify PIN unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.5.2.138 int unpack_dms_ValidateSPC ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_dms_ResetToFactoryDefaults_t** * *pOutput* )

Requests the device to perform automatic service activation unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.6 fms.h File Reference

### Data Structures

- struct CarrierImage_t
- struct pack_fms_GetImagesPreference_t
- struct FMSImageElement
- struct FMSPrefImageList
- struct unpack_fms_GetImagesPreference_t
- struct pack_fms_GetStoredImages_t
- struct FMSImageIdElement
- struct FMSImageIDEntries
- struct FMSImageList
- struct unpack_fms_GetStoredImages_t
- struct pack_fms_SetImagesPreference_t
- struct unpack_fms_SetImagesPreference_t

### Macros

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

### Functions

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

## 9.6.1 Macro Definition Documentation

### 9.6.1.1 #define FMS_FW_PRI_BUILD_MATCH_LEN 11

### 9.6.1.2 #define FMS_GOBI_LISTENTRIES_MAX 2

### 9.6.1.3 #define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100

### 9.6.1.4 #define FMS_GOBI_MBN_IMG_ID_STR_LEN 16

### 9.6.1.5 #define FMS_IMAGE_ID_BUILD_ID_LEN 32

### 9.6.1.6 #define FMS_IMAGE_ID_IMG_ID_LEN 16

### 9.6.1.7 #define FMS_IMAGE_ID_MAX_ENTRIES 2

### 9.6.1.8 #define FMS_IMAGE_ID_PRI_IMGTYPE 0x01

### 9.6.1.9 #define FMS_MAX_IMAGE_ID_ELEMENT 50

### 9.6.1.10 #define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255

## 9.6.2 Function Documentation

### 9.6.2.1 uint32_t GetValidFwPriCombinations ( **FMSImageList** ∗ *pStoredImageList,* uint32_t ∗ *pValidCombinationSize,* **CarrierImage_t** ∗ *pValidCombinations* )

This API distills valid Firmware/PRI combinations from GetStoredImages result

**Parameters**

| | | |
|---|---|---|
| in | *pStoredImageList* | • image list returned from GetStoredImages<br>• See FMSImageList |
| in,out | *pValidCombinationSize* | • number of combination passed in and returned |
| out | *pValidCombinations* | • valid combinations returned<br>• See CarrierImage_t |

**Returns**

- eQCWWAN_ERR_INVALID_ARG - Invalid parameters
- eQCWWAN_ERR_BUFFER_SZ - No enough element to store combinatons returned

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.6.2.2 int pack_fms_GetImagesPreference ( **pack_qmi_t** ∗ *pCtx,* uint8_t ∗ *pReqBuf,* uint16_t ∗ *pLen,* **pack_fms_GetImagesPreference_t** ∗ *reqArg* )

Get Images Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.6.2.3 int pack_fms_GetStoredImages ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_GetStoredImages_t * *reqArg* )

Get Images Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.6.2.4 int pack_fms_SetImagesPreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_fms_SetImagesPreference_t * *reqArg* )

Set Images Preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.6.2.5 int unpack_fms_GetImagesPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_GetImagesPreference_t * *pOutput* )

Get Images Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

#### 9.6.2.6 int unpack_fms_GetStoredImages ( uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_GetStoredImages_t * *pOutput* )

Get Images Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

#### 9.6.2.7 int unpack_fms_SetImagesPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_fms_SetImagesPreference_t * *pOutput* )

Set Images Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

## 9.7 ims.h File Reference

### Data Structures

- struct pack_ims_SLQSSetSIPConfig_t
- struct unpack_ims_SLQSSetSIPConfig_t
- struct pack_ims_SLQSSetRegMgrConfig_t
- struct unpack_ims_SLQSSetRegMgrConfig_t
- struct pack_ims_SLQSSetIMSSMSConfig_t
- struct unpack_ims_SLQSSetIMSSMSConfig_t
- struct pack_ims_SLQSSetIMSUserConfig_t
- struct unpack_ims_SLQSSetIMSUserConfig_t
- struct pack_ims_SLQSSetIMSVoIPConfig_t
- struct unpack_ims_SLQSSetIMSVoIPConfig_t
- struct unpack_ims_SLQSGetSIPConfig_t
- struct unpack_ims_SLQSGetRegMgrConfig_t
- struct unpack_ims_SLQSGetIMSSMSConfig_t
- struct unpack_ims_SLQSGetIMSUserConfig_t
- struct unpack_ims_SLQSGetIMSVoIPConfig_t
- struct pack_ims_SLQSImsConfigIndicationRegister_t
- struct ims_PCSCFPortInfo
- struct ims_CSCFPortNameInfo
- struct ims_IMSTestModeInfo
- struct unpack_ims_SLQSRegMgrCfgCallBack_ind_t
- struct ims_SIPPortInfo
- struct ims_SIPRegnTmrInfo
- struct ims_SubscrTmrInfo
- struct ims_TmrT1Info
- struct ims_TmrT2Info
- struct ims_TmrTfInfo
- struct ims_SigCompEnInfo
- struct unpack_ims_SLQSSIPCfgCallBack_ind_t
- struct ims_SMSFmtInfo
- struct ims_SMSoIPNwInfo
- struct ims_PhCtxtURIInfo
- struct unpack_ims_SLQSSMSCfgCallBack_ind_t
- struct ims_IMSDomainInfo
- struct unpack_ims_SLQSUserCfgCallBack_ind_t
- struct ims_SessDurInfo
- struct ims_MinSessExpInfo
- struct ims_EnabAMRWBInfo
- struct ims_EnabSCRAMRInfo
- struct ims_EnabSCRAMRWBInfo

- struct ims_AMRModeInfo
- struct ims_AMRWBModeInfo
- struct ims_AMROctAlgnInfo
- struct ims_AMRWBOctAlgnInfo
- struct ims_RngTmrInfo
- struct ims_RngBkTmrInfo
- struct ims_RTPRTCPInactTmrDurInfo
- struct unpack_ims_SLQSVoIPCfgCallBack_ind_t

## Macros

- #define MAX_NAME_LEN 255

## Typedefs

- typedef unpack_result_t unpack_ims_SLQSImsConfigIndicationRegister_t

## Functions

- int pack_ims_SLQSSetSIPConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_ims_SLQSSetSIPConfig_t *pReqParam)
- int unpack_ims_SLQSSetSIPConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSetSIPConfig_t *pOutput)
- int pack_ims_SLQSSetRegMgrConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_ims_SLQSSetRegMgrConfig_t *pReqParam)
- int unpack_ims_SLQSSetRegMgrConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSetRegMgrConfig_t *pOutput)
- int pack_ims_SLQSSetIMSSMSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_ims_SLQSSetIMSSMSConfig_t *pReqParam)
- int unpack_ims_SLQSSetIMSSMSConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSetIMSSMSConfig_t *pOutput)
- int pack_ims_SLQSSetIMSUserConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_ims_SLQSSetIMSUserConfig_t *pReqParam)
- int unpack_ims_SLQSSetIMSUserConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSetIMSUserConfig_t *pOutput)
- int pack_ims_SLQSSetIMSVoIPConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_ims_SLQSSetIMSVoIPConfig_t *pReqParam)
- int unpack_ims_SLQSSetIMSVoIPConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSetIMSVoIPConfig_t *pOutput)
- int pack_ims_SLQSGetSIPConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_ims_SLQSGetSIPConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSGetSIPConfig_t *pOutput)
- int pack_ims_SLQSGetRegMgrConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_ims_SLQSGetRegMgrConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSGetRegMgrConfig_t *pOutput)
- int pack_ims_SLQSGetIMSSMSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_ims_SLQSGetIMSSMSConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSGetIMSSMSConfig_t *pOutput)
- int pack_ims_SLQSGetIMSUserConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_ims_SLQSGetIMSUserConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSGetIMSUserConfig_t *pOutput)
- int pack_ims_SLQSGetIMSVoIPConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_ims_SLQSGetIMSVoIPConfig (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSGetIMSVoIPConfig_t *pOutput)

- int pack_ims_SLQSImsConfigIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_ims_SLQSImsConfigIndicationRegister_t *pReqParam)
- int unpack_ims_SLQSImsConfigIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSImsConfigIndicationRegister_t *pOutput)
- int unpack_ims_SLQSRegMgrCfgCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSRegMgrCfgCallBack_ind_t *pOutput)
- int unpack_ims_SLQSSIPCfgCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSIPCfgCallBack_ind_t *pOutput)
- int unpack_ims_SLQSSMSCfgCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSSMSCfgCallBack_ind_t *pOutput)
- int unpack_ims_SLQSUserCfgCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSUserCfgCallBack_ind_t *pOutput)
- int unpack_ims_SLQSVoIPCfgCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_ims_SLQSVoIPCfgCallBack_ind_t *pOutput)

## 9.7.1 Macro Definition Documentation

### 9.7.1.1 #define MAX_NAME_LEN 255

## 9.7.2 Typedef Documentation

### 9.7.2.1 typedef unpack_result_t unpack_ims_SLQSImsConfigIndicationRegister_t

## 9.7.3 Function Documentation

### 9.7.3.1 int pack_ims_SLQSGetIMSSMSConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Retrieves the SMS configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

### 9.7.3.2 int pack_ims_SLQSGetIMSUserConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Retrieves the IMS User configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.3 int pack_ims_SLQSGetIMSVoIPConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Retrieves the IMS VoIP configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.4 int pack_ims_SLQSGetRegMgrConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.5 int pack_ims_SLQSGetSIPConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.6 int pack_ims_SLQSImsConfigIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSImsConfigIndicationRegister_t * pReqParam )

Sets the registration state for different QMI_IMS indications for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

```
- This function is used by a device to register/deregister for different
  QMI IMS indications.
- The device's registration state variables that control registration
  for indications will be modified to reflect the settings indicated in the
  request message.
- At least one optional parameter must be present in the request.
```

#### 9.7.3.7 int pack_ims_SLQSSetIMSSMSConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSSetIMSSMSConfig_t * pReqParam )

Sets the IMS SMS configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.8 int pack_ims_SLQSSetIMSUserConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_ims_SLQSSetIMSUserConfig_t * *pReqParam* )

Sets the IMS user configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.9 int pack_ims_SLQSSetIMSVoIPConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_ims_SLQSSetIMSVoIPConfig_t * *pReqParam* )

Sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.10 int pack_ims_SLQSSetRegMgrConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_ims_SLQSSetRegMgrConfig_t * *pReqParam* )

Sets the IMS registration manager configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.11 int pack_ims_SLQSSetSIPConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_ims_SLQSSetSIPConfig_t * *pReqParam* )

Sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

#### 9.7.3.12 int unpack_ims_SLQSGetIMSSMSConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSGetIMSSMSConfig_t * *pOutput* )

Retrieves the SMS configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.13 int unpack_ims_SLQSGetIMSUserConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSGetIMSUserConfig_t * *pOutput* )

Retrieves the IMS User configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.14 int unpack_ims_SLQSGetIMSVoIPConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSGetIMSVoIPConfig_t * *pOutput* )

Retrieves the IMS VoIP configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.15 int unpack_ims_SLQSGetRegMgrConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSGetRegMgrConfig_t * *pOutput* )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.16 int unpack_ims_SLQSGetSIPConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSGetSIPConfig_t * *pOutput* )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.17 int unpack_ims_SLQSImsConfigIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSImsConfigIndicationRegister_t * *pOutput* )

Sets the registration state for different QMI_IMS indications for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.18 int unpack_ims_SLQSRegMgrCfgCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSRegMgrCfgCallBack_ind_t * *pOutput* )

Unpack indication about IMS registration manager configuration

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.19 int unpack_ims_SLQSSetIMSSMSConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSSetIMSSMSConfig_t * *pOutput* )

Sets the IMS SMS configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.20 int unpack_ims_SLQSSetIMSUserConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSSetIMSUserConfig_t * *pOutput* )

Sets the IMS user configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.21 int unpack_ims_SLQSSetIMSVoIPConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSSetIMSVoIP-Config_t * *pOutput* )

Sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.22 int unpack_ims_SLQSSetRegMgrConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSSetRegMgr-Config_t * *pOutput* )

Sets the IMS registration manager configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.23 int unpack_ims_SLQSSetSIPConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSSetSIPConfig_t * *pOutput* )

Sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.24 int unpack_ims_SLQSSIPCfgCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_ims_SLQSSIPCfgCallBack_ind_t** * *pOutput* )

Unpack indication about SIP configuration info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.25 int unpack_ims_SLQSSMSCfgCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_ims_SLQSSMSCfgCallBack_ind_t** * *pOutput* )

Unpack indication about SMS configuration info for IMS

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.26 int unpack_ims_SLQSUserCfgCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_ims_SLQSUserCfgCallBack_ind_t** * *pOutput* )

Unpack indication about User configuration info for IMS

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.7.3.27 int unpack_ims_SLQSVoIPCfgCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_ims_SLQSVoIPCfgCallBack_ind_t * *pOutput* )

Unpack indication about VOIP configuration info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.8 imsa.h File Reference

### Data Structures

- struct pack_imsa_SLQSRegisterIMSAIndication_t
- struct unpack_imsa_SLQSGetIMSARegStatus_t
- struct unpack_imsa_SLQSGetIMSAServiceStatus_t
- struct imsa_SmsSvcStatusInfo
- struct imsa_SmsRatInfo
- struct imsa_VoipSvcStatusInfo
- struct imsa_VoipRatInfo
- struct imsa_VtSvcStatusInfo
- struct imsa_VtRatInfo
- struct imsa_UtSvcStatusInfo
- struct imsa_UtRatInfo
- struct unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t
- struct imsa_IMSRegStatusInfo
- struct imsa_IMSRegStatusErrorCodeInfo
- struct imsa_NewIMSRegStatusInfo
- struct unpack_imsa_SLQSImsaRegStatusCallBack_ind_t
- struct imsa_RatHandoverStatusInfo
- struct unpack_imsa_SLQSImsaRatStatusCallBack_ind_t
- struct imsa_IMSFailErrCodeTlv
- struct unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t

### Macros

- #define MAX_ERROR_CODE_LEN 256

### Typedefs

- typedef unpack_result_t unpack_imsa_SLQSRegisterIMSAIndication_t

### Functions

- int pack_imsa_SLQSRegisterIMSAIndication (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_imsa_SLQSRegisterIMSAIndication_t ∗reqParam)
- int unpack_imsa_SLQSRegisterIMSAIndication (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSRegisterIMSAIndication_t ∗pOutput)
- int pack_imsa_SLQSGetIMSARegStatus (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_imsa_SLQSGetIMSARegStatus (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSGetIMSARegStatus_t ∗pOutput)
- int pack_imsa_SLQSGetIMSAServiceStatus (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_imsa_SLQSGetIMSAServiceStatus (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSGetIMSAServiceStatus_t ∗pOutput)
- int unpack_imsa_SLQSImsaSvcStatusCallBack_ind (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t ∗pOutput)
- int unpack_imsa_SLQSImsaRegStatusCallBack_ind (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSImsaRegStatusCallBack_ind_t ∗pOutput)
- int unpack_imsa_SLQSImsaRatStatusCallBack_ind (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSImsaRatStatusCallBack_ind_t ∗pOutput)
- int unpack_imsa_SLQSImsaPdpStatusCallBack_ind (uint8_t ∗pResp, uint16_t respLen, unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t ∗pOutput)

## 9.8.1 Macro Definition Documentation

#### 9.8.1.1 #define MAX_ERROR_CODE_LEN 256

## 9.8.2 Typedef Documentation

#### 9.8.2.1 typedef unpack_result_t unpack_imsa_SLQSRegisterIMSAIndication_t

## 9.8.3 Function Documentation

#### 9.8.3.1 int pack_imsa_SLQSGetIMSARegStatus ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen* )

Function to pack Get IMSA Registration Status command.

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.2 int pack_imsa_SLQSGetIMSAServiceStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Function to pack Get IMSA Service Status command

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.3 int pack_imsa_SLQSRegisterIMSAIndication ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_imsa_SLQSRegisterIMSAIndication_t * *reqParam* )

Function to pack Register IMSA Indication command

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [IN]<br>• See pack_imsa_SLQSRegisterIMSAIndication_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.4 int unpack_imsa_SLQSGetIMSARegStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_imsa_SLQSGetIMSA-RegStatus_t * *pOutput* )

Function to unpack Get IMSA Registration Status response from modem. *

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• See unpack_imsa_SLQSGetIMSARegStatus_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.5 int unpack_imsa_SLQSGetIMSAServiceStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_imsa_SLQSGetIMSAServiceStatus_t * *pOutput* )

Function to unpack Get IMSA Service Status response from modem

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• See unpack_imsa_SLQSGetIMSAServiceStatus_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.6 int unpack_imsa_SLQSImsaPdpStatusCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t** * *pOutput* )

Unpack indication about IMSA PDP status info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.7 int unpack_imsa_SLQSImsaRatStatusCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_imsa_SLQSImsaRatStatusCallBack_ind_t** * *pOutput* )

Unpack indication about IMSA RAT status info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.8 int unpack_imsa_SLQSImsaRegStatusCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_imsa_SLQSImsaRegStatusCallBack_ind_t** * *pOutput* )

Unpack indication about IMSA registration status info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.9 int unpack_imsa_SLQSImsaSvcStatusCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t * *pOutput* )

Unpack indication about IMSA service status info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.8.3.10 int unpack_imsa_SLQSRegisterIMSAIndication ( uint8_t * *pResp*, uint16_t *respLen*, unpack_imsa_SLQSRegisterIMSAIndication_t * *pOutput* )

Function to unpack Register IMSA Indication response from modem

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• See unpack_imsa_SLQSRegisterIMSAIndication_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.9 lite-fw.h File Reference

## Data Structures

- struct _litefw_FirmwareInfo_

## Macros

- #define IMG_MASK_MDM (1<<0)
- #define IMG_MASK_PRI (1<<1)
- #define IMG_MASK_CLEAR (0x000)
- #define IMG_MASK_GENERIC (0xFFFF)
- #define LITEFW_CARRIER_PACKAGE_SKU "9999999"
- #define LITEFW_SKU_STRING_LENGTH 7
- #define FIRMWARE_INFO_STRING_SIZE 20
- #define libSDP_GetVersion litefw_GetVersion
- #define libSDP_CalculateImageMask litefw_CalculateImageMask
- #define libSDP_getFileType litefw_getFileType
- #define libSDP_ExtractFirmwareParametersByPath litefw_ExtractFirmwareParametersByPath
- #define libSDP_GetModelFamily litefw_GetModelFamily
- #define libSDP_CheckValidFirmwareInfo litefw_CheckValidFirmwareInfo
- #define libSDP_BuildImagesPreferenceRequest litefw_BuildImagesPreferenceRequest
- #define libSDP_DownloadFW litefw_DownloadFW
- #define libsdp_set_log_func litefw_set_log_func
- #define libsdp_SetReadBlockSize litefw_SetReadBlockSize
- #define LIBSDP_CARRIER_PACKAGE_SKU LITEFW_CARRIER_PACKAGE_SKU
- #define LIBSDP_SKU_STRING_LENGTH LITEFW_SKU_STRING_LENGTH

## Typedefs

- typedef struct _litefw_FirmwareInfo_ litefw_FirmwareInfo
- typedef void(∗ litefwlogger )(uint8_t lvl, const char ∗buff)
- typedef litefw_FirmwareInfo libSDP_FirmwareInfo
- typedef litefwlogger libsdplogger

## Enumerations

- enum litefw_fwdwl_error_codes {
  eSDP_FWDWL_SUCCESS = 0,
  eSDP_FWDWL_ERR_GENERAL = 100,
  eSDP_FWDWL_ERR_SDK,
  eSDP_FWDWL_ERR_SET_CBK,
  eSDP_FWDWL_ERR_PATH_TOO_LONG,
  eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED,
  eSDP_FWDWL_ERR_FW_UPGRADE,
  eSDP_FWDWL_ERR_INVALID_DEV,
  eSDP_FWDWL_ERR_INVALID_PATH,
  eSDP_FWDWL_ERR_TIMEOUT,
  eSDP_FWDWL_ERR_FAIL,
  eSDP_FWDWL_ERR_PRI_FAIL,
  eSDP_FWDWL_ERR_FW_VERSION_FAIL,
  eSDP_FWDWL_ERR_SDP_TIMEOUT,
  eSDP_FWDWL_ERR_END }

- enum litefw_Fw_Type {
  eFW_TYPE_MBN_GOBI,
  eFW_TYPE_MBN,
  eFW_TYPE_CWE,
  eFW_TYPE_NVU,
  eFW_TYPE_SPK,
  eFW_TYPE_INVALID,
  eFW_TYPE_CWE_NVU }
- enum litefw_Models {
  eModel_Unknown =-1,
  eModel_9X15 =0,
  eModel_WP9X15 =1,
  eModel_9X30 =2,
  eModel_9x07 =3,
  eModel_9x06 =4 }
- enum litefw_QDL_MODEs {
  eQDL_MODE_Unknown =-1,
  eQDL_MODE_INIT =0,
  eQDL_MODE_TTYUSB =1,
  eQDL_MODE_UART0 =2 }
- enum litefw_QDL_FLOW_CONTROLs {
  eQDL_HW_FLOW_Unknown =-1,
  eQDL_HW_FLOW_INIT =0,
  eQDL_HW_FLOW_ENABLE =1,
  eQDL_HW_FLOW_DISABLE =2 }

### Functions

- void litefw_logsenable (int log_en)
- int litefw_ExtractFirmwareParametersByPath (char ∗pImagePath, litefw_FirmwareInfo ∗info)
- int litefw_BuildImagesPreferenceRequest (litefw_FirmwareInfo info, pack_fms_SetImagesPreference_t ∗pack)
- int litefw_CalculateImageMask (unpack_fms_SetImagesPreference_t SetPrefRspFromModem)
- int litefw_getFileType (char ∗szPath)
- unsigned int litefw_DownloadFW (char ∗pImagePath, char ∗szTTYPath, int iFWImageType, int image_mask, int iModelFamily)
- int litefw_GetModelFamily (char ∗pModelString)
- int litefw_CheckValidFirmwareInfo (litefw_FirmwareInfo info)
- char ∗ litefw_GetVersion ()
- int litefw_set_log_func (litefwlogger func)
- void litefw_SetReadBlockSize (unsigned long lBlockSize)
- int litefw_switch_9x07_to_downloadmode (char ∗szTTYPath)
- int litefw_switch_to_BootHoldMode (char ∗szTTYPath)
- int litefw_SetQTLDownloadMode (int iMode)
- int litefw_GetQTLDownloadMode ()
- int litefw_SetQTLHWFlowControl (int iMode)
- int litefw_GetQTLHWFlowControl ()

### 9.9.1 Detailed Description

Filename: lite-fw.h

Purpose: Global definitions used inside the SDK

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

## 9.9.2 Macro Definition Documentation

### 9.9.2.1 #define FIRMWARE_INFO_STRING_SIZE 20

### 9.9.2.2 #define IMG_MASK_CLEAR (0x000)

### 9.9.2.3 #define IMG_MASK_GENERIC (0xFFFF)

### 9.9.2.4 #define IMG_MASK_MDM (1<<0)

### 9.9.2.5 #define IMG_MASK_PRI (1<<1)

### 9.9.2.6 #define libSDP_BuildImagesPreferenceRequest litefw_BuildImagesPreferenceRequest

### 9.9.2.7 #define libSDP_CalculateImageMask litefw_CalculateImageMask

### 9.9.2.8 #define LIBSDP_CARRIER_PACKAGE_SKU LITEFW_CARRIER_PACKAGE_SKU

### 9.9.2.9 #define libSDP_CheckValidFirmwareInfo litefw_CheckValidFirmwareInfo

### 9.9.2.10 #define libSDP_DownloadFW litefw_DownloadFW

### 9.9.2.11 #define libSDP_ExtractFirmwareParametersByPath litefw_ExtractFirmwareParametersByPath

### 9.9.2.12 #define libSDP_getFileType litefw_getFileType

### 9.9.2.13 #define libSDP_GetModelFamily litefw_GetModelFamily

### 9.9.2.14 #define libSDP_GetVersion litefw_GetVersion

### 9.9.2.15 #define libsdp_set_log_func litefw_set_log_func

### 9.9.2.16 #define libsdp_SetReadBlockSize litefw_SetReadBlockSize

### 9.9.2.17 #define LIBSDP_SKU_STRING_LENGTH LITEFW_SKU_STRING_LENGTH

### 9.9.2.18 #define LITEFW_CARRIER_PACKAGE_SKU "9999999"

### 9.9.2.19 #define LITEFW_SKU_STRING_LENGTH 7

## 9.9.3 Typedef Documentation

### 9.9.3.1 typedef litefw_FirmwareInfo libSDP_FirmwareInfo

### 9.9.3.2 typedef litefwlogger libsdplogger

### 9.9.3.3 typedef struct _litefw_FirmwareInfo_ litefw_FirmwareInfo

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

**Parameters**

| | |
|---|---|
| *szModelid_str* | • Model Name String |

| | |
|---|---|
| *szFwversion_str* | • Firmware Version String. |
| *szSku_str* | • SKU String. |
| *szPackageid_str* | • Package ID String. |
| *szCarrier_str* | • Carrier String. |
| *szCarrier-Priversion_str* | • Carrier PRI Version String. |

#### 9.9.3.4 typedef void(∗ litefwlogger)(uint8_t lvl, const char ∗buff)

This Custom Log prototype.

**Parameters**

| | | |
|---|---|---|
| in | *lvl* | • Log level. |
| in | *buff* | • Log String. |

**Returns**

none

### 9.9.4 Enumeration Type Documentation

#### 9.9.4.1 enum litefw_Fw_Type

**Enumerator**

***eFW_TYPE_MBN_GOBI***

***eFW_TYPE_MBN***

***eFW_TYPE_CWE***

***eFW_TYPE_NVU***

***eFW_TYPE_SPK***

***eFW_TYPE_INVALID***

***eFW_TYPE_CWE_NVU***

#### 9.9.4.2 enum litefw_fwdwl_error_codes

**Enumerator**

***eSDP_FWDWL_SUCCESS***

*eSDP_FWDWL_ERR_GENERAL*

*eSDP_FWDWL_ERR_SDK*

*eSDP_FWDWL_ERR_SET_CBK*

*eSDP_FWDWL_ERR_PATH_TOO_LONG*

*eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED*

*eSDP_FWDWL_ERR_FW_UPGRADE*

*eSDP_FWDWL_ERR_INVALID_DEV*

*eSDP_FWDWL_ERR_INVALID_PATH*

*eSDP_FWDWL_ERR_TIMEOUT*

*eSDP_FWDWL_ERR_FAIL*

*eSDP_FWDWL_ERR_PRI_FAIL*

*eSDP_FWDWL_ERR_FW_VERSION_FAIL*

*eSDP_FWDWL_ERR_SDP_TIMEOUT*

*eSDP_FWDWL_ERR_END*

#### 9.9.4.3 enum litefw_Models

**Enumerator**

*eModel_Unknown*

*eModel_9X15*

*eModel_WP9X15*

*eModel_9X30*

*eModel_9x07*

*eModel_9x06*

#### 9.9.4.4 enum litefw_QDL_FLOW_CONTROLs

**Enumerator**

*eQDL_HW_FLOW_Unknown*

*eQDL_HW_FLOW_INIT*

*eQDL_HW_FLOW_ENABLE*

*eQDL_HW_FLOW_DISABLE*

#### 9.9.4.5 enum litefw_QDL_MODEs

**Enumerator**

*eQDL_MODE_Unknown*

*eQDL_MODE_INIT*

*eQDL_MODE_TTYUSB*

*eQDL_MODE_UART0*

### 9.9.5 Function Documentation

#### 9.9.5.1 int litefw_BuildImagesPreferenceRequest ( litefw_FirmwareInfo *info*, pack_fms_SetImagesPreference_t * *pack* )

This API Build Image Preference Request Using Firmware Information.

**Parameters**

| | | |
|---|---|---|
| in | *info* | • See litefw_FirmwareInfo for more information. |
| out | *pack* | • See pack_fms_SetImagesPreference_t for more information. |

**Returns**

0 on success, litefw_fwdwl_error_codes error value otherwise

**See Also**

See litefw_fwdwl_error_codes for error values

#### 9.9.5.2 int litefw_CalculateImageMask ( unpack_fms_SetImagesPreference_t *SetPrefRspFromModem* )

This API Calculate Image Mask for Firmware Download.

**Parameters**

| | | |
|---|---|---|
| in | *SetPrefRsp-FromModem* | • See unpack_fms_SetImagesPreference_t for more information. |

**Returns**

Image Mask

- IMG_MASK_MDM | IMG_MASK_PRI

#### 9.9.5.3 int litefw_CheckValidFirmwareInfo ( litefw_FirmwareInfo *info* )

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

**Parameters**

| | | |
|---|---|---|
| in | *info* | • See litefw_FirmwareInfo for more information. |

**Returns**

0 on success, -1 error value otherwise

#### 9.9.5.4 unsigned int litefw_DownloadFW ( char * *pImagePath*, char * *szTTYPath*, int *iFWImageType*, int *image_mask*, int *iModelFamily* )

This API Download Firmware.

**Parameters**

| | | |
|---|---|---|
| in | *pImagePath* | • Firmware Folder Path. |
| in | *szTTYPath* | • QDL Device Path. |
| in | *iFWImageType* | • Firmware Type.<br>• See litefw_Fw_Type |
| in | *image_mask* | • Image Mask.<br>– IMG_MASK_MDM \| IMG_MASK_PRI |
| in | *iModelFamily* | • Modem Family.<br>• See litefw_Models |

**Returns**

0 on success, litefw_fwdwl_error_codes error value otherwise

**See Also**

See litefw_fwdwl_error_codes for error values

#### 9.9.5.5 int litefw_ExtractFirmwareParametersByPath ( char * *pImagePath*, **litefw_FirmwareInfo** * *info* )

This API Extrace Firmware Parameters From Path.

**Parameters**

| | | |
|---|---|---|
| in | *pImagePath* | • Firmware Folder Path. |
| out | *info* | • See litefw_FirmwareInfo for more information. |

**Returns**

0 on success, litefw_fwdwl_error_codes error value otherwise

**See Also**

See litefw_fwdwl_error_codes for error values

#### 9.9.5.6 int litefw_getFileType ( char * *szPath* )

This API Get File Type By Path

**Parameters**

| | | |
|---|---|---|
| in | *szPath* | • See litefw_FirmwareInfo for more information. |

**Returns**

eFW_TYPE_INVALID on error, litefw_Fw_Type value otherwise

**See Also**

See litefw_Fw_Type for values

#### 9.9.5.7 int litefw_GetModelFamily ( char * *pModelString* )

This API Get Model Famliy from a model string.

**Parameters**

| | | |
|---|---|---|
| in | *pModelString* | • Model String. |

**Returns**

eModel_Unknown on Error, litefw_Models value otherwise

**See Also**

See litefw_Models for values

#### 9.9.5.8 int litefw_GetQTLDownloadMode ( )

This API Get QDL port donwload mode.

**Parameters**

| | | |
|---|---|---|
| in | *iMode* | • litefw_QDL_MODEs |

**Returns**

eQDL_MODE_Unknown on Error, litefw_QDL_MODEs value otherwise

#### 9.9.5.9 int litefw_GetQTLHWFlowControl ( )

This API Get QDL port Hardware flow control.

**Returns**

eQDL_HW_FLOW_Unknown on Error, litefw_QDL_FLOW_CONTROLs value otherwise

#### 9.9.5.10 char* litefw_GetVersion ( )

This API Get Lib SDP Version.

**Returns**

Version String

#### 9.9.5.11 void litefw_logsenable ( int *log_en* )

This function enable/disable lite firmware debug logs.

**Parameters**

| | | |
|---|---|---|
| in | *log_en* | • Logs enable/disable flag. |

#### 9.9.5.12 int litefw_set_log_func ( **litefwlogger** *func* )

This API Set Custom Log function.

**Parameters**

| | | |
|---|---|---|
| in | *func* | • See litefwlogger for more information. |

**Returns**

none

#### 9.9.5.13 int litefw_SetQTLDownloadMode ( int *iMode* )

This API set QDL port donwload mode.

**Parameters**

| | | |
|---|---|---|
| in | *iMode* | • QDL Download Mode.<br>– litefw_QDL_MODEs |

**Returns**

0 on success, -1 error value otherwise

#### 9.9.5.14 int litefw_SetQTLHWFlowControl ( int *iMode* )

This API Set QDL port Hardware flow control.

**Parameters**

| | | |
|---|---|---|
| in | *iMode* | • litefw_QDL_FLOW_CONTROLs |

**Returns**

0 on success, -1 error value otherwise

#### 9.9.5.15 void litefw_SetReadBlockSize ( unsigned long *lBlockSize* )

This API Set Read BlockSize.

**Parameters**

| | | |
|---|---|---|
| in | *lBlockSize* | • Firmware Read Block Size. |

**Returns**

none

#### 9.9.5.16 int litefw_switch_9x07_to_downloadmode ( char * *szTTYPath* )

This API switch 9x07 modem to download mode.

**Parameters**

| | | |
|---|---|---|
| in | *szTTYPath* | • QDL Device Path. |

**Returns**

0 on success, -1 error value otherwise

#### 9.9.5.17 int litefw_switch_to_BootHoldMode ( char * *szTTYPath* )

This API switch modem to boot hold modem via QDL port.

**Parameters**

| | | |
|---|---|---|
| in | *szTTYPath* | • QDL Device Path. |

**Returns**

0 on success, -1 error value otherwise

## 9.10 loc.h File Reference

### Data Structures

- struct loc_LocApplicationInfo
- struct loc_SV
- struct loc_SVInfo
- struct loc_GnssData
- struct loc_CellDb
- struct loc_ClkInfo
- struct loc_BdsSV
- struct loc_BdsSVInfo
- struct pack_loc_EventRegister_t
- struct unpack_loc_EventRegister_t
- struct pack_loc_SetExtPowerState_t
- struct unpack_loc_SetExtPowerState_t
- struct pack_loc_Start_t
- struct unpack_loc_Start_t
- struct pack_loc_Stop_t
- struct unpack_loc_Stop_t
- struct pack_loc_SetOperationMode_t
- struct unpack_loc_SetOperationMode_t
- struct pack_loc_Delete_Assist_Data_t
- struct unpack_loc_Delete_Assist_Data_t
- struct loc_precisionDilution
- struct loc_sensorDataUsage
- struct loc_svUsedforFix
- struct loc_gpsTime
- struct unpack_loc_PositionRpt_Ind_t
- struct unpack_loc_EngineState_Ind_t
- struct unpack_loc_SetExtPowerConfig_Ind_t
- struct unpack_loc_SLQSLOCGetBestAvailPos_t
- struct pack_loc_SLQSLOCGetBestAvailPos_t
- struct unpack_loc_BestAvailPos_Ind_t
- struct unpack_loc_SetOperationMode_Ind_t
- struct unpack_loc_DeleteAssistData_Ind_t
- struct loc_satelliteInfo
- struct unpack_loc_GnssSvInfo_Ind_t
- struct pack_loc_SLQSLOCInjectUTCTime_t
- struct altSrcInfo_t
- struct pack_loc_SLQSLOCInjectPosition_t
- struct pack_loc_SLQSLOCSetCradleMountConfig_t
- struct sensorData_t
- struct tempData_t
- struct pack_loc_SLQSLOCInjectSensorData_t
- struct unpack_loc_EventNMEA_Ind_t
- struct pack_loc_SLQSLOCGetServer_t
- struct loc_IPv4Info
- struct loc_IPv6Info
- struct loc_urlAddr

- struct unpack_loc_GetServer_Ind_t
- struct unpack_loc_CradleMountCallback_Ind_t
- struct unpack_loc_EventTimeSyncCallback_Ind_t
- struct unpack_loc_InjectTimeSyncDataCallback_Ind_t
- struct unpack_loc_InjectSensorDataCallback_Ind_t
- struct loc_accelAcceptReady
- struct loc_gyroAcceptReady
- struct loc_accelTempAcceptReady
- struct loc_gyroTempAcceptReady
- struct unpack_loc_SensorStreamingCallback_Ind_t
- struct unpack_loc_InjectUTCTimeCallback_Ind_t
- struct unpack_loc_InjectPositionCallback_Ind_t
- struct loc_IPv4Config
- struct loc_IPv6Config
- struct loc_URLAddrInfo
- struct pack_loc_SLQSLOCSetServer_t
- struct unpack_loc_SetServer_Ind_t

## Macros

- #define LOC_UINT8_MAX_STRING_SZ 255
- #define LOCEVENTMASKPOSITIONREPORT 0x00000001
- #define LOCEVENTMASKGNSSSVINFO 0x00000002
- #define LOCEVENTMASKNMEA 0x00000004
- #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008
- #define LOCEVENTMASKINJECTTIMEREQ 0x00000010
- #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020
- #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040
- #define LOCEVENTMASKENGINESTATE 0x00000080
- #define LOCEVENTMASKFIXSESSIONSTATE 0x00000100
- #define LOCEVENTMASKWIFIREQ 0x00000200
- #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400
- #define LOCEVENTMASKTIMESYNCREQ 0x00000800
- #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000
- #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000
- #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000
- #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000
- #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000
- #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000
- #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000
- #define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000
- #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000
- #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000
- #define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000
- #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000
- #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000
- #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF
- #define MAX_SENSOR_DATA_LEN 64
- #define MAX_TEMP_DATA_LEN 64
- #define MAX_LOC_NMEA_STR_LEN 201

## Typedefs

- typedef unpack_result_t unpack_loc_SLQSLOCInjectUTCTime_t
- typedef unpack_result_t unpack_loc_SLQSLOCInjectPosition_t
- typedef unpack_result_t unpack_loc_SLQSLOCSetCradleMountConfig_t
- typedef unpack_result_t unpack_loc_SLQSLOCInjectSensorData_t
- typedef unpack_result_t unpack_loc_SLQSLOCGetServer_t

## Enumerations

- enum {
  eQMI_LOC_SESS_STATUS_SUCCESS =0,
  eQMI_LOC_SESS_STATUS_IN_PROGRESS =1,
  eQMI_LOC_SESS_STATUS_FAILURE =2,
  eQMI_LOC_SESS_STATUS_TIMEOUT =3 }

## Functions

- int pack_loc_EventRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_EventRegister_t *reqArg)
- int unpack_loc_EventRegister (uint8_t *pResp, uint16_t respLen, unpack_loc_EventRegister_t *pOutput)
- int pack_loc_SetExtPowerState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SetExtPowerState_t *reqArg)
- int unpack_loc_SetExtPowerState (uint8_t *pResp, uint16_t respLen, unpack_loc_SetExtPowerState_t *pOutput)
- int pack_loc_Start (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Start_t *reqArg)
- int unpack_loc_Start (uint8_t *pResp, uint16_t respLen, unpack_loc_Start_t *pOutput)
- int pack_loc_Stop (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Stop_t *reqArg)
- int unpack_loc_Stop (uint8_t *pResp, uint16_t respLen, unpack_loc_Stop_t *pOutput)
- int pack_loc_SetOperationMode (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SetOperationMode_t *reqArg)
- int unpack_loc_SetOperationMode (uint8_t *pResp, uint16_t respLen, unpack_loc_SetOperationMode_t *pOutput)
- int pack_loc_DeleteAssistData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_Delete_Assist_Data_t *reqArg)
- int unpack_loc_DeleteAssistData (uint8_t *pResp, uint16_t respLen, unpack_loc_Delete_Assist_Data_t *pOutput)
- int unpack_loc_PositionRpt_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_PositionRpt_Ind_t *pOutput)
- int unpack_loc_EngineState_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_EngineState_Ind_t *pOutput)
- int unpack_loc_SetExtPowerConfig_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_SetExtPowerConfig_Ind_t *pOutput)
- int pack_loc_SLQSLOCGetBestAvailPos (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCGetBestAvailPos_t *reqArg)
- int unpack_loc_SLQSLOCGetBestAvailPos (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCGetBestAvailPos_t *pOutput)
- int unpack_loc_BestAvailPos_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_BestAvailPos_Ind_t *pOutput)
- int unpack_loc_SetOperationMode_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_SetOperationMode_Ind_t *pOutput)
- int unpack_loc_DeleteAssistData_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_DeleteAssistData_Ind_t *pOutput)
- int unpack_loc_GnssSvInfo_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_GnssSvInfo_Ind_t *pOutput)
- int pack_loc_SLQSLOCInjectUTCTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectUTCTime_t *reqArg)

- int unpack_loc_SLQSLOCInjectUTCTime (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCInjectUTCTime_t *pOutput)
- int pack_loc_SLQSLOCInjectPosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectPosition_t *reqArg)
- int unpack_loc_SLQSLOCInjectPosition (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCInjectPosition_t *pOutput)
- int pack_loc_SLQSLOCSetCradleMountConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCSetCradleMountConfig_t *reqArg)
- int unpack_loc_SLQSLOCSetCradleMountConfig (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCSetCradleMountConfig_t *pOutput)
- int pack_loc_SLQSLOCInjectSensorData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCInjectSensorData_t *reqArg)
- int unpack_loc_SLQSLOCInjectSensorData (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCInjectSensorData_t *pOutput)
- int unpack_loc_EventNMEA_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_EventNMEA_Ind_t *pOutput)
- int pack_loc_SLQSLOCGetServer (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCGetServer_t *reqArg)
- int unpack_loc_SLQSLOCGetServer (uint8_t *pResp, uint16_t respLen, unpack_loc_SLQSLOCGetServer_t *pOutput)
- int unpack_loc_GetServer_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_GetServer_Ind_t *pOutput)
- int unpack_loc_CradleMountCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_CradleMountCallback_Ind_t *pOutput)
- int unpack_loc_EventTimeSyncCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_EventTimeSyncCallback_Ind_t *pOutput)
- int unpack_loc_InjectTimeSyncDataCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_InjectTimeSyncDataCallback_Ind_t *pOutput)
- int unpack_loc_InjectSensorDataCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_InjectSensorDataCallback_Ind_t *pOutput)
- int unpack_loc_SensorStreamingCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_SensorStreamingCallback_Ind_t *pOutput)
- int unpack_loc_InjectUTCTimeCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_InjectUTCTimeCallback_Ind_t *pOutput)
- int unpack_loc_InjectPositionCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_InjectPositionCallback_Ind_t *pOutput)
- int pack_loc_SLQSLOCSetServer (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_loc_SLQSLOCSetServer_t *reqArg)
- int unpack_loc_SLQSLOCSetServer (uint8_t *pResp, uint16_t respLen)
- int unpack_loc_SetServer_Ind (uint8_t *pResp, uint16_t respLen, unpack_loc_SetServer_Ind_t *pOutput)

### 9.10.1 Macro Definition Documentation

#### 9.10.1.1 #define LOC_UINT8_MAX_STRING_SZ 255

#### 9.10.1.2 #define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000

The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.

#### 9.10.1.3 #define LOCEVENTMASKENGINESTATE 0x00000080

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

#### 9.10.1.4 #define LOCEVENTMASKFIXSESSIONSTATE 0x00000100

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

#### 9.10.1.5 #define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

#### 9.10.1.6 #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.

#### 9.10.1.7 #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000

The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, for example, if GPS is turned off or if the network is unavailable.

#### 9.10.1.8 #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000

The control point must enable this mask to receive system clock and satellite measurement report events (system clock, SV time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_ CONSTELL_REPORT_CONFIG(Not yet supported).

#### 9.10.1.9 #define LOCEVENTMASKGNSSSVINFO 0x00000002

The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate.

#### 9.10.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040

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

#### 9.10.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020

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

#### 9.10.1.12 #define LOCEVENTMASKINJECTTIMEREQ 0x00000010

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

#### 9.10.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000

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

#### 9.10.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF

Invalid Event Mask

#### 9.10.1.15 #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000

The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.

#### 9.10.1.16 #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000

The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.

#### 9.10.1.17 #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000

The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.

#### 9.10.1.18 #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000

The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.

#### 9.10.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008

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

#### 9.10.1.20 #define LOCEVENTMASKNMEA 0x00000004

The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

#### 9.10.1.21 #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000

The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

#### 9.10.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001

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

#### 9.10.1.23 #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.).

#### 9.10.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000

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

#### 9.10.1.25 #define LOCEVENTMASKTIMESYNCREQ 0x00000800

The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

#### 9.10.1.26 #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).

#### 9.10.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200

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

#### 9.10.1.28 #define MAX_LOC_NMEA_STR_LEN 201

#### 9.10.1.29 #define MAX_SENSOR_DATA_LEN 64

#### 9.10.1.30 #define MAX_TEMP_DATA_LEN 64

### 9.10.2 Typedef Documentation

#### 9.10.2.1 typedef unpack_result_t unpack_loc_SLQSLOCGetServer_t

#### 9.10.2.2 typedef unpack_result_t unpack_loc_SLQSLOCInjectPosition_t

#### 9.10.2.3 typedef unpack_result_t unpack_loc_SLQSLOCInjectSensorData_t

#### 9.10.2.4 typedef unpack_result_t unpack_loc_SLQSLOCInjectUTCTime_t

#### 9.10.2.5 typedef unpack_result_t unpack_loc_SLQSLOCSetCradleMountConfig_t

### 9.10.3 Enumeration Type Documentation

#### 9.10.3.1 anonymous enum

**Enumerator**

***eQMI_LOC_SESS_STATUS_SUCCESS***

***eQMI_LOC_SESS_STATUS_IN_PROGRESS***

***eQMI_LOC_SESS_STATUS_FAILURE***

***eQMI_LOC_SESS_STATUS_TIMEOUT***

### 9.10.4 Function Documentation

#### 9.10.4.1 int pack_loc_DeleteAssistData ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Delete_Assist_Data_t * *reqArg* )

Delete Assistant Data pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.2 int pack_loc_EventRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_EventRegister_t * *reqArg* )

Event Register pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.3 int pack_loc_SetExtPowerState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SetExtPowerState_t * *reqArg* )

Set Ext Power State pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.4 int pack_loc_SetOperationMode ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetOperationMode_t * reqArg )

Set Operation Mode pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.5 int pack_loc_SLQSLOCGetBestAvailPos ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCGetBestAvailPos_t * reqArg )

Get Best Avail position pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.6 int pack_loc_SLQSLOCGetServer ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCGetServer_t * reqArg )

Pack get server.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through unpack_loc_GetServer_Ind callback If successful, the callback also contains the A-GPS server address.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.7 int pack_loc_SLQSLOCInjectPosition ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectPosition_t * *reqArg* )

Pack inject position.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.8 int pack_loc_SLQSLOCInjectSensorData ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectSensorData_t * *reqArg* )

Pack inject sensor data.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.9 int pack_loc_SLQSLOCInjectUTCTime ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCInjectUTCTime_t * *reqArg* )

Pack inject UTC time.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.10 int pack_loc_SLQSLOCSetCradleMountConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCSetCradleMountConfig_t * *reqArg* )

Pack set cradle mount configure.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.11 int pack_loc_SLQSLOCSetServer ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_SLQSLOCSetServer_t * *reqArg* )

Pack Set server.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

If multiple types of addresses are specified in the request, the IPv4 address takes precedence over the IPv6 address and the IPv6 address takes precedence over the URL address.The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through unpack_loc_SetServer_Ind callback

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.12 int pack_loc_Start ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Start_t * *reqArg* )

LOC Start pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.13 int pack_loc_Stop ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_loc_Stop_t * *reqArg* )

Loc Stop pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.14 int unpack_loc_BestAvailPos_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_BestAvailPos_Ind_t** * *pOutput* )

Loc Best Avial position Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.15 int unpack_loc_CradleMountCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_CradleMountCallback_Ind_t** * *pOutput* )

Unpack the Cradle Mount Indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.16 int unpack_loc_DeleteAssistData ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_Delete_Assist_Data_t** * *pOutput* )

Delete Assistant Data unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.17 int unpack_loc_DeleteAssistData_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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.18 int unpack_loc_EngineState_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EngineState_Ind_t * *pOutput* )

Loc Engine State Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.19 int unpack_loc_EventNMEA_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_EventNMEA_Ind_t * *pOutput* )

Unpack the NMEA Event Indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.20 int unpack_loc_EventRegister ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_EventRegister_t** * *pOutput* )

Event Register unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.21 int unpack_loc_EventTimeSyncCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_EventTimeSyncCallback_Ind_t** * *pOutput* )

Unpack the Event Time Sync indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.22 int unpack_loc_GetServer_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_GetServer_Ind_t** * *pOutput* )

Unpack the Get Server Indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.23 int unpack_loc_GnssSvInfo_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_GnssSvInfo_Ind_t** * *pOutput* )

Unpack the GNSS SV Info Indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.24 int unpack_loc_InjectPositionCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_InjectPositionCallback_Ind_t** * *pOutput* )

Unpack the Inject position indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.25 int unpack_loc_InjectSensorDataCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_InjectSensorDataCallback_Ind_t** * *pOutput* )

Unpack the inject sensor data indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.26 int unpack_loc_InjectTimeSyncDataCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_InjectTimeSyncDataCallback_Ind_t** * *pOutput* )

Unpack the inject time sync data indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.27 int unpack_loc_InjectUTCTimeCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_InjectUTCTimeCallback_Ind_t** * *pOutput* )

Unpack the Inject UTC time indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.28 int unpack_loc_PositionRpt_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_PositionRpt_Ind_t** * *pOutput* )

Loc Position Report Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.29 int unpack_loc_SensorStreamingCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SensorStreamingCallback_Ind_t * *pOutput* )

Unpack the Sensor streaming status indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.30 int unpack_loc_SetExtPowerConfig_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerConfig_Ind_t * *pOutput* )

Loc Set External Power Configure Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.31 int unpack_loc_SetExtPowerState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SetExtPowerState_t * *pOutput* )

Set Ext Power State unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.32 int unpack_loc_SetOperationMode ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_SetOperationMode_t** * *pOutput* )

Set Operation Mode unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.33 int unpack_loc_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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.34 int unpack_loc_SetServer_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_loc_SetServer_Ind_t** * *pOutput* )

Unpack the Set Server Indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.35 int unpack_loc_SLQSLOCGetBestAvailPos ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCGetBestAvailPos_t * *pOutput* )

Get Best Avail position unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.36 int unpack_loc_SLQSLOCGetServer ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCGetServer_t * *pOutput* )

Unpack get server.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.37 int unpack_loc_SLQSLOCInjectPosition ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCInjectPosition_t * *pOutput* )

Unpack inject position.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.38 int unpack_loc_SLQSLOCInjectSensorData ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCInjectSensorData_t * *pOutput* )

Unpack inject sensor data.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.39 int unpack_loc_SLQSLOCInjectUTCTime ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCInjectUTCTime_t * *pOutput* )

Unpack inject UTC time.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.40 int unpack_loc_SLQSLOCSetCradleMountConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_SLQSLOCSetCradleMountConfig_t * *pOutput* )

Unpack set cradle mount configure.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.41 int unpack_loc_SLQSLOCSetServer ( uint8_t * *pResp*, uint16_t *respLen* )

Unpack set server.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.42 int unpack_loc_Start ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Start_t * *pOutput* )

Loc Start unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.10.4.43 int unpack_loc_Stop ( uint8_t * *pResp*, uint16_t *respLen*, unpack_loc_Stop_t * *pOutput* )

Loc Stop unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.11 nas.h File Reference

### Data Structures

- struct unpack_nas_GetSignalStrengths_t
- struct unpack_nas_SLQSGetSysSelectionPref_t
- struct nas_netSelectionPref
- struct nas_acqOrderPref
- struct nas_CSGID
- struct pack_nas_SLQSSetSysSelectionPref_t
- struct nas_lteBandPrefExt
- struct nas_ciotAcqOrderPref
- struct nas_nr5gBandPref
- struct pack_nas_SLQSSetSysSelectionPrefExt_t
- struct pack_nas_SLQSSetBandPreference_t
- struct pack_nas_SLQSNasIndicationRegisterExt_t
- struct RFBandInfoElements
- struct unpack_nas_GetRFInfo_t
- struct cdmaSSInfo
- struct hdrSSInfo
- struct lteSSInfo
- struct tdscdmaSigInfoExt
- struct unpack_nas_SLQSNasGetSigInfo_t
- struct unpack_nas_SLQSNasSigInfoCallback_ind_t
- struct unpack_nas_GetHomeNetwork_t
- struct nas_SrvStatusInfo
- struct nas_GSMSrvStatusInfo
- struct nas_sysInfoCommon
- struct nas_CDMASysInfo
- struct nas_HDRSysInfo
- struct nas_GSMSysInfo
- struct nas_WCDMASysInfo
- struct nas_LTESysInfo
- struct nas_AddCDMASysInfo
- struct nas_AddSysInfo
- struct nas_CallBarringSysInfo
- struct 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 unpack_nas_GetANAAAAuthenticationStatus_t
- struct unpack_nas_GetACCOLC_t
- struct pack_nas_SetACCOLC_t
- struct nas_CDMARSSIThresh
- struct nas_CDMAECIOThresh
- struct nas_HDRRSSIThresh
- struct nas_HDRECIOThresh
- struct nas_HDRSINRThreshold
- struct nas_HDRIOThresh
- struct nas_GSMRSSIThresh
- struct nas_WCDMARSSIThresh
- struct nas_WCDMAECIOThresh
- struct nas_LTERSSIThresh
- struct nas_LTESNRThreshold
- struct nas_LTERSRQThresh
- struct nas_LTERSRPThresh
- struct nas_LTESigRptConfig
- struct nas_TDSCDMARSCPThresh
- 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 NASServingSystemInfo

- struct unpack_nas_SetServingSystemCallback_ind_t
- struct NASPhyCaAggScellIndType
- struct NASPhyCaAggScellDlBw
- 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_PhyCaAggScellDlBw
- struct nas_PhyCaAggScellInfo

- struct nas_PhyCaAggPcellInfo
- struct nas_PhyCaAggScellIndex
- struct unpack_nas_SetNasLTECphyCaIndCallback_ind_t
- struct nas_RxSigInfo
- struct nas_SccRxInfo
- struct unpack_nas_SLQSSwiGetLteSccRxInfo_t
- struct unpack_nas_SLQSNasTimerCallback_ind_t
- struct pack_nas_InitiateDomainAttach_t
- struct pack_nas_SetCDMANetworkParameters_t
- struct unpack_nas_SLQSNasGetHDRColorCode_t
- struct pack_nas_SLQSNasGetTxRxInfo_t
- struct nas_rxInfo
- struct nas_txInfo
- struct unpack_nas_SLQSNasGetTxRxInfo_t
- struct nas_OperatorPLMNData
- struct nas_operatorPLMNList
- struct nas_serviceProviderName
- struct nas_PLMNNetworkNameData
- struct nas_PLMNNetworkName
- struct nas_operatorNameString
- struct unpack_nas_SLQSGetOperatorNameData_t
- struct pack_nas_SLQSNasGet3GPP2Subscription_t
- struct nas_namName
- struct nas_dirNum
- struct nas_sidNid
- struct nas_homeSIDNID
- struct nas_minBasedIMSI
- struct nas_trueIMSI
- struct nas_CDMAChannel
- struct unpack_nas_SLQSNasGet3GPP2Subscription_t
- struct nas_protocolSubtypeElement
- struct unpack_nas_SLQSSwiGetHDRPersonality_t
- struct unpack_nas_SLQSSwiGetHDRProtSubtype_t
- struct pack_nas_SLQSSwiPSDetach_t
- struct unpack_nas_SLQSGetErrorRate_t
- struct nas_DRCParams
- struct nas_PilotSetParams
- struct nas_PilotSetData
- struct unpack_nas_SLQSSwiGetHRPDStats_t
- struct nas_ActPilotPNElement
- struct nas_NetworkStat1x
- struct nas_NetworkStatEVDO
- struct nas_DeviceConfigDetail
- struct nas_DataStatusDetail
- struct unpack_nas_SLQSSwiNetworkDebug_t
- struct nas_RSSIThresh
- struct nas_ECIOThresh
- struct nas_HDRSINRThresh
- struct nas_LTESNRThresh
- struct nas_IOThresh
- struct nas_RSRQThresh
- struct nas_RSRPThresh
- struct nas_LTESigRptCfg
- struct nas_TDSCDMASINRCONFThresh
- struct pack_nas_SLQSConfigSigInfo_t

- struct unpack_nas_GetHomeNetwork3GPP2_t
- struct nas_wcdmaUARFCN
- struct nas_lteEARFCN
- struct nas_ltePCI
- struct pack_nas_SLQSNASSwiSetChannelLock_t
- struct unpack_nas_SLQSNASSwiGetChannelLock_t
- struct unpack_nas_SLQSNASGeteDRXParams_t
- struct pack_nas_SLQSNASSeteDRXParams_t
- struct nas_HDRPersonality_Ind_Data
- struct unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t
- struct nas_RankIndicatorTlv
- struct unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t
- struct nas_RFBandInfoElements
- struct nas_RfDedicatedBandInfoElements
- struct nas_RfBandInfoExtFormatElements
- struct nas_RfBandwidthInfoElements
- struct nas_RfBandInfoList
- struct nas_RfDedicatedBandInfo
- struct nas_RfBandInfoExtFormat
- struct nas_RfBandwidthInfo
- struct nas_LTEOperationMode
- struct unpack_nas_SLQSNasGetRFInfo_t
- struct nas_EmerModeTlv
- struct nas_ModePrefTlv
- struct nas_BandPrefTlv
- struct nas_PRLPrefTlv
- struct nas_RoamPrefTlv
- struct nas_LTEBandPrefTlv
- struct nas_NetSelPrefTlv
- struct nas_SrvDomainPrefTlv
- struct nas_GWAcqOrderPrefTlv
- struct nas_AcqOrderPrefTlv
- struct nas_RatDisabledMaskTlv
- struct nas_CiotLteOpModePrefTlv
- struct nas_LteM1BandPrefTlv
- struct nas_LteNb1BandPrefTlv
- struct nas_CiotAcqOrderPrefTlv
- struct nas_NR5gBandPrefTlv
- struct unpack_nas_SLQSGetSysSelectionPrefExt_t

## Macros

- #define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048
- #define NAS_MAX_NUM_NETWORKS 30
- #define NAS_MAX_DESCRIPTION_LENGTH 255
- #define NAS_PLMN_LENGTH 3
- #define NAS_MAX_SCC_RX_INFO_INSTANCES 255
- #define NAS_NAM_NAME_LENGTH 12
- #define NAS_IMSI_M_S1_LENGTH 7
- #define NAS_IMSI_M_S2_LENGTH 3
- #define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255

## Typedefs

- typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPref_t
- typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPrefExt_t
- typedef unpack_result_t unpack_nas_SLQSSetBandPreference_t
- typedef unpack_result_t unpack_nas_SLQSNasIndicationRegisterExt_t
- typedef unpack_result_t unpack_nas_SLQSSetSignalStrengthsCallback_t
- typedef unpack_result_t unpack_nas_SetRFInfoCallback_t
- typedef unpack_result_t unpack_nas_SetLURejectCallback_t
- typedef unpack_result_t unpack_nas_SetACCOLC_t
- typedef unpack_result_t unpack_nas_SLQSNasConfigSigInfo2_t
- typedef unpack_result_t unpack_nas_SLQSInitiateNetworkRegistration_t
- typedef unpack_result_t unpack_nas_SLQSNasSwiIndicationRegister_t
- typedef unpack_result_t unpack_nas_InitiateDomainAttach_t
- typedef unpack_result_t unpack_nas_SetCDMANetworkParameters_t
- typedef unpack_result_t unpack_nas_SLQSSwiPSDetach_t
- typedef unpack_result_t unpack_nas_SLQSConfigSigInfo_t
- typedef unpack_result_t unpack_nas_SLQSNASSwiSetChannelLock_t
- typedef unpack_result_t unpack_nas_SLQSNASSeteDRXParams_t

## Enumerations

- enum LITEQMI_NAS_LTE_CPHY_SCELL_STATE {
  eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED =0x00,
  eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED =0x01,
  eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED =0x02 }
- enum LITEQMI_NAS_LTE_CPHY_CA_BW_NRB {
  eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6 =0x00,
  eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15 =0x01,
  eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25 =0x02,
  eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50 =0x03,
  eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75 =0x04,
  eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100 =0x05 }
- enum NAS_LTE_CPHY_CA_BW_NRB_LITE {
  eNAS_LTE_CPHY_CA_BW_NRB_LITE_6 =0x00,
  eNAS_LTE_CPHY_CA_BW_NRB_LITE_15 =0x01,
  eNAS_LTE_CPHY_CA_BW_NRB_LITE_25 =0x02,
  eNAS_LTE_CPHY_CA_BW_NRB_LITE_50 =0x03,
  eNAS_LTE_CPHY_CA_BW_NRB_LITE_75 =0x04,
  eNAS_LTE_CPHY_CA_BW_NRB_LITE_100 =0x05 }
- enum NAS_LTE_CPHY_SCELL_STATE_LITE {
  eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE =0x00,
  eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE =0x01,
  eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE =0x02 }

## Functions

- int unpack_nas_GetSignalStrengths (uint8_t *pResp, uint16_t respLen, unpack_nas_GetSignalStrengths_t *pOutput)
- int pack_nas_GetSignalStrengths (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen)
- int pack_nas_SLQSGetSysSelectionPref (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetSysSelectionPref (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSys-SelectionPref_t *pOutput)
- int pack_nas_SLQSSetSysSelectionPref (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_-SLQSSetSysSelectionPref_t *pReqParam)

- int unpack_nas_SLQSSetSysSelectionPref (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPref_t *pOutput)
- int pack_nas_SLQSSetSysSelectionPrefExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSysSelectionPrefExt_t *pReqParam)
- int unpack_nas_SLQSSetSysSelectionPrefExt (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPrefExt_t *pOutput)
- int pack_nas_SLQSSetBandPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetBandPreference_t *pReqParam)
- int unpack_nas_SLQSSetBandPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetBandPreference_t *pOutput)
- int pack_nas_SLQSNasIndicationRegisterExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasIndicationRegisterExt_t *pReqParam)
- int unpack_nas_SLQSNasIndicationRegisterExt (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasIndicationRegisterExt_t *pOutput)
- int pack_nas_GetRFInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetRFInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_GetRFInfo_t *pOutput)
- int pack_nas_SLQSNasGetSigInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetSigInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetSigInfo_t *pOutput)
- int unpack_valid_nas_SLQSNasGetSigInfo (uint8_t *pResp, uint8_t u8Info)
- int unpack_nas_SLQSNasSigInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSigInfoCallback_ind_t *pOutput)
- int unpack_nas_GetHomeNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t *pOutput)
- int pack_nas_GetHomeNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int pack_nas_SLQSGetSysInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetSysInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t *pOutput)
- int unpack_nas_SLQSNasSysInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSysInfoCallback_ind_t *pOutput)
- int pack_nas_GetServingNetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetwork (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t *pOutput)
- int pack_nas_GetServingNetworkCapabilities (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetServingNetworkCapabilities (uint8_t *pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t *pOutput)
- int pack_nas_PerformNetworkScan (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_PerformNetworkScan (uint8_t *pResp, uint16_t respLen, unpack_nas_PerformNetworkScan_t *pOutput)
- int pack_nas_SLQSSwiGetLteCQI (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetLteCQI (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteCQI_t *pOutput)
- int pack_nas_SLQSNasSwiModemStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasSwiModemStatus (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiModemStatus_t *pOutput)
- int pack_nas_SLQSGetServingSystem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetServingSystem (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetServingSystem_t *pOutput)
- int unpack_valid_nas_SLQSGetServingSystem (uint8_t *pResp, uint8_t u8Info)
- int pack_nas_SLQSGetSignalStrength (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint16_t reqMask)
- int unpack_nas_SLQSGetSignalStrength (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t *pOutput)
- int unpack_valid_nas_SLQSGetSignalStrength (uint8_t *pResp, uint8_t u8Info)
- int pack_nas_SLQSSetSignalStrengthsCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSetSignalStrengthsCallback_t *pReqParam)

- int unpack_nas_SLQSSetSignalStrengthsCallback (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSignalStrengthsCallback_t *pOutput)
- int pack_nas_SetRFInfoCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetRFInfoCallback (uint8_t *pResp, uint16_t respLen, unpack_nas_SetRFInfoCallback_t *pOutput)
- int pack_nas_SetLURejectCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pBenable)
- int unpack_nas_SetLURejectCallback (uint8_t *pResp, uint16_t respLen, unpack_nas_SetLURejectCallback_t *pOutput)
- int unpack_nas_SetEventReportInd (uint8_t *pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *pOutput)
- int pack_nas_GetCDMANetworkParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetCDMANetworkParameters (uint8_t *pResp, uint16_t respLen, unpack_nas_GetCDMANetworkParameters_t *pOutput)
- int unpack_valid_nas_GetCDMANetworkParameters (uint8_t *pResp, uint8_t u8Info)
- int pack_nas_GetANAAAAuthenticationStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetANAAAAuthenticationStatus (uint8_t *pResp, uint16_t respLen, unpack_nas_GetANAAAAuthenticationStatus_t *pOutput)
- int pack_nas_GetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetACCOLC (uint8_t *pResp, uint16_t respLen, unpack_nas_GetACCOLC_t *pOutput)
- int pack_nas_SetACCOLC (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetACCOLC_t reqParam)
- int unpack_nas_SetACCOLC (uint8_t *pResp, uint16_t respLen, unpack_nas_SetACCOLC_t *pOutput)
- int pack_nas_SLQSNasConfigSigInfo2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasConfigSigInfo2_t *pReqParam)
- int unpack_nas_SLQSNasConfigSigInfo2 (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasConfigSigInfo2_t *pOutput)
- int unpack_nas_SetDataCapabilitiesCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetDataCapabilitiesCallback_ind_t *pOutput)
- int pack_nas_GetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t *pOutput)
- int pack_nas_SetNetworkPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetNetworkPreference_t *reqArg)
- int unpack_nas_SetNetworkPreference (uint8_t *pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t *pOutput)
- int unpack_nas_SetRoamingIndicatorCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t *pOutput)
- int unpack_nas_SetServingSystemCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t *pOutput)
- int pack_nas_SlqsGetLTECphyCAInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SlqsGetLTECphyCAInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t *pOutput)
- int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t *pOutput)
- int unpack_nas_SLQSNasSwiOTAMessageCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t *pOutput)
- int pack_nas_SLQSInitiateNetworkRegistration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSInitiateNetworkRegistration_t *pReqParam)
- int unpack_nas_SLQSInitiateNetworkRegistration (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSInitiateNetworkRegistration_t *pOutput)
- int pack_nas_SLQSNasSwiIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasSwiIndicationRegister_t *pReqParam)
- int unpack_nas_SLQSNasSwiIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasSwiIndicationRegister_t *pOutput)

- int pack_nas_SLQSGetPLMNName (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSGetPLMNName_t *reqArg)
- int unpack_nas_SLQSGetPLMNName (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t *pOutput)
- int pack_nas_SLQSNasGetCellLocationInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetCellLocationInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetCellLocationInfo_t *pOutput)
- int pack_nas_SLQSGetNetworkTime (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetNetworkTime (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t *pOutput)
- int unpack_nas_SLQSNasNetworkTimeCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t *pOutput)
- int unpack_nas_SetNasLTECphyCaIndCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCaIndCallback_ind_t *pOutput)
- int pack_nas_SLQSSwiGetLteSccRxInfo (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetLteSccRxInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetLteSccRxInfo_t *pOutput)
- int unpack_nas_SLQSNasTimerCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasTimerCallback_ind_t *pOutput)
- int pack_nas_InitiateDomainAttach (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_InitiateDomainAttach_t *pReqParam)
- int unpack_nas_InitiateDomainAttach (uint8_t *pResp, uint16_t respLen, unpack_nas_InitiateDomainAttach_t *pOutput)
- int pack_nas_SetCDMANetworkParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SetCDMANetworkParameters_t *pReqParam)
- int unpack_nas_SetCDMANetworkParameters (uint8_t *pResp, uint16_t respLen, unpack_nas_SetCDMANetworkParameters_t *pOutput)
- int pack_nas_SLQSNasGetHDRColorCode (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetHDRColorCode (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetHDRColorCode_t *pOutput)
- int pack_nas_SLQSNasGetTxRxInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasGetTxRxInfo_t *reqArg)
- int unpack_nas_SLQSNasGetTxRxInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetTxRxInfo_t *pOutput)
- int pack_nas_SLQSGetOperatorNameData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetOperatorNameData (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetOperatorNameData_t *pOutput)
- int pack_nas_SLQSNasGet3GPP2Subscription (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNasGet3GPP2Subscription_t *reqArg)
- int unpack_nas_SLQSNasGet3GPP2Subscription (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGet3GPP2Subscription_t *pOutput)
- int pack_nas_SLQSSwiGetHDRPersonality (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetHDRPersonality (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHDRPersonality_t *pOutput)
- int pack_nas_SLQSSwiGetHDRProtSubtype (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiGetHDRProtSubtype (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHDRProtSubtype_t *pOutput)
- int pack_nas_SLQSSwiPSDetach (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSSwiPSDetach_t *pReqParam)
- int unpack_nas_SLQSSwiPSDetach (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiPSDetach_t *pOutput)
- int pack_nas_SLQSGetErrorRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetErrorRate (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetErrorRate_t *pOutput)
- int pack_nas_SLQSSwiGetHRPDStats (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int unpack_nas_SLQSSwiGetHRPDStats (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHRPDStats_t *pOutput)
- int pack_nas_SLQSSwiNetworkDebug (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSSwiNetworkDebug (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiNetworkDebug_t *pOutput)
- int pack_nas_SLQSConfigSigInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSConfigSigInfo_t *pReqParam)
- int unpack_nas_SLQSConfigSigInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSConfigSigInfo_t *pOutput)
- int pack_nas_GetHomeNetwork3GPP2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_GetHomeNetwork3GPP2 (uint8_t *pResp, uint16_t respLen, unpack_nas_GetHomeNetwork3GPP2_t *pOutput)
- int pack_nas_SLQSNASSwiSetChannelLock (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNASSwiSetChannelLock_t *pReqParam)
- int unpack_nas_SLQSNASSwiSetChannelLock (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNASSwiSetChannelLock_t *pOutput)
- int pack_nas_SLQSNASSwiGetChannelLock (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNASSwiGetChannelLock (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNASSwiGetChannelLock_t *pOutput)
- int pack_nas_SLQSNASGeteDRXParams (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNASGeteDRXParams (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNASGeteDRXParams_t *pOutput)
- int pack_nas_SLQSNASSeteDRXParams (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_nas_SLQSNASSeteDRXParams_t *pReqParam)
- int unpack_nas_SLQSNASSeteDRXParams (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNASSeteDRXParams_t *pOutput)
- int unpack_nas_SLQSSwiHDRPersonalityCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t *pOutput)
- int unpack_nas_SLQSSwiRandIndicatorCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t *pOutput)
- int pack_nas_SLQSNasGetRFInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSNasGetRFInfo (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSNasGetRFInfo_t *pOutput)
- int pack_nas_SLQSGetSysSelectionPrefExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_nas_SLQSGetSysSelectionPrefExt (uint8_t *pResp, uint16_t respLen, unpack_nas_SLQSGetSysSelectionPrefExt_t *pOutput)

### 9.11.1 Macro Definition Documentation

#### 9.11.1.1 #define NAS_IMSI_M_S1_LENGTH 7

#### 9.11.1.2 #define NAS_IMSI_M_S2_LENGTH 3

#### 9.11.1.3 #define NAS_MAX_DESCRIPTION_LENGTH 255

#### 9.11.1.4 #define NAS_MAX_NUM_NETWORKS 30

#### 9.11.1.5 #define NAS_MAX_SCC_RX_INFO_INSTANCES 255

#### 9.11.1.6 #define NAS_NAM_NAME_LENGTH 12

#### 9.11.1.7 #define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048

#### 9.11.1.8 #define NAS_PLMN_LENGTH 3

#### 9.11.1.9 #define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255

### 9.11.2 Typedef Documentation

#### 9.11.2.1 typedef unpack_result_t unpack_nas_InitiateDomainAttach_t

#### 9.11.2.2 typedef unpack_result_t unpack_nas_SetACCOLC_t

#### 9.11.2.3 typedef unpack_result_t unpack_nas_SetCDMANetworkParameters_t

#### 9.11.2.4 typedef unpack_result_t unpack_nas_SetLURejectCallback_t

#### 9.11.2.5 typedef unpack_result_t unpack_nas_SetRFInfoCallback_t

#### 9.11.2.6 typedef unpack_result_t unpack_nas_SLQSConfigSigInfo_t

#### 9.11.2.7 typedef unpack_result_t unpack_nas_SLQSInitiateNetworkRegistration_t

#### 9.11.2.8 typedef unpack_result_t unpack_nas_SLQSNasConfigSigInfo2_t

#### 9.11.2.9 typedef unpack_result_t unpack_nas_SLQSNasIndicationRegisterExt_t

#### 9.11.2.10 typedef unpack_result_t unpack_nas_SLQSNASSeteDRXParams_t

#### 9.11.2.11 typedef unpack_result_t unpack_nas_SLQSNasSwiIndicationRegister_t

#### 9.11.2.12 typedef unpack_result_t unpack_nas_SLQSNASSwiSetChannelLock_t

#### 9.11.2.13 typedef unpack_result_t unpack_nas_SLQSSetBandPreference_t

#### 9.11.2.14 typedef unpack_result_t unpack_nas_SLQSSetSignalStrengthsCallback_t

#### 9.11.2.15 typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPref_t

#### 9.11.2.16 typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPrefExt_t

#### 9.11.2.17 typedef unpack_result_t unpack_nas_SLQSSwiPSDetach_t

### 9.11.3 Enumeration Type Documentation

#### 9.11.3.1 enum LITEQMI_NAS_LTE_CPHY_CA_BW_NRB

**Enumerator**

*eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6*
*eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15*
*eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25*
*eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50*
*eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75*
*eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100*

#### 9.11.3.2 enum LITEQMI_NAS_LTE_CPHY_SCELL_STATE

**Enumerator**

*eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED*

***eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED***

***eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED***

#### 9.11.3.3 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***

#### 9.11.3.4 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***

### 9.11.4 Function Documentation

#### 9.11.4.1 int pack_nas_GetACCOLC ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Retrieves information about the access overload class (ACCOLC) pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.2 int pack_nas_GetANAAAAuthenticationStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

AN-AAA authentication status of the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.3 int pack_nas_GetCDMANetworkParameters ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get CDMA Network Parameters pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.4 int pack_nas_GetHomeNetwork ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get home network pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.5 int pack_nas_GetHomeNetwork3GPP2 ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get Home Network 3GPP2 Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.6 int pack_nas_GetNetworkPreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets the network registration preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

#### 9.11.4.7 int pack_nas_GetRFInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get rf info pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.8 int pack_nas_GetServingNetwork ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack provides information about the system that provides service to the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.9 int pack_nas_GetServingNetworkCapabilities ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack returns information regarding the data capabilities of the system that currently provides service to the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.10 int pack_nas_GetSignalStrengths ( pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen* )

get signal strengths pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15-C_05_xx_xx_xx and all EM74xx firmware versions. Please use pack_nas_SLQSNasGetSigInfo() for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReq* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.11 int pack_nas_InitiateDomainAttach ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_InitiateDomainAttach_t * *pReqParam* )

Initiates a domain attach/detach of the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.12 int pack_nas_PerformNetworkScan ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack performs scan for available networks and scans for RAT info as well.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.13 int pack_nas_SetACCOLC ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetACCOLC_t *reqParam* )

Pack Sets the access overload class (ACCOLC)

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.14 int pack_nas_SetCDMANetworkParameters ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetCDMANetworkParameters_t * *pReqParam* )

CDMA Network Parameters pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.15 int pack_nas_SetLURejectCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable* )

Pack set registration reject Callback

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pBenable* | 0/1 value to disable/enable indication respectively |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.16 int pack_nas_SetNetworkPreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SetNetworkPreference_t * *reqArg* )

Sets the network registration preference. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use pack_nas_SLQSSetSysSelection-Pref() for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

#### 9.11.4.17 int pack_nas_SetRFInfoCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint8_t * *pBenable* )

Pack Set RF Band Information callback.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pBenable* | 0/1 value to disable/enable indication respectively |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.18 int pack_nas_SLQSConfigSigInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSConfigSigInfo_t * *pReqParam* )

Config Sig Info Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.19 int pack_nas_SLQSGetErrorRate ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get Error Rate Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.20 int pack_nas_SlqsGetLTECphyCAInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack get carrier aggregation event information.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

#### 9.11.4.21 int pack_nas_SLQSGetNetworkTime ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Pack Get Network Time.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.22 int pack_nas_SLQSGetOperatorNameData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Operator Name Data Parameters pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.23 int pack_nas_SLQSGetPLMNName ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSGetPLMNName_t * reqArg )

Pack get operator name for specificed network.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.24 int pack_nas_SLQSGetServingSystem ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack queries information regarding the system that currently provides service.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.25 int pack_nas_SLQSGetSignalStrength ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, uint16_t *reqMask* )

Queries the current signal strength as measured by the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use pack_nas_SLQS-NasGetSigInfo() for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqMask* | request mask for fetching extra signal info |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.26 int pack_nas_SLQSGetSysInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack get system information.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.27 int pack_nas_SLQSGetSysSelectionPref ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Queries the different system selection preferences of the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.28 int pack_nas_SLQSGetSysSelectionPrefExt ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Queries the different system selection preferences of the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.29 int pack_nas_SLQSInitiateNetworkRegistration ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSInitiateNetworkRegistration_t * *pReqParam* )

Pack initiates a network registration.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.30 int pack_nas_SLQSNasConfigSigInfo2 ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasConfigSigInfo2_t * *pReqParam* )

Pack sets the signal strength reporting thresholds

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.31 int pack_nas_SLQSNasGet3GPP2Subscription ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasGet3GPP2Subscription_t * *reqArg* )

3GPP2 Subscription Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.32 int pack_nas_SLQSNasGetCellLocationInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack retrieves cell location-related information.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.33 int pack_nas_SLQSNASGeteDRXParams ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get eDRX Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.34 int pack_nas_SLQSNasGetHDRColorCode ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

HDR Color Code Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.35 int pack_nas_SLQSNasGetRFInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets radio band/channel information pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.36 int pack_nas_SLQSNasGetSigInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get sig info pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.37 int pack_nas_SLQSNasGetTxRxInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasGetTxRxInfo_t * *reqArg* )

Get Tx Rx Info Parameters pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.38 int pack_nas_SLQSNasIndicationRegisterExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasIndicationRegisterExt_t * pReqParam )

Pack used to Registers/De-registers for different NAS (Network access service) indications.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

This function is used by a control point to register/deregister for different QMI_NAS indications. The control point's registration state variables, controlling registration for indications, are modified to reflect the settings indicated in the parameters that are present in the request message. At least one optional parameter must be present in the request.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.39 int pack_nas_SLQSNASSeteDRXParams ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNASSeteDRXParams_t * pReqParam )

Set eDRX Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.40 int pack_nas_SLQSNASSwiGetChannelLock ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Get Channel Lock Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.41 int pack_nas_SLQSNasSwiIndicationRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNasSwiIndicationRegister_t * *pReqParam* )

Pack sets the registration state for different QMI_NAS SWI indications.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.42 int pack_nas_SLQSNasSwiModemStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

This function requests the device to return the current status of modem.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.43 int pack_nas_SLQSNASSwiSetChannelLock ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSNASSwiSetChannelLock_t * *pReqParam* )

Set Channel Lock Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.44 int pack_nas_SLQSSetBandPreference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSSetBandPreference_t * *pReqParam* )

Pack sets the different system selection preferences of the device.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.45 int pack_nas_SLQSSetSignalStrengthsCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSSetSignalStrengthsCallback_t * *pReqParam* )

Pack set signal strength thresholds callback.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request prarmeters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.46 int pack_nas_SLQSSetSysSelectionPref ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSysSelectionPref_t * pReqParam )

Sets the different system selection preferences of the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values4780

#### 9.11.4.47 int pack_nas_SLQSSetSysSelectionPrefExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSysSelectionPrefExt_t * pReqParam )

Sets the different system selection preferences of the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values4780

#### 9.11.4.48 int pack_nas_SLQSSwiGetHDRPersonality ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

HDR Personality Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.49 int pack_nas_SLQSSwiGetHDRProtSubtype ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

HDR Protocol Sub type Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.50 int pack_nas_SLQSSwiGetHRPDStats ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

HRPD Stats Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.51 int pack_nas_SLQSSwiGetLteCQI ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack fetch CQI parameters for LTE data session.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.52 int pack_nas_SLQSSwiGetLteSccRxInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReq*, uint16_t * *pLen* )

get LTE Secondary carrier Rx signal level information pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReq* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.53 int pack_nas_SLQSSwiNetworkDebug ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Network Debug Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.54 int pack_nas_SLQSSwiPSDetach ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_nas_SLQSSwiPSDetach_t * *pReqParam* )

PS Connection Detach Parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request Parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.55 int unpack_nas_GetACCOLC ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_GetACCOLC_t** * *pOutput* )

Retrieves information about the access overload class (ACCOLC) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.56 int unpack_nas_GetANAAAAuthenticationStatus ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_GetANAAAAuthenticationStatus_t** * *pOutput* )

AN-AAA authentication status of the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.57 int unpack_nas_GetCDMANetworkParameters ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_GetCDMANetworkParameters_t** * *pOutput* )

Get CDMA Network Parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | qmi output parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.58 int unpack_nas_GetHomeNetwork ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetHomeNetwork_t * *pOutput* )

Retrieves information about the home network of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.59 int unpack_nas_GetHomeNetwork3GPP2 ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetHomeNetwork3GPP2_t * *pOutput* )

Get Home Network 3GPP2 Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.60 int unpack_nas_GetNetworkPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetNetwork-Preference_t * *pOutput* )

Gets the network registration preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi resp from modem |
| in | *respLen* | qmi resp length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.61 int unpack_nas_GetRFInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_GetRFInfo_t** * *pOutput* )

get rf info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.62 int unpack_nas_GetServingNetwork ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_GetServingNetwork_t** * *pOutput* )

Unpack provides information about the system that provides service to the device. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use API unpack_nas_SLQSGetSysInfo() for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.63 int unpack_nas_GetServingNetworkCapabilities ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_GetServingNetworkCapabilities_t** * *pOutput* )

Unpack returns information regarding the data capabilities of the system that currently provides service to the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.64 int unpack_nas_GetSignalStrengths ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_GetSignalStrengths_t * *pOutput* )

Returns the available signal strengths ( in dBm ) as measured by the device in an array. This will also provides the corresponding radio interfaces in an array. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C_05_xx_xx_xx and all EM74xx firmware versions. Please use unpack_nas_SLQSNasGetSigInfo() for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.65 int unpack_nas_InitiateDomainAttach ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_InitiateDomainAttach_t * *pOutput* )

Initiates a domain attach/detach of the device unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.66 int unpack_nas_PerformNetworkScan ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_PerformNetworkScan_t * *pOutput* )

Unpack performs scan for available networks and scans for RAT info as well.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.67 int unpack_nas_SetACCOLC ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetACCOLC_t * *pOutput* )

Unpack Sets the access overload class (ACCOLC)

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.68 int unpack_nas_SetCDMANetworkParameters ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetCDMANetworkParameters_t * *pOutput* )

CDMA Network Parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.69 int unpack_nas_SetDataCapabilitiesCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_SetDataCapabilitiesCallback_ind_t** * *pOutput* )

Data Capabilities indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi indication from modem |
| in | *respLen* | qmi indication length |
| out | *pOutput* | indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.70 int unpack_nas_SetEventReportInd ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_SetEventReportInd_t** * *pOutput* )

Unpack set event report indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.71 int unpack_nas_SetLURejectCallback ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_SetLURejectCallback_t** * *pOutput* )

Unpack set registration reject Callback

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.72 int unpack_nas_SetNasLTECphyCaIndCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNasLTECphyCaIndCallback_ind_t * *pOutput* )

Unpack set LTE PHY CA indication callback.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | sig info indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.73 int unpack_nas_SetNetworkPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetNetwork-Preference_t * *pOutput* )

Sets the network registration preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi resp from modem |
| in | *respLen* | qmi resp length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.74 int unpack_nas_SetRFInfoCallback ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetRFInfoCallback_t * *pOutput* )

Unack Set RF Band Information callback.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.75 int unpack_nas_SetRoamingIndicatorCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetRoamingIndicatorCallback_ind_t * *pOutput* )

Roaming indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi indication from modem |
| in | *respLen* | qmi indication length |
| out | *pOutput* | indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.76 int unpack_nas_SetServingSystemCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SetServingSystemCallback_ind_t * *pOutput* )

Set Serving system indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi indication from modem |
| in | *respLen* | qmi indication length |
| out | *pOutput* | indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.77 int unpack_nas_SLQSConfigSigInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSConfigSigInfo_t * *pOutput* )

Config Sig Info Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.78 int unpack_nas_SLQSGetErrorRate ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetErrorRate_t * *pOutput* )

Get Error Rate Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.79 int unpack_nas_SlqsGetLTECphyCAInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SlqsGetLTECphyCAInfo_t * *pOutput* )

Unack get carrier aggregation event information.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi resp from modem |
| in | *respLen* | qmi resp length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.80 int unpack_nas_SLQSGetNetworkTime ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetNetworkTime_t * *pOutput* )

Unpack get network time.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.81 int unpack_nas_SLQSGetOperatorNameData ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetOperatorNameData_t * *pOutput* )

Operator Name Data Parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.82 int unpack_nas_SLQSGetPLMNName ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetPLMNName_t * *pOutput* )

unpack get operator name for specificed network.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.83 int unpack_nas_SLQSGetServingSystem ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetServingSystem_t * *pOutput* )

Unack queries information regarding the system that currently provides service.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.84 int unpack_nas_SLQSGetSignalStrength ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSignalStrength_t * *pOutput* )

Unpack get signal strength.This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C-_05_xx_xx_xx and all EM74xx firmware versions. Please use unpack_nas_SLQSNasGetSigInfo() for new firmware versions and new modules

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.85 int unpack_nas_SLQSGetSysInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysInfo_t * *pOutput* )

Provides the system information. This API is preferred when trying to get the service status info and serving system info. The function unpack_nas_SLQSGetServingSystem_t() reports similar NAS information, but it is deprecated. Please refer to the header description of unpack_nas_SLQSGetServingSystem_t() for more information.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

This API queries current serving system information, including registration information and system property. The registration information for all RATs specified in the mode capability setting are included regardless of registration status. The RAT-specific system property are included only for RATs that are specified in the mode capability setting and which are not in either No Service or Power Save modes.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.86 int unpack_nas_SLQSGetSysSelectionPref ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysSelectionPref_t * *pOutput* )

Queries the different system selection preferences of the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.87 int unpack_nas_SLQSGetSysSelectionPrefExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSGetSysSelectionPrefExt_t * *pOutput* )

Queries the different system selection preferences of the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.88 int unpack_nas_SLQSInitiateNetworkRegistration ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSInitiateNetworkRegistration_t * *pOutput* )

Unpack initiates a network registration.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.89 int unpack_nas_SLQSNasConfigSigInfo2 ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasConfigSigInfo2_t * *pOutput* )

Unpack sets the signal strength reporting thresholds

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.90 int unpack_nas_SLQSNasGet3GPP2Subscription ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGet3GPP2Subscription_t * *pOutput* )

3GPP2 Subscription Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.91 int unpack_nas_SLQSNasGetCellLocationInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetCellLocationInfo_t * *pOutput* )

Unpack retrieves cell location-related information.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.92 int unpack_nas_SLQSNASGeteDRXParams ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNASGeteDRXParams_t * *pOutput* )

Get eDRX Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.93 int unpack_nas_SLQSNasGetHDRColorCode ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetHDRColorCode_t * *pOutput* )

HDR Color Code Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.94 int unpack_nas_SLQSNasGetRFInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetRFInfo_t * *pOutput* )

Gets radio band/channel information unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.95 int unpack_nas_SLQSNasGetSigInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetSigInfo_t * *pOutput* )

get sig info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

This command queries the signal strength information for currently active RATs. Information is reported only if the corresponding RATs have signal strength values to be reported. If no signal strength information is available for any RAT, the response message contains only the mandatory response message

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.96 int unpack_nas_SLQSNasGetTxRxInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasGetTxRxInfo_t * *pOutput* )

Get Tx Rx Info Parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.97 int unpack_nas_SLQSNasIndicationRegisterExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasIndicationRegisterExt_t * *pOutput* )

Unpack used to Registers/De-registers for different NAS (Network access service) indications.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.98 int unpack_nas_SLQSNasNetworkTimeCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasNetworkTimeCallBack_ind_t * *pOutput* )

Unpack Network time callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | sig info indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.99 int unpack_nas_SLQSNASSeteDRXParams ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNASSeteDRXParams_t * *pOutput* )

Set eDRX Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.100 int unpack_nas_SLQSNasSigInfoCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSigInfoCallback_ind_t * *pOutput* )

Unpack Signal Information callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | sig info indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.101 int unpack_nas_SLQSNASSwiGetChannelLock ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNASSwiGetChannelLock_t * *pOutput* )

Get Channel Lock Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.102 int unpack_nas_SLQSNasSwiIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSwiIndicationRegister_t * *pOutput* )

unpack sets the registration state for different QMI_NAS SWI indications.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.103 int unpack_nas_SLQSNasSwiModemStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSwiModemStatus_t * *pOutput* )

This function requests the device to return the current status of modem.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.104 int unpack_nas_SLQSNasSwiOTAMessageCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t * *pOutput* )

OTA message indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi indication from modem |
| in | *respLen* | qmi indication length |
| out | *pOutput* | indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.105 int unpack_nas_SLQSNASSwiSetChannelLock ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNASSwiSetChannelLock_t * *pOutput* )

Set Channel Lock Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.106 int unpack_nas_SLQSNasSysInfoCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSysInfoCallback_ind_t * *pOutput* )

Unpack system information callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.107 int unpack_nas_SLQSNasTimerCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSNasTimerCallback_ind_t * *pOutput* )

Unpack timer callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | network timer indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.108 int unpack_nas_SLQSSetBandPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSetBandPreference_t * *pOutput* )

Unpack sets the different system selection preferences of the device.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

#### 9.11.4.109 int unpack_nas_SLQSSetSignalStrengthsCallback ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSetSignalStrengthsCallback_t * *pOutput* )

Unpack set signal strength thresholds callback.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.110 int unpack_nas_SLQSSetSysSelectionPref ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSetSysSelectionPref_t * *pOutput* )

Sets the different system selection preferences of the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.111 int unpack_nas_SLQSSetSysSelectionPrefCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t * *pOutput* )

System Selection Preference indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi indication from modem |
| in | *respLen* | qmi indication length |
| out | *pOutput* | indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.112 int unpack_nas_SLQSSetSysSelectionPrefExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSetSysSelectionPrefExt_t * *pOutput* )

Sets the different system selection preferences of the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.113 int unpack_nas_SLQSSwiGetHDRPersonality ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetHDRPersonality_t * *pOutput* )

HDR Personality Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.114 int unpack_nas_SLQSSwiGetHDRProtSubtype ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetHDRProtSubtype_t * *pOutput* )

HDR Protocol Sub type Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.115 int unpack_nas_SLQSSwiGetHRPDStats ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetHRPDStats_t * *pOutput* )

HRPD Stats Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.116 int unpack_nas_SLQSSwiGetLteCQI ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetLteCQI_t * *pOutput* )

Unack fetch CQI parameters for LTE data session.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.117 int unpack_nas_SLQSSwiGetLteSccRxInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiGetLteSccRxInfo_t * *pOutput* )

get LTE Secondary carrier Rx signal level information unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.118 int unpack_nas_SLQSSwiHDRPersonalityCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t** * *pOutput* )

unpack nas SWI Hdr Personality indication

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.119 int unpack_nas_SLQSSwiNetworkDebug ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_SLQSSwiNetworkDebug_t** * *pOutput* )

Network Debug Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.120 int unpack_nas_SLQSSwiPSDetach ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_nas_SLQSSwiPSDetach_t** * *pOutput* )

PS Connection Detach Parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.121 int unpack_nas_SLQSSwiRandIndicatorCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t * *pOutput* )

unpack nas SWI Rank indicator indication

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.122 int unpack_valid_nas_GetCDMANetworkParameters ( uint8_t * *pResp*, uint8_t *u8Info* )

valid CDMA Network Parameters

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *u8Info* | value to check unpack_nas_GetCDMANetworkParameters_t param valid.<br>• 0 - SCI<br>• 1 - SCM<br>• 2 - RegHomeSID, RegForeignSID and RegForeignNID<br>• 3 - ForceRev0<br>• 4 - CustomSCP, Protocol ,Broadcast and Application<br>• 5 - Roaming |

**Returns**

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.123 int unpack_valid_nas_SLQSGetServingSystem ( uint8_t * *pResp*, uint8_t *u8Info* )

valid queries information regarding the system that currently provides service

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *u8Info* | value to check unpack_nas_SLQSGetServingSystem_t param valid.<br>• 0 - RoamIndicatorVal<br>• 1 - DataSrvCapabilities<br>• 2 - CurrentPLMN<br>• 3 - SystemID and NetworkID<br>• 4 - BaseStationID, BaseStationLatitude and BasestationLongitude<br>• 5 - RoamingIndicatorList<br>• 6 - DefaultRoamInd<br>• 7 - Gpp2TimeZone<br>• 8 - CDMA_P_Rev<br>• 9 - GppTimeZone<br>• 10 - GppNetworkDSTAdjustment<br>• 11 - Lac<br>• 12 - CellID<br>• 13 - ConcSvcInfo<br>• 14 - PRLInd<br>• 15 - DTMInd<br>• 16 - DetailedSvcInfo<br>• 17 - CDMASystemInfoExt<br>• 18 - HdrPersonality<br>• 19 - TrackAreaCode<br>• 20 - CallBarStatus |

**Returns**

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.124 int unpack_valid_nas_SLQSGetSignalStrength ( uint8_t * *pResp*, uint8_t *u8Info* )

valid queries information regarding the system that currently provides service

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *u8Info* | value to check unpack_nas_SLQSGetSignalStrength_t param valid.<br>• 0 - rxSignalStrengthList and rxSignalStrengthListLen<br>• 1 - ecioList and ecioListLen<br>• 2 - Io<br>• 3 - sinr<br>• 4 - errorRateListLen, errorRateList<br>• 5 - rsrqInfo<br>• 6 - ltesnr<br>• 7 - ltersrp |

**Returns**

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.11.4.125 int unpack_valid_nas_SLQSNasGetSigInfo ( uint8_t * *pResp*, uint8_t *u8Info* )

valid get sig info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *u8Info* | value to check unpack_nas_SLQSNasGetSigInfo_t param valid.<br>• 0 - CDMASSInfo<br>• 1 - HDRSSInfo.<br>• 2 - GSMSSInfo.<br>• 3 - WCDMASSInfo.<br>• 4 - LTESSInfo. |

**Returns**

eQCWWAN_ERR_NONE on signal info valid, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.12 pds.h File Reference

### Data Structures

- struct unpack_pds_GetPDSState_t
- struct pack_pds_SetPDSState_t
- struct unpack_pds_GetPDSDefaults_t
- struct pack_pds_SetPDSDefaults_t
- struct unpack_pds_GetPortAutomaticTracking_t
- struct pack_pds_SetPortAutomaticTracking_t
- struct pack_pds_StartPDSTrackingSessionExt_t
- struct pack_pds_PDSInjectTimeReference_t
- struct unpack_pds_GetXTRAAutomaticDownload_t
- struct pack_pds_SetXTRAAutomaticDownload_t
- struct unpack_pds_GetXTRANetwork_t
- struct pack_pds_SetXTRANetwork_t
- struct unpack_pds_GetXTRAValidity_t
- struct unpack_pds_GetServiceAutomaticTracking_t
- struct pack_pds_SetServiceAutomaticTracking_t
- struct pack_pds_ResetPDSData_t
- struct pack_pds_SLQSSetAGPSConfig_t
- struct pack_pds_SLQSPDSInjectAbsoluteTimeReference_t

- struct pack_pds_SLQSGetAGPSConfig_t
- struct unpack_pds_SLQSGetAGPSConfig_t
- struct pack_pds_SLQSPDSInjectPositionData_t
- struct unpack_pds_SLQSGetGPSStateInfo_t
- struct pack_pds_SLQSSetPositionMethodState_t
- struct pack_pds_SetEventReportCallback_t
- struct unpack_pds_SetEventReport_Ind_t
- struct unpack_pds_SetPdsState_Ind_t

## Typedefs

- typedef unpack_result_t unpack_pds_SetPDSState_t
- typedef unpack_result_t unpack_pds_SetPDSDefaults_t
- typedef unpack_result_t unpack_pds_SetPortAutomaticTracking_t
- typedef unpack_result_t unpack_pds_StartPDSTrackingSessionExt_t
- typedef unpack_result_t unpack_pds_StopPDSTrackingSession_t
- typedef unpack_result_t unpack_pds_PDSInjectTimeReference_t
- typedef unpack_result_t unpack_pds_SetXTRAAutomaticDownload_t
- typedef unpack_result_t unpack_pds_SetXTRANetwork_t
- typedef unpack_result_t unpack_pds_ForceXTRADownload_t
- typedef unpack_result_t unpack_pds_SetServiceAutomaticTracking_t
- typedef unpack_result_t unpack_pds_ResetPDSData_t
- typedef unpack_result_t unpack_pds_SLQSSetAGPSConfig_t
- typedef unpack_result_t unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t
- typedef unpack_result_t unpack_pds_SLQSPDSInjectPositionData_t
- typedef unpack_result_t unpack_pds_SLQSPDSDeterminePosition_t
- typedef unpack_result_t unpack_pds_SLQSSetPositionMethodState_t
- typedef unpack_result_t unpack_pds_SetEventReportCallback_t

## Functions

- int pack_pds_GetPDSState (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_pds_GetPDSState (uint8_t ∗pResp, uint16_t respLen, unpack_pds_GetPDSState_t ∗pOutput)
- int pack_pds_SetPDSState (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_pds_SetPDSState_t ∗pReqParam)
- int unpack_pds_SetPDSState (uint8_t ∗pResp, uint16_t respLen, unpack_pds_SetPDSState_t ∗pOutput)
- int pack_pds_GetPDSDefaults (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_pds_GetPDSDefaults (uint8_t ∗pResp, uint16_t respLen, unpack_pds_GetPDSDefaults_t ∗pOutput)
- int pack_pds_SetPDSDefaults (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_pds_SetPDSDefaults_t ∗pReqParam)
- int unpack_pds_SetPDSDefaults (uint8_t ∗pResp, uint16_t respLen, unpack_pds_SetPDSDefaults_t ∗pOutput)
- int pack_pds_GetPortAutomaticTracking (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_pds_GetPortAutomaticTracking (uint8_t ∗pResp, uint16_t respLen, unpack_pds_GetPortAutomaticTracking_t ∗pOutput)
- int pack_pds_SetPortAutomaticTracking (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_pds_SetPortAutomaticTracking_t ∗pReqParam)
- int unpack_pds_SetPortAutomaticTracking (uint8_t ∗pResp, uint16_t respLen, unpack_pds_SetPortAutomaticTracking_t ∗pOutput)
- int pack_pds_StartPDSTrackingSessionExt (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_pds_StartPDSTrackingSessionExt_t ∗pReqParam)
- int unpack_pds_StartPDSTrackingSessionExt (uint8_t ∗pResp, uint16_t respLen, unpack_pds_StartPDSTrackingSessionExt_t ∗pOutput)

- int pack_pds_StopPDSTrackingSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_StopPDSTrackingSession (uint8_t *pResp, uint16_t respLen, unpack_pds_StopPDSTrackingSession_t *pOutput)
- int pack_pds_PDSInjectTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_PDSInjectTimeReference_t *pReqParam)
- int unpack_pds_PDSInjectTimeReference (uint8_t *pResp, uint16_t respLen, unpack_pds_PDSInjectTimeReference_t *pOutput)
- int pack_pds_GetXTRAAutomaticDownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRAAutomaticDownload (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAAutomaticDownload_t *pOutput)
- int pack_pds_SetXTRAAutomaticDownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRAAutomaticDownload_t *pReqParam)
- int unpack_pds_SetXTRAAutomaticDownload (uint8_t *pResp, uint16_t respLen, unpack_pds_SetXTRAAutomaticDownload_t *pOutput)
- int pack_pds_GetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRANetwork (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRANetwork_t *pOutput)
- int pack_pds_SetXTRANetwork (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetXTRANetwork_t *pReqParam)
- int unpack_pds_SetXTRANetwork (uint8_t *pResp, uint16_t respLen, unpack_pds_SetXTRANetwork_t *pOutput)
- int pack_pds_GetXTRAValidity (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetXTRAValidity (uint8_t *pResp, uint16_t respLen, unpack_pds_GetXTRAValidity_t *pOutput)
- int pack_pds_ForceXTRADownload (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_ForceXTRADownload (uint8_t *pResp, uint16_t respLen, unpack_pds_ForceXTRADownload_t *pOutput)
- int pack_pds_GetServiceAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_GetServiceAutomaticTracking (uint8_t *pResp, uint16_t respLen, unpack_pds_GetServiceAutomaticTracking_t *pOutput)
- int pack_pds_SetServiceAutomaticTracking (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SetServiceAutomaticTracking_t *pReqParam)
- int unpack_pds_SetServiceAutomaticTracking (uint8_t *pResp, uint16_t respLen, unpack_pds_SetServiceAutomaticTracking_t *pOutput)
- int pack_pds_ResetPDSData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_ResetPDSData_t *pReqParam)
- int unpack_pds_ResetPDSData (uint8_t *pResp, uint16_t respLen, unpack_pds_ResetPDSData_t *pOutput)
- int pack_pds_SLQSSetAGPSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSSetAGPSConfig_t *pReqParam)
- int unpack_pds_SLQSSetAGPSConfig (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSSetAGPSConfig_t *pOutput)
- int pack_pds_SLQSPDSInjectAbsoluteTimeReference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t *pReqParam)
- int unpack_pds_SLQSPDSInjectAbsoluteTimeReference (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t *pOutput)
- int pack_pds_SLQSGetAGPSConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSGetAGPSConfig_t *pReqParam)
- int unpack_pds_SLQSGetAGPSConfig (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSGetAGPSConfig_t *pOutput)
- int pack_pds_SLQSPDSInjectPositionData (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_pds_SLQSPDSInjectPositionData_t *pReqParam)
- int unpack_pds_SLQSPDSInjectPositionData (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSPDSInjectPositionData_t *pOutput)
- int pack_pds_SLQSPDSDeterminePosition (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSPDSDeterminePosition (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSPDSDeterminePosition_t *pOutput)

- int pack_pds_SLQSGetGPSStateInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_pds_SLQSGetGPSStateInfo (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSGetGPS-StateInfo_t *pOutput)
- int pack_pds_SLQSSetPositionMethodState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_-pds_SLQSSetPositionMethodState_t *pReqParam)
- int unpack_pds_SLQSSetPositionMethodState (uint8_t *pResp, uint16_t respLen, unpack_pds_SLQSSet-PositionMethodState_t *pOutput)
- int pack_pds_SetEventReportCallback (pack_qmi_t *pCtx, uint8_t *pReq, uint16_t *pLen, pack_pds_Set-EventReportCallback_t *pReqParam)
- int unpack_pds_SetEventReportCallback (uint8_t *pResp, uint16_t respLen, unpack_pds_SetEventReport-Callback_t *pOutput)
- int unpack_pds_SetEventReport_Ind (uint8_t *pResp, uint16_t respLen, unpack_pds_SetEventReport_Ind_t *pOutput)
- int unpack_pds_SetPdsState_Ind (uint8_t *pResp, uint16_t respLen, unpack_pds_SetPdsState_Ind_t *p-Output)

## 9.12.1 Typedef Documentation

### 9.12.1.1 typedef unpack_result_t unpack_pds_ForceXTRADownload_t

### 9.12.1.2 typedef unpack_result_t unpack_pds_PDSInjectTimeReference_t

### 9.12.1.3 typedef unpack_result_t unpack_pds_ResetPDSData_t

### 9.12.1.4 typedef unpack_result_t unpack_pds_SetEventReportCallback_t

### 9.12.1.5 typedef unpack_result_t unpack_pds_SetPDSDefaults_t

### 9.12.1.6 typedef unpack_result_t unpack_pds_SetPDSState_t

### 9.12.1.7 typedef unpack_result_t unpack_pds_SetPortAutomaticTracking_t

### 9.12.1.8 typedef unpack_result_t unpack_pds_SetServiceAutomaticTracking_t

### 9.12.1.9 typedef unpack_result_t unpack_pds_SetXTRAAutomaticDownload_t

### 9.12.1.10 typedef unpack_result_t unpack_pds_SetXTRANetwork_t

### 9.12.1.11 typedef unpack_result_t unpack_pds_SLQSPDSDeterminePosition_t

### 9.12.1.12 typedef unpack_result_t unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t

### 9.12.1.13 typedef unpack_result_t unpack_pds_SLQSPDSInjectPositionData_t

### 9.12.1.14 typedef unpack_result_t unpack_pds_SLQSSetAGPSConfig_t

### 9.12.1.15 typedef unpack_result_t unpack_pds_SLQSSetPositionMethodState_t

### 9.12.1.16 typedef unpack_result_t unpack_pds_StartPDSTrackingSessionExt_t

### 9.12.1.17 typedef unpack_result_t unpack_pds_StopPDSTrackingSession_t

## 9.12.2 Function Documentation

#### 9.12.2.1 int pack_pds_ForceXTRADownload ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Forces the XTRA database to be downloaded to the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.2 int pack_pds_GetPDSDefaults ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get the default tracking session configuration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.3 int pack_pds_GetPDSState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get current PDS state pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.4 int pack_pds_GetPortAutomaticTracking ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Returns the automatic tracking configuration for the NMEA COM port pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.5 int pack_pds_GetServiceAutomaticTracking ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

automatic tracking state for the service pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.6 int pack_pds_GetXTRAAutomaticDownload ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

XTRA automatic database download configuration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.7 int pack_pds_GetXTRANetwork ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

XTRA WWAN network preference parameter pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.8 int pack_pds_GetXTRAValidity ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

XTRA database validity period parameter pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.9 int pack_pds_PDSInjectTimeReference ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_PDSInjectTimeReference_t * *pReqParam* )

Inject system time parameters pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.10 int pack_pds_ResetPDSData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_ResetPDSData_t * pReqParam )

Resets the specified PDS data pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.11 int pack_pds_SetEventReportCallback ( pack_qmi_t * pCtx, uint8_t * pReq, uint16_t * pLen, pack_pds_SetEventReportCallback_t * pReqParam )

Enables/disables the PDS Event report callback function pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReq* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.12 int pack_pds_SetPDSDefaults ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SetPDSDefaults_t * pReqParam )

Sets the default tracking session configuration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.13 int pack_pds_SetPDSState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPDSState_t * *pReqParam* )

Sets the PDS state pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.14 int pack_pds_SetPortAutomaticTracking ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetPortAutomaticTracking_t * *pReqParam* )

Sets the automatic tracking configuration for the NMEA COM port pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.15 int pack_pds_SetServiceAutomaticTracking ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetServiceAutomaticTracking_t * *pReqParam* )

Sets the automatic tracking state for the service pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.16 int pack_pds_SetXTRAAutomaticDownload ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetXTRAAutomaticDownload_t * *pReqParam* )

Sets the XTRA automatic database download configuration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.17 int pack_pds_SetXTRANetwork ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SetXTRANetwork_t * *pReqParam* )

Sets the XTRA WWAN network preference pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.18 int pack_pds_SLQSGetAGPSConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSGetAGPSConfig_t * *pReqParam* )

Gets the PDS AGPS (MS-based) configuration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.19 int pack_pds_SLQSGetGPSStateInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Queries the MSM GPS server for receiver state information pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.20 int pack_pds_SLQSPDSDeterminePosition ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.21 int pack_pds_SLQSPDSInjectAbsoluteTimeReference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t * pReqParam )

Injects a absolute time reference into the PDS engine pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.22 int pack_pds_SLQSPDSInjectPositionData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSPDSInjectPositionData_t * pReqParam )

Injects position data into the PDS engine pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.23 int pack_pds_SLQSSetAGPSConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSSetAGPSConfig_t * pReqParam )

Sets the PDS AGPS (MS-based) configuration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.24 int pack_pds_SLQSSetPositionMethodState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_SLQSSetPositionMethodState_t * *pReqParam* )

Sets the state of positioning methods for the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.25 int pack_pds_StartPDSTrackingSessionExt ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_pds_StartPDSTrackingSessionExt_t * *pReqParam* )

Starts a PDS tracking session pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.26 int pack_pds_StopPDSTrackingSession ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Stops a PDS tracking session pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.27 int unpack_pds_ForceXTRADownload ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_ForceXTRADownload_t * *pOutput* )

Forces the XTRA database to be downloaded to the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.28 int unpack_pds_GetPDSDefaults ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPDSDefaults_t * *pOutput* )

Get the default tracking session configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.29 int unpack_pds_GetPDSState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPDSState_t * *pOutput* )

Get current PDS state unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.12.2.30 int unpack_pds_GetPortAutomaticTracking ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetPortAutomaticTracking_t * *pOutput* )

Returns the automatic tracking configuration for the NMEA COM port unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.12.2.31 int unpack_pds_GetServiceAutomaticTracking ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_GetServiceAutomaticTracking_t * *pOutput* )

automatic tracking state for the service unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.32 int unpack_pds_GetXTRAAutomaticDownload ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_pds_GetXTRAAutomaticDownload_t** * *pOutput* )

XTRA automatic database download configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.33 int unpack_pds_GetXTRANetwork ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_pds_GetXTRANetwork_t** * *pOutput* )

XTRA WWAN network preference parameter unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.34 int unpack_pds_GetXTRAValidity ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_pds_GetXTRAValidity_t** * *pOutput* )

XTRA database validity period parameter unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.35 int unpack_pds_PDSInjectTimeReference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_PDSInjectTimeReference_t * *pOutput* )

Inject system time parameters unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.36 int unpack_pds_ResetPDSData ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_ResetPDSData_t * *pOutput* )

Resets the specified PDS data unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.37 int unpack_pds_SetEventReport_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetEventReport_Ind_t * *pOutput* )

PDS Event Report Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.38 int unpack_pds_SetEventReportCallback ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetEventReportCallback_t * *pOutput* )

Enables/disables the PDS Event report callback function unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.39 int unpack_pds_SetPDSDefaults ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetPDSDefaults_t * *pOutput* )

Sets the default tracking session configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.40 int unpack_pds_SetPDSState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetPDSState_t * *pOutput* )

Sets the PDS state unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.41 int unpack_pds_SetPdsState_Ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_pds_SetPdsState_Ind_t** * *pOutput* )

PDS State Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.42 int unpack_pds_SetPortAutomaticTracking ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_pds_SetPortAutomaticTracking_t** * *pOutput* )

Sets the automatic tracking configuration for the NMEA COM port unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.43 int unpack_pds_SetServiceAutomaticTracking ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_pds_SetServiceAutomaticTracking_t** * *pOutput* )

Sets the automatic tracking state for the service unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.44 int unpack_pds_SetXTRAAutomaticDownload ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetXTRAAutomaticDownload_t * *pOutput* )

Sets the XTRA automatic database download configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.45 int unpack_pds_SetXTRANetwork ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SetXTRANetwork_t * *pOutput* )

Sets the XTRA WWAN network preference unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.46 int unpack_pds_SLQSGetAGPSConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSGetAGPSConfig_t * *pOutput* )

Gets the PDS AGPS (MS-based) configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.47 int unpack_pds_SLQSGetGPSStateInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSGetGPSStateInfo_t * *pOutput* )

Queries the MSM GPS server for receiver state information unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.48 int unpack_pds_SLQSPDSDeterminePosition ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSPDSDeterminePosition_t * *pOutput* )

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.49 int unpack_pds_SLQSPDSInjectAbsoluteTimeReference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t * *pOutput* )

Injects a absolute time reference into the PDS engine unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.50 int unpack_pds_SLQSPDSInjectPositionData ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSPDSInjectPositionData_t * *pOutput* )

Injects position data into the PDS engine unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.51 int unpack_pds_SLQSSetAGPSConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSSetAGPSConfig_t * *pOutput* )

Sets the PDS AGPS (MS-based) configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.52 int unpack_pds_SLQSSetPositionMethodState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_SLQSSetPositionMethodState_t * *pOutput* )

Sets the state of positioning methods for the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.53 int unpack_pds_StartPDSTrackingSessionExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_StartPDSTrackingSessionExt_t * *pOutput* )

Starts a PDS tracking session unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.12.2.54 int unpack_pds_StopPDSTrackingSession ( uint8_t * *pResp*, uint16_t *respLen*, unpack_pds_StopPDSTrackingSession_t * *pOutput* )

Stops a PDS tracking session unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.13 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

### 9.13.1 Detailed Description

Network Access Service API Band Classes table.

### 9.13.2 Band Classes (Value - Description)

- 0 - CDMA Band Class 0
- 1 - CDMA Band Class 1
- 3 - CDMA Band Class 3
- 4 - CDMA Band Class 4
- 5 - CDMA Band Class 5
- 6 - CDMA Band Class 6
- 7 - CDMA Band Class 7
- 8 - CDMA Band Class 8
- 9 - CDMA Band Class 9
- 10 - CDMA Band Class 10
- 11 - CDMA Band Class 11
- 12 - CDMA Band Class 12
- 13 - CDMA Band Class 13
- 14 - CDMA Band Class 14
- 15 - CDMA Band Class 15
- 16 - CDMA Band Class 16
- 17 - CDMA Band Class 17
- 18 - CDMA Band Class 18
- 19 - CDMA Band Class 19
- 40 - GSM 450
- 41 - GSM 480
- 42 - GSM 750
- 43 - GSM 850
- 44 - GSM 900 (Extended)
- 45 - GSM 900 (Primary)
- 46 - GSM 900 (Railways)
- 47 - GSM 1800
- 48 - GSM 1900
- 80 - WCDMA 2100
- 81 - WCDMA PCS 1900
- 82 - WCDMA DCS 1800
- 83 - WCDMA 1700 (US)
- 84 - WCDMA 850
- 85 - WCDMA 800

- 86 - WCDMA 2600
- 87 - WCDMA 900
- 88 - WCDMA 1700 (Japan)
- 90 - WCDMA 1500 band (Japan)
- 91 - WCDMA 850 band (Japan)
- < Reserved 89, 92-109 for WCDMA band classes>
- 110 - WLAN US 2400 MHz
- 111 - WLAN JAPAN 2400 MHz
- 112 - WLAN EUROPEAN 2400 MHz
- 113 - WLAN FRANCE 2400 MHz
- 114 - WLAN SPAIN 2400 MHz
- 115 - WLAN US 5000 MHz band
- 116 - WLAN JAPAN 5000 MHz
- 117 - WLAN EUROPEAN 5000 MHz
- 118 - WLAN FRANCE 5000 MHz
- 119 - WLAN SPAIN 5000 MHz

#### 9.13.2.1 LTE Bands

- 28 - LTE Band Class 28
- 39 - LTE Band Class 39
- 40 - LTE Band Class 40
- 41 - LTE Band Class 41
- 120 - FDD UL:1920-1980; DL:2110-2170; E-UTRA Operating Band 1
- 121 - FDD UL:1850-1910; DL:1930-1990; E-UTRA Operating Band 2
- 122 - FDD UL:1710-1785; DL:1805-1880; E-UTRA Operating Band 3
- 123 - FDD UL:1710-1755; DL:2110-2155; E-UTRA Operating Band 4
- 124 - FDD UL: 824- 849; DL: 869- 894; E-UTRA Operating Band 5
- 125 - FDD UL: 830- 840; DL: 875- 885; E-UTRA Operating Band 6
- 126 - FDD UL:2500-2570; DL:2620-2690; E-UTRA Operating Band 7
- 127 - FDD UL: 880- 915; DL: 925- 960; E-UTRA Operating Band 8
- 128 - FDD UL:1749.9-1784.9; DL:1844.9-1879.9; E-UTRA Operating Band 9
- 129 - FDD UL:1710-1770; DL:2110-2170; E-UTRA Operating Band 10
- 130 - FDD UL:1427.9-1452.9; DL:1475.9-1500.9; E-UTRA Operating Band 11
- 131 - FDD UL:698-716; DL:728-746; E-UTRA Operating Band 12
- 132 - FDD UL: 777- 787; DL: 746-756; E-UTRA Operating Band 13
- 133 - FDD UL: 788- 798; DL: 758-768; E-UTRA Operating Band 14

- 134 - FDD UL: 704-716; DL: 734-746; E-UTRA Operating Band 17
- 135 - TDD LTE UL: 1900-1920; DL: 1900-1920; E-UTRA Operating Band 33
- 136 - TDD LTE UL: 2010-2025; DL: 2010-2025; E-UTRA Operating Band 34
- 137 - TDD LTE UL: 1850-1910; DL: 1850-1910; E-UTRA Operating Band 35
- 138 - TDD LTE UL: 1930-1990; DL: 1930-1990; E-UTRA Operating Band 36
- 139 - TDD LTE UL: 1910-1930; DL: 1910-1930; E-UTRA Operating Band 37
- 140 - TDD LTE UL: 2570-2620; DL: 2570-2620; E-UTRA Operating Band 38
- 141 - TDD LTE UL: 1880-1920; DL: 1880-1920; E-UTRA Operating Band 39
- 142 - TDD LTE UL: 2300-2400; DL: 2300-2400; E-UTRA Operating Band 40
- 143 - FDD LTE UL: 815-830; DL: 860-875; E-UTRA Operating Band 18
- 144 - FDD LTE UL: 830-845; DL: 875-890; E-UTRA Operating Band 19
- 145 - FDD LTE UL: 832-862; DL: 791-821; E-UTRA Operating Band 20
- 146 - FDD LTE UL: 1447.9-1462.9; DL: 1495.9-1510.9; E-UTRA Operating Band 21
- 147 - FDD LTE UL: 1626.5-1660.5; DL: 1525-1559; E-UTRA Operating Band 24
- 148 - FDD LTE UL: 1850-19195; DL: 1930-1995; E-UTRA Operating Band 25
- 149 - TDD LTE UL: 2496-2690; DL: 2496-2690; E-UTRA Operating Band 41
- 150 - TDD LTE UL: 3400-3600; DL: 3400-3600; E-UTRA Operating Band 42
- 151 - TDD LTE UL: 3600-3800; DL: 3600-3800; E-UTRA Operating Band 43
- 200 - TD-SCDMA Band A
- 201 - TD-SCDMA Band B
- 202 - TD-SCDMA Band C
- 203 - TD-SCDMA Band D
- 204 - TD-SCDMA Band E
- 205 - TD-SCDMA Band F
- 250 - NR5G Band 1
- 251 - NR5G Band 2

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.14 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

### 9.14.1 Detailed Description

Call Control Return Reasons table.

### 9.14.2 S1

- 0x01 - QMI_VOICE_REASON_FWD_UNCONDITIONAL - Unconditional call forwarding
- 0x02 - QMI_VOICE_REASON_FWD_MOBILEBUSY - Forward when the mobile is busy
- 0x03 - QMI_VOICE_REASON_FWD_NOREPLY - Forward when there is no reply
- 0x04 - QMI_VOICE_REASON_FWD_UNREACHABLE - Forward when the call is unreachable
- 0x05 - QMI_VOICE_REASON_FWD_ALLFORWARDING - All forwarding
- 0x06 - QMI_VOICE_REASON_FWD_ALLCONDITIONAL - All conditional forwarding
- 0x07 - QMI_VOICE_REASON_BARR_ALLOUTGOING - All outgoing
- 0x08 - QMI_VOICE_REASON_BARR_OUTGOINGINT - Outgoing internal
- 0x09 - QMI_VOICE_REASON_BARR_OUTGOINGINTEXTOHOME - Outgoing external to home
- 0x0A - QMI_VOICE_REASON_BARR_ALLINCOMING - All incoming
- 0x0B - QMI_VOICE_REASON_BARR_INCOMINGROAMING - Roaming incoming
- 0x0C - QMI_VOICE_REASON_BARR_ALLBARRING - All calls are barred
- 0x0D - QMI_VOICE_REASON_BARR_ALLOUTGOINGBARRING - All outgoing calls are barred
- 0x0E - QMI_VOICE_REASON_BARR_ALLINCOMINGBARRING - All incoming calls are barred
- 0x0F - QMI_VOICE_REASON_CALLWAITING - Call waiting
- 0x10 - VOICE_CC_SUPS_RESULT_REASON_CLIP - CLIP
- 0x11 - VOICE_CC_SUPS_RESULT_REASON_CLIR - CLIR
- 0x12 - VOICE_CC_SUPS_RESULT_REASON_COLP - COLP
- 0x13 - VOICE_CC_SUPS_RESULT_REASON_COLR - COLR
- 0x14 - VOICE_CC_SUPS_RESULT_REASON_CNAP - CNAP
- 0xFF - Not Available

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.15 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

### 9.15.1 Detailed Description

Wireless Data Service Call End Reasons.

### 9.15.2 Call end reason codes (Code - Reason)

#### 9.15.2.1 Technology-agnostic call end reasons

- 1 - Reason unspecified, check the verbose call end reason
- 2 - Client ended the call
- 3 - Device has no service
- 4 - Call ended abnormally
- 5 - Received release from base station; no reason given
- 6 - Access attempt already in progress; SD2.0 only
- 7 - Access failure for reason other than the above
- 8 - Call rejected because of redirection or handoff
- 9 - Call failed because close is in progress
- 10 - Authentication failed, 3GPP equivalent ESM(EPS Session Management) cause code value 29, User authentication failed
- 11 - Call ended because of internal call end. This error code is returned when data call is brought down due to some unknown error, such as not specific to any RAT
- 12 - Call ended because of internal error. This error code is returned when data call is brought down due to some unspecified internal error, such as NULL pointer
- 13 - Internal unknown cause code

#### 9.15.2.2 EVDO CDMA 1xEV-DO

- 500 - Device is CDMA-locked until power cycle
- 501 - Received intercept from base station; origination only
- 502 - Received reorder from base station; origination only
- 503 - Received release from base station; service option reject
- 504 - Received incoming call from base station
- 505 - Received alert stop from base station; incoming only
- 506 - Received end activation; OTASP call only
- 507 - Max access probes transmitted
- 508 - Concurrent service is not supported by base station
- 509 - No response received from base station
- 510 - Call rejected by the base station; CDMA only
- 511 - Concurrent services requested were not compatible; CDMA only
- 512 - Corresponds to CM CALL ORIG ERR ALREADY IN TC
- 513 - Used if Call manager subsystem is ending a GPS call in favor of a user call
- 514 - Used if Call manager subsystem is ending a SMS call in favor of a user call
- 515 - CDMA Only; Device has no service

#### 9.15.2.3 WCDMA/GSM call end reasons

- 1000 - Call origination request failed; WCDMA/GSM Only
- 1001 - Client rejected the incoming call; WCDMA/GSM Only
- 1002 - Device has no UMTS service; WCDMA/GSM Only
- 1003 - Network ended the call, look in cc call; WCDMA/GSM Only
- 1004 - LLC(Logical Link Control) or SNDCP(Sub Network Dependent Convergence Protocol) failure
- 1005 - Insufficient resources, 3GPP equivalent ESM(EPS Session Management) cause code value 26, Insufficient resources
- 1006 - Service option temporarily out of order, 3GPP equivalent ESM(EPS Session Management) cause code value 34, Service option temporarily out of order
- 1007 - PTI already used, 3GPP equivalent ESM(EPS Session Management) cause code value 35, PTI(-Procedure Transaction Identity) already in use
- 1008 - Regular PDP context deactivation, 3GPP equivalent ESM(EPS Session Management) cause code value 36, Regular deactivation
- 1009 - Network failure, 3GPP equivalent ESM(EPS Session Management) cause code value 38, Network failure
- 1010 - Reactivation requested, 3GPP equivalent ESM(EPS Session Management) cause code value 39, Reactivation requested
- 1011 - Protocol error, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 111, Protocol error, unspecified
- 1012 - Operator determined barring, 3GPP equivalent ESM(EPS Session Management) cause code value 8, Operator Determined Barring
- 1013 - Unknown or missing Access Point Name (APN), 3GPP equivalent ESM(EPS Session Management) cause code value 27, Missing or unknown APN
- 1014 - Unknown PDP address or PDP type, 3GPP equivalent ESM(EPS Session Management) cause code value 28, Unknown PDN type
- 1015 - Activation rejected by GGSN, 3GPP equivalent ESM(EPS Session Management) cause code value 30, Requested rejected by Serving GW or PDN GW
- 1016 - Activation rejected, unspecified, 3GPP equivalent ESM(EPS Session Management) cause code value 31, Request rejected, unspecified
- 1017 - Service option not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 32, Service option not supported
- 1018 - Requested service option not subscribed, 3GPP equivalent ESM(EPS Session Management) cause code value 33, Requested service option not subscribed
- 1019 - EPS Quality of Service (QoS) not accepted, 3GPP equivalent ESM(EPS Session Management) cause code value 37, EPS QoS not accepted
- 1020 - Semantic error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 41, Semantic error in the TFT operation
- 1021 - Syntactical error in the TFT operation, 3GPP equivalent ESM(EPS Session Management) cause code value 42, Syntactical error in the TFT operation
- 1022 - Unknown PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 43, Invalid EPS bearer identity

- 1023 - Semantic errors in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 44, Semantic errors in packet filter(s)
- 1024 - Syntactical error in packet filter(s), 3GPP equivalent ESM(EPS Session Management) cause code value 45, Syntactical errors in packet filter(s)
- 1025 - PDP context without TFT already activated, 3GPP equivalent ESM(EPS Session Management) cause code value 46, Unused
- 1026 - Invalid transaction identifier value, 3GPP equivalent ESM(EPS Session Management) cause code value 81, Invalid PTI value
- 1027 - Semantically incorrect message, 3GPP equivalent ESM(EPS Session Management) cause code value 95, Semantically incorrect message
- 1028 - Invalid mandatory information, 3GPP equivalent ESM(EPS Session Management) cause code value 96, Invalid mandatory information
- 1029 - Message type non-existent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 97, Message type non-existent or not implemented
- 1030 - Message not compatible with state, 3GPP equivalent ESM(EPS Session Management) cause code value 98, Message type not compatible with the protocol state
- 1031 - Information element nonexistent or not implemented, 3GPP equivalent ESM(EPS Session Management) cause code value 99, Information element non-existent or not implemented
- 1032 - Conditional information element error, 3GPP equivalent ESM(EPS Session Management) cause code value 100, Conditional IE error
- 1033 - Message not compatible with protocol state, 3GPP equivalent ESM(EPS Session Management) cause code value 101, Message not compatible with the protocol state
- 1034 - APN restriction value incompatible with active PDP context, 3GPP equivalent ESM(EPS Session Management) cause code value 112, APN restriction value incompatible with
  - active EPS bearer context
- 1035 - No GPRS context present
- 1036 - Requested feature not supported, 3GPP equivalent ESM(EPS Session Management) cause code value 40, Feature not supported
- 1037 - Illegal MS, 3GPP equivalent EMM(EPS Mobility Management) cause code value 3, Illegal UE (MS)
- 1038 - Illegal ME, 3GPP equivalent EMM(EPS Mobility Management) cause code value 6, Illegal ME. This error code is sent to the MS if the ME used is not acceptable
  - to the network, e.g. blacklisted
- 1039 - GPRS and non GPRS services not allowed
- 1040 - GPRS services not allowed
- 1041 - MS identity not derived by the network, 3GPP equivalent EMM(EPS Mobility Management) cause code value 9, UE (MS) Identify cannot be derived by the network
- 1042 - Implicitly detached, 3GPP equivalent EMM(EPS Mobility Management) cause code value 10, Implicitly Detached
- 1043 - PLMN not allowed, 3GPP equivalent EMM(EPS Mobility Management) cause code value 11, PLMN not allowed
- 1044 - LA not allowed, this cause is sent to the MS if it requests location updating in a location area where the HPLMN determines that the MS, by subscription, is not allowed to operate.
- 1045 - GPRS services not allowed in PLMN

- 1046 - PDP duplicate
- 1047 - UE radio access technology change
- 1048 - app preempted
- 1049 - Congestion, This cause is sent if the service request or LOCATION UPDATING REQUEST message cannot be actioned because of congestion (e.g. congestion of the MSC or SGSN or GGSN or PDN Gateway; no channel; facility busy/congested etc.).
- 1050 - No PDP context activated
- 1051 - Access class DSAC rejection

#### 9.15.2.4 EVDO CDMA 1xEV-DO

- 1500 - Abort connection setup due to the reception of a Connection Deny message with deny code set to either general or network busy.
- 1501 - Abort connection setup due to the reception of a Connection Deny message with deny code set to either billing or authentication failure.
- 1502 - Change HDR system due to redirection or PRL not preferred
- 1503 - Exit HDR due to redirection or PRL not preferred
- 1504 - No HDR session
- 1505 - Used if Call manager is ending an HDR call origination in favor of a GPS fix
- 1506 - Connection setup timeout
- 1507 - Call manager released HDR call so 1x call can continue

#### 9.15.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

#### 9.15.2.6 Mobile IP call end reasons (Type=1)

- 64 - MIP(Mobile IP) FA(Foreign Agent) ERR REASON UNSPECIFIED, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration with unspecified reason
- 65 - MIP(Mobile IP) FA(Foreign Agent) ERR ADMINISTRATIVELY PROHIBITED, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent administratively prohibited MIP registration
- 66 - MIP(Mobile IP) FA(Foreign Agent) ERR INSUFFICIENT RESOURCES, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to insufficient resources

- 67 - MIP(Mobile IP) FA(Foreign Agent) ERR MOBILE NODE AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because MN-AAA authenticator was wrong
- 68 - MIP(Mobile IP) FA(Foreign Agent) ERR HA AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because of home agent authentication failure
- 69 - MIP(Mobile IP) FA(Foreign Agent) ERR REQUESTED LIFETIME TOO LONG, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because requested lifetime is too long
- 70 - MIP(Mobile IP) FA(Foreign Agent) ERR MALFORMED REQUEST, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to malformed request
- 71 - MIP(Mobile IP) FA(Foreign Agent) ERR MALFORMED REPLY, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to malformed reply
- 72 - MIP(Mobile IP) FA(Foreign Agent) ERR ENCAPSULATION UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because requested encapsulation is unavailable
- 73 - MIP(Mobile IP) FA(Foreign Agent) ERR VJHC UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because VJ Header Compression is unavailable
- 74 - MIP(Mobile IP) FA(Foreign Agent) ERR REVERSE TUNNEL UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because reverse tunnel is unavailable
- 75 - MIP(Mobile IP) FA(Foreign Agent) ERR REVERSE TUNNEL IS MANDATORY AND T BIT NOT SET, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because reverse tunnel is mandatory but not requested by device
- 79 - MIP(Mobile IP) FA(Foreign Agent) ERR DELIVERY STYLE NOT SUPPORTED, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration because delivery style is not supported
- 97 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING NAI, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing NAI
- 98 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING HA, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing Home Agent
- 99 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING HOME ADDR, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing Home Address
- 104 - MIP(Mobile IP) FA(Foreign Agent) ERR UNKNOWN CHALLENGE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to unknown challenge
- 105 - MIP(Mobile IP) FA(Foreign Agent) ERR MISSING CHALLENGE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to missing challenge
- 106 - MIP(Mobile IP) FA(Foreign Agent) ERR STALE CHALLENGE, this error code is returned when the data call bring up fails in MIP setup phase since foreign agent rejected MIP registration due to stale challenge
- 128 - MIP(Mobile IP) FA(Home Agent) ERR REASON UNSPECIFIED, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration with unspecified reason

- 129 - MIP(Mobile IP) FA(Home Agent) ERR ADMINISTRATIVELY PROHIBITED, this error code is returned when the data call bring up fails in MIP setup phase since home agent administratively prohibited MIP registration
- 130 - MIP(Mobile IP) FA(Home Agent) ERR INSUFFICIENT RESOURCES, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to insufficient resources
- 131 - MIP(Mobile IP) FA(Home Agent) ERR MOBILE NODE AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since home agent fails authentication because MN-HA authenticator was wrong
- 132 - MIP(Mobile IP) FA(Home Agent) ERR FA AUTHENTICATION FAILURE, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to FA authentication failure
- 133 - MIP(Mobile IP) FA(Home Agent) ERR REGISTRATION ID MISMATCH, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to registration id mismatch
- 134 - MIP(Mobile IP) FA(Home Agent) ERR MALFORMED REQUEST, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to malformed request
- 136 - MIP(Mobile IP) FA(Home Agent) ERR UNKNOWN HA ADDR, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to unknown Home Agent address. This code is returned by a home agent when the mobile node is performing dynamic home agent address resolution as described in RFC 3220 (IP Mobility Support for IPV4) Sections 3.6.1.1 and 3.6.1.2
- 137 - MIP(Mobile IP) FA(Home Agent) ERR REVERSE TUNNEL UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration because reverse tunnel is unavailable
- 138 - MIP(Mobile IP) FA(Home Agent) ERR REVERSE TUNNEL IS MANDATORY AND T BIT NOT SET, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration because reverse tunnel is mandatory but not requested by device
- 139 - MIP(Mobile IP) FA(Home Agent) ERR ENCAPSULATION UNAVAILABLE, this error code is returned when the data call bring up fails in MIP setup phase since home agent rejected MIP registration due to encapsulation unavailable
- 65536 - MIP ERR REASON UNKNOWN

#### 9.15.2.7 Internal call end reasons (Type=2)

- 200 - INTERNAL MIN, internal error table offset value, no meaningful message to the error.
- 201 - INTERNAL ERROR, this error code is returned when data call is brought down due to some unspecified internal error
- 202 - CALL ENDED
- 203 - INTERNAL UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 204 - UNKNOWN CAUSE CODE, this error code is returned when data call is brought down due to some unknown error
- 205 - CLOSE IN PROGRESS
- 206 - NETWORK INITIATED TERMINATION
- 207 - APP PREEMPTED

- 208 - ERR PDN IPV4 CALL DISALLOWED, this error code is returned when V4 PDN is in throttled state due to network providing only V6 address during the previous VSNCP bring up (subs_limited_to_v6). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 209 - ERR PDN IPV4 CALL THROTTLED, this error code is returned when V4 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv4 PDN is throttled is determined by the IPv4 throttling timers maintained in the profile
- 210 - ERR PDN IPV6 CALL DISALLOWED, this error code is returned when V6 PDN is in throttled state due to network providing only V4 address during the previous VSNCP bring up (subs_limited_to_v4). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 211 - ERR PDN IPV6 CALL THROTTLED, this error code is returned when V6 PDN is in throttled state due to previous VSNCP bring up failure(s). The time for which the IPv6 PDN is throttled is determined by the IPv6 throttling timers maintained in the profile
- 212 - MODEM RESTART
- 213 - PDP PPP NOT SUPPORTED
- 214 - UNPREFERRED RAT, this error code is returned when data call is brought down since the RAT on which the data call is attempted/connected is no longer the preferred RAT
- 215 - PHYS LINK CLOSE IN PROGRESS, this error code is returned when data call bring up is rejected because physical link is in the process of cleanup
- 216 - APN PENDING HANDOVER, this error code is returned when interface bring up is attempted for an APN that is yet to be handed over to target RAT
- 217 - PROFILE BEARER INCOMPATIBLE
- 218 - MMGSDI CARD EVT, this error code is returned when data call is brought down because card got refreshed/removed
- 219 - LPM OR PWR DOWN, this error code is returned when data call is brought down because device is going into lower power mode or powering down
- 220 - APN DISABLED, this error code is returned when APN is disabled in card
- 221 - MPIT EXPIRED, this error code is returned when data call is brought down because maximum PPP inactivity timer expired
- 222 - IPV6 ADDR TRANSFER FAILED
- 223 - TRAT SWAP FAILED
- 224 - EHRPD TO HRPD FALLBACK, this error code is returned when data call is brought down because device falls back from eHRPD to HRPD (not because of OOS on eHRPD but due to operator/spec driven eHRPD to HRPD fallback requirements)
- 225 - MANDATORY APN DISABLED, this error code is returned when any mandatory APN is disabled, and MinApnList Disallow call config item is set to TRUE in device
- 226 - MIP CONFIG FAILURE, this error code is returned when UE is in MIP Only config (QCMIP=2) but MIP config fails on call bring up due to incorrect provisioning
- 227 - INTERNAL_PDN_INACTIVITY_TIMER_EXPIRED, this error code is returned when PDN inactivity timer expired due to no data transmission in a configurable duration of time
- 228 - MAX_V4_CONNECTIONS, this error code is returned when IPv4 data call bring up is rejected because the UE already maintains the allotted maximum number of IPv4 data connections
- 229 - MAX_V6_CONNECTIONS, this error code is returned when IPv6 data call bring up is rejected because the UE already maintains the allotted maximum number of IPv6 data connections

- 230 - APN_MISMATCH, this error code is returned when New PDN bring up is rejected during interface selection because the UE has already allotted the available interfaces for other PDNs
- 231 - IP_VERSION_MISMATCH, this error code is returned when New call bring up is rejected because the existing data call IP type does not match the requested IP type
- 232 - DUN_CALL_DISALLOWED, this error code is returned when DUN call bring up is rejected because the UE is in eHRPD RAT
- 233 - INVALID_PROFILE, this error code is returned when Call bring up was requested with an invalid profile
- 234 - INTERNAL_EPC_NONEPC_TRANSITION, this error code is returned when Data call is rejected or brought down because the UE is in transition between EPC and non-EPC RAT
- 235 - INVALID_PROFILE_ID, this error code is returned when Call bring up was requested with an invalid profile ID
- 236 - INTERNAL_CALL_ALREADY_PRESENT, this error code is returned when a data call with the same policy is already connected or in the process of bring up on another RmNet instance
- 237 - IFACE_IN_USE, this error code is returned when the current IFACE is in use
- 238 - IP_PDP_MISMATCH, this error code is returned when a PPP call is attempted on a PDP profile
- 239 - APN_DISALLOWED_ON_ROAMING, this error code is returned when PDN connection to the APN is disallowed on the roaming network. The DS generates DS_SYS_EVENT_3GPP_ROAMING_DISALLOWED_INFO to inform clients of the APNs that must deregister and disconnect (if the PDN is up) while the UE is on the roaming network. Clients must deregister or disconnect if the APN name corresponding to the client is listed in this system event. Clients can register again if the APN name corresponding to the client is not listed in this system event.
- 240 - APN_PARAM_CHANGE, this error code is returned when Failure to reestablish the PDN with the changed parameters; when APN-related parameters are changed, the PDN associated with the parameters must be brought down and reestablished with the changed parameters
- 241 - IFACE_IN_USE_CFG_MATCH, this error code is returned when the IFACE is already in use with a matching configuration
- 242 - NULL_APN_DISALLOWED, this error code is returned when a PDN is attempted to be brought up with a NULL APN and when a NULL APN is not supported
- 243 - THERMAL_MITIGATION, this error code is returned when the thermal level increases and causes calls to be torn down when normal mode of operation is not allowed
- 244 - SUBS_ID_MISMATCH, this error code is returned when a new call bring up is rejected due to a mismatch between the subs_id in the profile and the subs_id in the ACL policy information
- 245 - DATA_SETTINGS_DISABLED, this error code is returned when the PDN connection to a given APN is disallowed because data is disabled from the device user interface (UI) settings
- 246 - DATA_ROAMING_SETTINGS_DISABLED, this error code is returned when the PDN Connection to a given APN is disallowed because data roaming is disabled from the device UI settings and the UE is roaming
- 247 - APN_FORMAT_INVALID, this error code is returned when the APN specified in the policy does not follow the format specified in the 3GPP Specification
- 248 - DDS_CALL_ABORT, this error code is returned when a DDS switch occurs; the data call is brought down with the DDS abort reason
- 249 - VALIDATION_FAILURE, this error code is returned when a data call is brought down due to an internal validation failure

#### 9.15.2.8 Call Manager defined call end reasons (Type=3)

- 500 - CDMA LOCK, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to device in CDMA locked state
- 501 - INTERCEPT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an intercept order from the base station
- 502 - REORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a reorder from base station
- 503 - REL SO REJ, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to receiving a release from base station with reason: SO Reject
- 504 - INCOM CALL, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since it received an incoming call from base station
- 505 - ALERT STOP, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RL/FL fade (or) receiving call release from base stations
- 506 - ACTIVATION, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to channel acquisition failures. This indicates that device has failed acquiring all the channels in the PRL
- 507 - MAX ACCESS PROBE, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes transmitted
- 508 - CCS NOT SUPPORTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since concurrent service is not supported by base station
- 509 - NO RESPONSE FROM BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since there is no response received from base station
- 510 - REJECTED BY BS, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to base station rejecting the call
- 511 - INCOMPATIBLE, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since concurrent services requested were not compatible
- 512 - ALREADY IN TC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since traffic channel is already up for voice calls
- 513 - USER CAL ORIG DURING GPS
- 514 - USER CAL ORIG DURING SMS, this error code is returned when data call is brought down because traffic channel request got rejected since SMS is ongoing
- 515 - NO CDMA SRV, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since device does not have CDMA service
- 516 - MC ABORT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) since MC aborted the origination/conversation
- 517 - PSIST NG, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to persistence test failure
- 518 - UIM NOT PRESENT, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) due to RUIM not present
- 519 - RETRY ORDER, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due to receiving a retry order from base station
- 520 - ACCESS BLOCK, this error code is returned when data call is brought down because traffic channel rejected/released due to Access blocked by base station

- 521 - ACCESS BLOCK ALL, this error code is returned when data call is brought down because traffic channel rejected due to Access blocked by the base station for all mobile devices
- 522 - IS707B MAX ACC, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) due maximum access probes for IS-707B call
- 523 - THERMAL EMERGENCY, this error code is returned when data call is brought down because traffic channel got rejected/released by CM(Call Manager) to put device in thermal emergency
- 524 - CALL ORIG THROTTLED, this error code is returned when data call is brought down because traffic channel request got rejected by CM(Call Manager) since call origination is throttled by DCTM module
- 535 - USER CALL ORIGINATE DURING VOICE CALL, this error code is returned when data call is brought down because traffic channel got released by CM(Call Manager) in favor of voice call or SMS when concurrent voice and data are not supported
- 1000 - CONFERENCE FAILED
- 1001 - INCOMING REJECTED
- 1002 - NO GATEWAY SERVICE
- 1003 - NO GPRS CONTEXT
- 1004 - ILLEGAL MS, This cause is sent to the MS when the network refuses service to the MS either because an identity of the MS is not acceptable to the network or because the MS does not pass the authentication check, i.e. the SRES received from the MS is different from that generated by the network
- 1005 - ILLEGAL ME, This cause is sent to the MS if the ME used is not acceptable to the network, e.g. blacklisted
- 1006 - GPRS SERVICES AND NON GPRS SERVICES NOT ALLOWED
- 1007 - GPRS SERVICES NOT ALLOWED
- 1008 - MS IDENTITY CANNOT BE DERIVED BY THE NETWORK
- 1009 - IMPLICITLY DETACHED, this error code is sent to the MS either if the network has implicitly detached the MS, e.g. some while after the Mobile reachable timer has expired, or if the GMM context data related to the subscription does not exist in the SGSN e.g. because of a SGSN restart.
- 1010 - PLMN NOT ALLOWED, this error code is sent to the MS if it requests location updating in a PLMN where the MS, by subscription or due to operator determined barring is not allowed to operate
- 1011 - LOCAL AREA NOT ALLOWED
- 1012 - GPRS SERVICES NOT ALLOWED IN THIS PLMN
- 1013 - PDP DUPLICATE
- 1014 - USER EQUIPMENT RADIO ACCESS TECHNOLOGY CHANGE
- 1015 - CONGESTION
- 1016 - NO PDP CONEXT ACTIVATED
- 1017 - ACCESS CLASS DSAC REJECTION
- 1018 - PDP ACTIVATE MAX RETRY FAILED
- 1019 - RAB FAILURE
- 1020 - EPS SERVICE NOT ALLOWED
- 1021 - TRACKING AREA NOT ALLOWED
- 1022 - ROAMING NOT ALLOWED IN THIS TRACKING AREA

- 1023 - NO SUITABLE CELLS IN TRACKING AREA
- 1024 - NOT AUTHORIZED FOR THIS CLOSED SUBSCRIBER GROUP
- 1025 - ESM UNKNOWN EPS BEARER CONTEXT
- 1026 - DRB RELEASED AT RRC
- 1027 - NAS SIG CONN RELEASED
- 1028 - EPS MOBILITY MANAGEMENT DETACHED
- 1029 - EPS MOBILITY MANAGEMENT ATTACH FAILED
- 1030 - EPS MOBILITY MANAGEMENT ATTACH STARTED
- 1031 - LTE NAS SERVICE REQ FAILED
- 1032 - ESM(EPS Session Management) ACTIVE DEDICATED BEARER REACTIVATED BY NW
- 1033 - ESM(EPS Session Management) LOWER LAYER FAILURE
- 1034 - ESM(EPS Session Management) SYNC UP WITH NW
- 1035 - ESM(EPS Session Management) NW ACTIVATED DED BEARER WITH ID OF DEF BEARER
- 1036 - ESM(EPS Session Management) BAD OTA MESSAGE
- 1037 - ESM DS REJECTED THE CALL
- 1038 - ESM(EPS Session Management) CONTEXT TRANSFERRED DUE TO IRAT
- 1039 - DS EXPLICIT DEACT
- 1040 - ESM(EPS Session Management) LOCAL CAUSE NONE
- 1041 - LTE NAS SERVICE REQ FAILED NO THROTTLE
- 1042 - ACL FAILURE, This error code should rarely triggered and reported to the application
- 1043 - LTE NAS SERVICE REQ FAILED DS DISALLOW
- 1044 - EMM(EPS Mobility Management) T3417 EXPIRED
- 1045 - EMM(EPS Mobility Management) T3417 EXT EXPIRED
- 1046 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CONN REL
- 1049 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE RLF
- 1050 - LRRC(LTE Radio Resource Control) UL DATA CNF FAILURE CTRL NOT CONN
- 1051 - LRRC(LTE Radio Resource Control) CONN EST FAILURE
- 1052 - LRRC(LTE Radio Resource Control) CONN EST FAILURE ABORTED
- 1053 - LRRC(LTE Radio Resource Control) CONN EST FAILURE ACCESS BARRED
- 1054 - LRRC(LTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRC(LTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRC(LTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRC(LTE Radio Resource Control) CONN EST FAILURE LINK FAILURE

- 1058 - LRRC(LTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRC(LTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRC(LTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRC(LTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRC(LTE Radio Resource Control) CONN REL RLF
- 1063 - LRRC(LTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRC(LTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRC(LTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRC(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

#### 9.15.2.9 3GPP specification defined call end reasons (Type=6)

- 8 - OPERATOR DETERMINED BARRING, this reason code is posted by the MME(Mobility Management Entity) to indicate operator has barred the UE
- 25 - LLC SNDCP 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 endup 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 and information element that is either not defined or defined but not implemented by the equipment sending the ESM cause
- 100 - CONDITIONAL IE ERROR, the reason is posted by the network to indicate receipt of a message that includes a syntactically incorrect information element. This message is ignored by the network.
- 101 - MSG AND PROTOCOL STATE UNCOMPATIBLE, the reason is posted by the network to indicate receipt of a message that cannot be handled in the current network protocol state
- 111 - PROTOCOL ERROR, the reason is posted by the network to indicate a protocol error when no other error applies
- 112 - APN TYPE CONFLICT
- 113 - INVALID PROXY-CALL SESSION CONTROL FUNCTION ADDRESS

#### 9.15.2.10 PPP call end reasons (Type=7)

- 1 - TIMEOUT, this error code is returned when the data call bring up fails in PPP setup due to timeout (For e.g: LCP Conf Ack not received from network)
- 2 - AUTH FAILURE, this error code is returned when the data call bring up fails in PPP setup due to authentication failure
- 3 - OPTION MISMATCH, this error code is returned when the data call bring up fails in PPP setup due option mismatch (e.g: Authentication is required, but not negotiated with network during LCP phase)
- 31 - PAP FAILURE, this error code is returned when the data call bring up fails in PPP setup due to PAP failure
- 32 - CHAP FAILURE, this error code is returned when the data call bring up fails in PPP setup due to CHAP failure
- 33 - CLOSE IN PROGRESS, this error code is returned when the data call bring up fails in PPP setup since PPP is in the process of cleaning the previous PPP session
- -1 - UNKNOWN, this error code is unused

#### 9.15.2.11 EHRPD call end reasons (Type=8)

- 1 - SUBS LIMITED TO V4, this error code is returned when the V6 interface bring up fails because network provided only V4 address for the upcoming PDN
- 2 - SUBS LIMITED TO V6, this error code is returned when the V4 interface bring up fails because network provided only V6 address for the upcoming PDN
- 4 - VSNCP(Vendor Specific Network Control Protocol) TIMEOUT, this error code is returned when the data call bring up fails in VSNCP phase due to VSNCP timeout error
- 5 - VSNCP(Vendor Specific Network Control Protocol) FAILURE, this error code is returned when VSNCP configuration failed during call bring up
- 6 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I GEN ERROR, this error code is returned when the data call bring up fails in VSNCP phase due to general error
- 7 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I UNAUTH APN, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason requested APN is unauthorized
- 8 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN LIMIT EXCEED, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN limit exceeded
- 9 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I NO PDN GW, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason no PDN gateway
- 10 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN GW UNREACH, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN gateway unreachable
- 11 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN GW REJ, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN gateway reject
- 12 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I INSUFF PARAM, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason insufficient parameter
- 13 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I RESOURCE UNAVAIL, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason resource unavailable
- 14 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I ADMIN PROHIBIT, this error code is returned when the data call bring up fails in SNCP phase since network rejected VSNCP config request with reason admin prohibited
- 15 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN ID IN USE, this error code is returned when the data call bring up fails in VSNCP phase because network rejected with reason PDN ID IN USE (or) All existing PDNs are brought down with this end reason because one of the PDN bring up got rejected by network with reason PDN ID IN USE
- 16 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I SUBSCR LIMITATION, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason subscriber limitation
- 17 - VSNCP(Vendor Specific Network Control Protocol) 3GPP2I PDN EXISTS FOR THIS APN, this error code is returned when the data call bring up fails in VSNCP phase since network rejected VSNCP config request with reason PDN exists for this APN

#### 9.15.2.12 IPV6 call end reasons (Type=9)

- 1 - PREFIX UNAVAILABLE, this error code is returned when V6 data call is brought down because device failed to get the prefix from network
- 2 - IPV6 ERR HRPD IPV6 DISABLED, this error code is returned when V6 data call bring up is rejected because IPV6 is disabled in 1X/HRPD mode
- 3 - IPV6 DISABLED, this error code is returned when IPv6 data call bring up is rejected because NV1896 (IPV6 enable) is disabled

Copyright: © 2011-2013 Sierra Wireless, Inc. all rights reserved

## 9.16 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

### 9.16.1 Detailed Description

Carrier Codes table.

### 9.16.2 Carrier Codes (Number - Carrier)

- 0 - no carrier specified
- 1 - Generic
- 2 - Telstra
- 4 - AT&T
- 5 - Verizon
- 11 - Sprint
- 12 - Telefonica
- 101 - Verizon
- 102 - Sprint
- 103 - Alltel
- 104 - Bell Mobility
- 105 - Telus
- 106 - U.S. Cellular
- 107 - Telstra
- 108 - China Unicom
- 109 - Telecom New Zealand
- 110 - SK Telecom
- 111 - Reliance Communications
- 112 - Tata Communications
- 113 - MetroPCS Communications

- 114 - Leap Wireless
- 115 - KDDI
- 116 - Grupo Iusacell
- 117 - China Telecom
- 118 - Open Mobile Handset
- 176 - Rogers
- 177 - NetIndex
- 178 - DNA
- 179 - Big Pond
- 201 - AT&T
- 202 - Vodafone
- 203 - T-Mobile
- 204 - Orange
- 205 - Telefonica
- 206 - Telecom Italia
- 207 - 3
- 208 - O2
- 209 - SFR
- 210 - Swisscom AG
- 211 - China Mobile
- 212 - Telstra
- 213 - Singapore Telecommunications
- 214 - Reliance Telecommunications
- 215 - Bharti Airtel
- 216 - NTT docomo
- 217 - E Mobile
- 218 - Softbank
- 219 - Korea Telecom Freetel
- 220 - SK Telecom
- 221 - Telenor
- 222 - NetCom Norway
- 223 - TeliaSonera
- 224 - América Móvil
- 225 - Brasil Vivo
- 0xFFFFFFFF - Unknown

Copyright: © 2011-2014 Sierra Wireless, Inc. all rights reserved

## 9.17 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

### Macros

- #define __GOBI_API_CODING_SCHEME_H__

### 9.17.1 Detailed Description

Data Coding Scheme.

### 9.17.2 S1

#### 9.17.2.1 Use of bits 3..0\n\n

- Language using the GSM 7 bit default alphabet Bits 3..0
  indicate the language:
  0000 German
  0001 English
  0010 Italian
  0011 French
  0100 Spanish
  0101 Dutch
  0110 Swedish
  0111 Danish
  1000 Portuguese
  1001 Finnish
  1010 Norwegian
  1011 Greek
  1100 Turkish
  1101 Hungarian
  1110 Polish
  1111 Language unspecified

### 9.17.3 Coding Group Bits 7..4(0001)

#### 9.17.3.1 use of bits 3..0

- 0000 GSM 7 bit default alphabet; message preceded by language indication.
  The first 3 characters of the message are a two-character representation of the language encoded according to ISO 639 [12], followed by a CR character. The CR character is then followed by 90 characters of text.
- 0001 UCS2; message preceded by language indication
  The message starts with a two GSM 7-bit default alphabet character representation of the language encoded according to ISO 639. This is padded

to the octet boundary with two bits set to 0 and then followed by 40 characters
of UCS2-encoded message.
An MS not supporting UCS2 coding will present the two character language
identifier followed by improperly interpreted user data.

### 9.17.4 Coding Group Bits 7..4(0010)

#### 9.17.4.1 use of bits 3..0

- 0000 Czech
  0001 Hebrew
  0010 Arabic
  0011 Russian
  0100 Icelandic
  0101..1111 Reserved for other languages using the GSM 7 bit
  default alphabet, with unspecified handling at the MS

### 9.17.5 Coding Group Bits 7..4(0011)

#### 9.17.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit
  default alphabet, with unspecified handling at the MS

### 9.17.6 Coding Group Bits 7..4(01xx)

#### 9.17.6.1 use of bits 3..0

- General Data Coding indication
  - Bits 5..0 indicate the following:
  - Bit 5, if set to 0, indicates the text is uncompressed
  - Bit 5, if set to 1, indicates the text is compressed using the compression algorithm defined in 3GPP TS 23.042
  - Bit 4, if set to 0, indicates that bits 1 to 0 are reserved and have no message class meaning
  - Bit 4, if set to 1, indicates that bits 1 to 0 have a message class meaning: Bit 1 Bit 0 Message Class:
    0 0 Class 0
    0 1 Class 1 Default meaning: ME-specific.
    1 0 Class 2 (U)SIM specific message.
    1 1 Class 3 Default meaning: TE-specific (see 3GPP TS 27.005 )
    Bits 3 and 2 indicate the character set being used, as follows:
    Bit 3 Bit 2 Character set:
    0 0 GSM 7 bit default alphabet
    0 1 8 bit data
    1 0 UCS2 (16 bit) [10]
    1 1 Reserved

### 9.17.7 Coding Group Bits 7..4(1001)

#### 9.17.7.1 Reserved coding groups

- Message with User Data Header (UDH) structure:
  Bit 1 Bit 0 Message Class:
  0 0 Class 0
  0 1 Class 1 Default meaning: ME-specific.
  1 0 Class 2 (U)SIM specific message.
  1 1 Class 3 Default meaning: TE-specific (see 3GPP TS 27.005 [8])
  Bits 3 and 2 indicate the alphabet being used, as follows:
  Bit 3 Bit 2 Alphabet:
  0 0 GSM 7 bit default alphabet
  0 1 8 bit data
  1 0 USC2 (16 bit) [10]
  1 1 Reserved

### 9.17.8 Coding Group Bits 7..4(1010..1101)

#### 9.17.8.1 Reserved coding groups

### 9.17.9 Coding Group Bits 7..4(1110)

#### 9.17.9.1 Defined by the WAP Forum

### 9.17.10 Coding Group Bits 7..4 (1111)

#### 9.17.10.1 Data coding / message handling

- Bit 3 is reserved, set to 0.
  Bit 2 Message coding:
  0 GSM 7 bit default alphabet
  1 8 bit data
  Bit 1 Bit 0 Message Class:
  0 0 No message class.
  0 1 Class 1 user defined.
  1 0 Class 2 user defined.
  1 1 Class 3
  default meaning: TE specific(3GPP TS 27.005)
  Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

### 9.17.11 Macro Definition Documentation

#### 9.17.11.1 #define __GOBI_API_CODING_SCHEME_H__

## 9.18 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

### 9.18.1 Detailed Description

Position Determination Service API GPS Capability Codes.

### 9.18.2 GPS capability (Value - Capability)

- 0 - None
- 1 - Standalone
- 2 - Assisted (including XTRA and implying standalone is also supported)
- 3 - Assisted (without XTRA and implying standalone is also supported)
- 0xFFFFFFFF - Unknown

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.19 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

### 9.19.1 Detailed Description

Device Management Service API Power Modes table.

### 9.19.2 Power Modes (Value - Description)

- 0 - Online (default)
- 1 - Low power (airplane) mode
- 2 - Factory test mode
- 3 - Offline
- 4 - Resetting
- 5 - Power off
- 6 - Persistent low power (airplane) mode
- 7 - Mode - only low power

Valid transitions for Power Modes

- Online to Low Power, Persistent low power, Factory test, Offline or Shut Down
- Low power to online, Persistent low power, Offline, or Shut Down
- Persistent low power to Online, Low power, Offline or Shut down
- Factory test to online
- Offline to Reset

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.20 qaGobiApiTableRadioInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

### 9.20.1 Detailed Description

Network Access Service API Radio Interfaces table.

### 9.20.2 Radio interface

#### 9.20.2.1 Technology (Value - Radio Interface Technology)

- 0 - No service
- 1 - CDMA 1xRTT
- 2 - CDMA 1xEV-DO
- 3 - AMPS (Unsupported)
- 4 - GSM
- 5 - UMTS
- 6 - WLAN
- 7 - GPS
- 8 - LTE
- 9 - TD-SCDMA
- 12 - NR5G

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.21 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

### 9.21.1 Detailed Description

Region Codes table.

### 9.21.2 Region Codes (Code - Region)

- 0 - North America
- 1 - Latin America
- 2 - Europe
- 3 - Asia
- 4 - Australia
- 5 - Global
- 0xFFFFFFFF - Unknown

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.22 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

### 9.22.1 Detailed Description

Voice Service Options.

### 9.22.2 Service Option codes (Code - Reason)

#### 9.22.2.1 Description

- 0x0001 - Basic variable rate voice service (8 kbps)
- 0x0002 - Mobile station loopback (8 kbps)
- 0x0003 - Enhanced variable rate voice service (8 kbps)
- 0x0004 - Asynchronous data service (9.6 kbps)
- 0x0005 - Group 3 facsimile (9.6 kbps)
- 0x0006 - Short message service (rate set 1)
- 0x0007 - Packet data service: Internet or ISO Protocol stack (9.6 kbps)
- 0x0008 - Packet data service: CDPD Protocol stack (9.6 kbps)
- 0x0009 - Mobile station loopback (13 kbps)
- 0x000A - transparent service
- 0x000B - III nontransparent service
- 0x000C - Asynchronous data service (14.4 or 9.6 kbps)
- 0x000D - Group 3 facsimile (14.4 or 9.6 kbps)
- 0x000E - Short message service (rate set 2)
- 0x000F - Packet data service: Internet or ISO Protocol stack (14.4 kbps)
- 0x0010 - Packet data service: CDPD Protocol stack (14.4 kbps)
- 0x0011 - High-rate voice service (13 kbps)
- 0x0012 - Over-the-air parameter administration (rate set 1)
- 0x0013 - Over-the-air parameter administration (rate set 2)
- 0x0014 - Group 3 analog facsimile (rate set 1)
- 0x0015 - Group 3 analog facsimile (rate set 2)
- 0x0016 - High-speed packet data service: Internet or ISO Protocol stack (RS1 forward, RS1 reverse)
- 0x0017 - High-speed packet data service: Internet or ISO Protocol stack (RS1 forward, RS2 reverse)
- 0x0018 - High-speed packet data service: Internet or ISO Protocol stack (RS2 forward, RS1 reverse)
- 0x0019 - High-speed packet data service: Internet or ISO Protocol stack (RS2 forward, RS2 reverse)
- 0x001A - High-speed packet data service: CDPD Protocol stack (RS1 forward, RS1 reverse)
- 0x001B - High-speed packet data service: CDPD Protocol stack (RS1 forward, RS2 reverse)

- 0x001C - High-speed packet data service: CDPD Protocol stack (RS2 forward, RS1 reverse)
- 0x001D - High-speed packet data service: CDPD Protocol stack (RS2 forward, RS2 reverse)
- 0x001E - RATE_SET_1 Supplemental channel loopback test for rate set 1
- 0x001F - RATE_SET_2 Supplemental channel loopback test for rate set 2
- 0x0020 - Test Data Service Option (TDSO)
- 0x0021 - cdma2000 high-speed packet data service, Internet or ISO Protocol stack
- 0x0022 - cdma2000 high-speed packet data service, CDPD Protocol
- 0x0023 - Location services, rate set 1 (9.6 kbps)
- 0x0024 - Location services, rate set 2 (14.4 kbps)
- 0x0025 - ISDN interworking service (64 kbps)
- 0x0026 - GSM voice
- 0x0027 - GSM circuit data
- 0x0028 - GSM packet data
- 0x0029 - GSM short message service
- 0x0036 - Markov Service Option (MSO)
- 0x0037 - Loopback Service Option (LSO)
- 0x0038 - Selectable mode vocoder
- 0x0039 - 32 kbps circuit video conferencing
- 0x003A - CONFERENCING 64 kbps circuit video conferencing
- 0x003B - HRPD packet data service, which when used in paging over the 1X air interface, a page response is not required
- 0x003C - Link Layer Assisted Robust Header Compression (LLA ROHC) - header removal
- 0x003D - LLA ROHC - Header Compression
- 0x003E - Source-controlled Variable-Rate Multimode Wideband (VMR-WB) speech codec rate set 2
- 0x003F - Source-controlled VMR-WB speech codec rate set 1
- 0x0040 - HRPD auxiliary packet data service instance
- 0x0041 - cdma2000/GPRS interworking
- 0x0042 - ISO_PROTOCOL_SO_66 cdma2000 high-speed packet data service, Internet or ISO Protocol stack
- 0x0043 - HRPD packet data IP service where higher layer protocol is IP or ROHC
- 0x0044 - Enhanced variable rate voice service (EVRC-B)
- 0x0045 - HRPD packet data service, which when used in paging over the 1X air interface, a page response is required
- 0x0046 - Enhanced variable rate voice service (EVRC-WB)
- 0x1004 - Asynchronous data service, Revision 1 (9.6 or 14.4 kbps)
- 0x1005 - Group 3 facsimile, Revision 1 (9.6 or 14.4 kbps)
- 0x1007 - Packet data service: Internet or ISO Protocol stack, Revision 1 (9.6 or 14.4 kbps)
- 0x1008 - Packet data service: CDPD Protocol stack, Revision 1 (9.6 or 14.4 kbps)

- 0x7FF8 - Identifies service reference identifier 0
- 0x7FF9 - Identifies service reference identifier 1
- 0x7FFA - Identifies service reference identifier 2
- 0x7FFB - Identifies service reference identifier 3
- 0x7FFC - Identifies service reference identifier 4
- 0x7FFD - Identifies service reference identifier 5
- 0x7FFE - Identifies service reference identifier 6
- 0x7FFF - Identifies service reference identifier 7

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.23 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

### 9.23.1 Detailed Description

Voice Supplementary Service Information Classes.

### 9.23.2 Supplementary Service Information Classes (Value - Service Class)

- 0X00 - CLASS_NONE
- 0X01 - CLASS_VOICE
- 0X02 - CLASS_DATA
- 0X04 - CLASS_FAX
- 0X08 - CLASS_SMS
- 0X10 - CLASS_DATACIRCUITSYNC
- 0X20 - CLASS_DATACIRCUITASYNC
- 0X40 - CLASS_PACKETACCESS
- 0X80 - CLASS_PADACCESS

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.24 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

### 9.24.1 Detailed Description

Swi Audio related tables.

### 9.24.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

### 9.24.3 Physical Interface (Device ID - description - Interface parameters)

- 0 - PCM - Mode: 0-slave, 1-master, 2-Auxiliary PCM; Rate: 0-8k, 1-16k; Format: 0-linear, 1-u-law, 2-A-law; Padding: 0-disable, 1-enable; Bits-frame: 0-8BPF, 1-16BPF, 2-32BPF, 3-64BPF, 4-128BPF, 5-256BPF;
- 1 - I2S - None
- 2 - Analog(with internal codec) - None
- 3 - USB - None

Copyright: © 2013 Sierra Wireless, Inc. all rights reserved

## 9.25 qaGobiApiTableSwiOMADMSessionStatus.h File Reference

Session Status table.

### 9.25.1 Detailed Description

Session Status table.

### 9.25.2 OMA DM Session Status (Session Status - Meaning - Usage)

- 0x0000 - Successful - Session succeeded
- 0x0001 - Break - Session succeeded
- 0x0002 - Large Object Handled - Session succeeded
- 0x0003 - No status - Session succeeded
- 0x0004 - No more commands - Session succeeded
- 0x0005 - User cancel - Session cancelled
- 0x0100-0x01FF - General errors
- 0x0200-0x02FF - Syncml errors
- 0x0300-0x03FF - Authentication errors
- 0x0400-0x04FF - Protocol errors
- 0x0500-0x05FF - Tree errors (DM Only)
- 0x0600-0x06FF - Not applicable
- 0x0700-0x07FF - Trigger errors

- 0x0800-0x08FF - FUMO errors
- 0x0900-0x09FF - Communication errors
- 0x0A00-0x0AFF - Parsing errors
- 0x0B00-0x0CFF - Not applicable
- 0x7F00-0x7F12 - Insignia errors
- 0x7F13 - Illegal text - Text received contains illegal characters
- 0x7F14 - Download failure - Failed to download FOTA image
- 0x7F15 - Empty session - Session ran successfully, but no information was updated
- 0x7F16 - Factory reset successful - Factory reset succeeded
- 0x7F17 - Factory reset fail - Factory reset failed

Copyright: © 2017 Sierra Wireless, Inc. all rights reserved

## 9.26 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

### 9.26.1 Detailed Description

Update Complete Status table.

### 9.26.2 OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)

- 200 - Successful - The request has succeeded
- 250-299 - Successful(vendor specified) - successful operation with vendor specified ResultCode
- 400 - Management Client Error - Management Client error - based on User or Device behavior
- 401 - User Cancelled - User chose not to accept the operation when prompted
- 402 - Corrupted Firmware Update Package - Corrupted firmware update package did not store correctly. Detected for example, by mismatch CRCs between actual and expected
- 403 - Firmware UpdatePackage( Device Mismatch ) - Wrong firmware update package delivered to device based on current device characteristics
- 404 - Failed Firmware Update Package Validation - Failure to positively validate digital signature of firmware update package
- 405 - Firmware Update Package Not acceptable - firmware update package is not acceptable
- 406 - Alternate Download Authentication Failure - authentication required but authentication failure was encountered when downloading firmware update package
- 407 - Alternate Download Request Timeout - client has encountered a timeout when downloading firmware update package
- 408 - Not Implemented - the device does not support the requested operation
- 409 - Undefined Error - indicates failure not defined by any other error code
- 410 - Firmware Update Failed - firmware update operation failed in device

- 411 - Malformed or Bad URL - the URL provided for alternate download is bad
- 412 - Alternate Download Server Unavailable - the alternate download server is unavailable or does not respond
- 450 - Client Error ( OMADM General ) - Vendor defined client error
- 451 - Client Error ( OMADM SyncML ) - Vendor defined client error
- 452 - Client Error ( OMADM Auth ) - Vendor defined client error
- 453 - Client Error ( OMADM Protocol )- Vendor defined client error
- 454 - Client Error ( OMADM Tree ) - Vendor defined client error
- 455 - Client Error ( OMADM DStore ) - Vendor defined client error
- 456 - Client Error ( OMADM Trigger ) - Vendor defined client error
- 457 - Client Error ( OMADM Fumo ) - Vendor defined client error
- 458 - Client Error ( OMADM Comms ) - Vendor defined client error
- 459 - Client Error ( OMADM Parse ) - Vendor defined client error
- 460 - Client Error ( OMADM 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

## 9.27 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

### 9.27.1 Detailed Description

Voice Service Call and supplementary services end reasons.

### 9.27.2 Voice Call and supplementary services end reason codes (Code - Reason)

#### 9.27.2.1 General

- 0 - Phone is offline
- 20 - Phone is CDMA locked until a power cycle; CDMA only
- 21 - Phone has no service, this is for backward compatibility
- 22 - Call has ended abnormally; CDMA only
- 23 - Received intercept from the base station; originating only; CDMA only
- 24 - Received reorder from the base station; originating only; CDMA only
- 25 - Received release from the base station; no reason was given
- 26 - Received release from the base station; SO reject; CDMA only
- 27 - Received incoming call from the base station
- 28 - Received alert stop from the base station; incoming only; CDMA only
- 29 - Client ended the call
- 30 - Received end activation; OTASP call only; CDMA only
- 31 - MC aborted the origination/conversation; CDMA only
- 32 - Maximum access probes were transmitted; CDMA only
- 33 - Persistence test failure; FEATURE_JCDMA only; CDMA only
- 34 - R-UIM is not present
- 35 - Access attempt is already in progress
- 36 - Access failure for a reason other than the above
- 37 - Received retry order; originating only; IS 2000; CDMA only
- 38 - BYBS Concurrent service is not supported by the base station
- 39 - No response was received from the base station
- 40 - Call was rejected by the base station; CDMA only
- 41 - Concurrent services requested were not compatible; CDMA only
- 42 - Access is blocked by the base station; CDMA only
- 43 - Corresponds to CM_CALL_ORIG_ ERR_ALREADY_IN_TC
- 44 - Call is ended because an emergency call is flashed over this call; CDMA only
- 45 - Used if CM is ending a GPS call in preference of a user call
- 46 - Used if CM is ending an SMS call in preference of a user call
- 47 - Used if CM is ending a data call in preference of an emergency call
- 48 - Call was rejected because of a redirection or handoff
- 49 - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only
- 50 - To support OTASP SPC Error indication
- 51 - Maximum access probes for an IS-707B call; CDMA only

- 52 - Base station reject order
- 53 - Base station retry order
- 54 - Timer T42 is expired
- 55 - Timer T40 is expired
- 56 - Service initialization failure - Traffic Channel Initialization
- 57 - Timer T50m is expired - Traffic Channel Initialization
- 58 - Timer T51m is expired - Traffic Channel Initialization
- 59 - Acknowledgement timeout due to 12 retransmissions
- 60 - Bad forward link or timer T5M is expired
- 61 - Transceiver Resource Manager request failed
- 62 - Timer T41 is expired
- 100 - WCDMA/GSM/TDS only; call end LL cause, Received a reason for ending the call from the lower layer
- 101 - WCDMA/GSM only; Call origination request failed
- 102 - WCDMA/GSM only; client rejected an incoming call
- 103 - WCDMA/GSM only; client rejected a setup indication
- 104 - WCDMA/GSM only; network ended the call
- 105 - WCDMA/GSM only
- 106 - GWM/WCDMA only; phone has no service
- 107 - 1X only; phone has no service
- 108 - Full service is unavailable
- 109 - Indicates resources are not available to handle a new MO/MT PS call

#### 9.27.2.2 service Errors

- 110 - Unknown subscriber
- 111 - Illegal subscriber
- 112 - Bearer service not provisioned
- 113 - Tele service not provisioned
- 114 - Illegal equipment
- 115 - Call barred
- 116 - Illegal ss operation
- 117 - Ss error status
- 118 - Ss not available
- 119 - Ss subscription violation
- 120 - Ss incompatibility
- 121 - Facility not supported
- 122 - Absent subscriber

- 123 - Short term denial
- 124 - Long term denial
- 125 - System failure
- 126 - Data missing
- 127 - Unexpected data value
- 128 - Pwd registration failure
- 129 - Negative pwd check
- 130 - Num of pwd attempts violation
- 131 - Position method failure
- 132 - Unknown alphabet
- 133 - Ussd busy
- 134 - Rejected by user
- 135 - Rejected by network
- 136 - Deflection to served subscriber
- 137 - Special service code
- 138 - Invalid deflected to number
- 139 - Mpty participants exceeded
- 140 - Resources not available

#### 9.27.2.3 control cause values

- 141 - Unassigned number
- 142 - No route to destination
- 143 - Channel unacceptable
- 144 - Operator determined barring
- 145 - Normal call clearing
- 146 - User busy sEE [s3, aNNEX h]
- 147 - No user responding sEE [s3, aNNEX h]
- 148 - User alerting no answer
- 149 - Call rejected sEE [s3, aNNEX h]
- 150 - Number changed sEE [s3, aNNEX h]
- 151 - Preemption sEE [s3, aNNEX h]
- 152 - Destination out of order
- 153 - Invalid number format
- 154 - Facility rejected
- 155 - Resp to status enquiry
- 156 - Normal unspecified

- 157 - No circuit or channel available
- 158 - Network out of order
- 159 - Temporary failure
- 160 - Switching equipment congestion
- 161 - Access information discarded
- 162 - Requested circuit or channel not available
- 163 - Resources unavailable or unspecified
- 164 - Qos unavailable
- 165 - Requested facility not subscribed
- 166 - Incoming calls barred within cug
- 167 - Bearer capability not auth
- 168 - Bearer capability unavailable
- 169 - Service option not available
- 170 - Acm 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

#### 9.27.2.4 reject causes

- 197 - Imsi unknown in hlr
- 198 - Illegal ms
- 199 - Imsi unknown in vlr
- 200 - Imei not accepted
- 201 - Illegal me sEE
- 202 - Plmn not allowed
- 203 - Location area not allowed
- 204 - Roaming not allowed in this location area
- 205 - No suitable cells in location area
- 206 - Network failure sEE
- 207 - mac failure sEE
- 208 - Synch failure
- 209 - Network congestion
- 210 - GSM authentication unacceptable
- 211 - Service not subscribed
- 212 - Service temporarily out of order
- 213 - Call cannot be identified
- 214 - Incorrect semantics in message
- 215 - Mandatory information invalid
- 216 - Call failed due to other access stratum failures
- 217 - SIM is invalid
- 218 - Invalid call state
- 219 - Access class is blocked
- 220 - No resources are in the protocol stack to allow the call
- 221 - Invalid user data was received

#### 9.27.2.5 reject causes

- 222 - Timer T3230 is expired
- 223 - No cell is available
- 224 - Abort message was received
- 225 - Radio link was lost due to other lower layer causes

#### 9.27.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

#### 9.27.2.7 stratum reject causes

- 228 - Access stratum RR release indication
- 229 - Access stratum random access failure
- 230 - RRC_REL_IND Access stratum RRC release indication
- 231 - Access stratum close session indication
- 232 - Access stratum open session failure
- 233 - Access stratum low level failure
- 234 - Access stratum low level failure redial is not allowed
- 235 - Access stratum low level immediate retry
- 236 - Access stratum abort radio is unavailable

#### 9.27.2.8 reject causes

- 237 - Service option is not supported

#### 9.27.2.9 IP end reasons

- 300 - Received SIP 400 bad request;waiting for INVITE response
- 301 - Received SIP 400 bad request;waiting for INVITE response
- 302 - Received SIP 404 not found; call failed; called party does not exist
- 303 - Received SIP 415 unsupported media type; call failed; called party does not support media
- 304 - Received SIP 480 temporarily unavailable; call failed; called party is not in the LTE area
- 305 - No network response; call failed
- 306 - No network response; unable to put call on hold
- 307 - Moved to eHRPD; call failed or dropped; not in the LTE area
- 308 - Upgrade/downgrade rejected (200 OK with the current call SDP)
- 309 - Received 403 call forbidden; waiting for INVITE response
- 310 - Generic timeout; did not receive a response from the server or other end
- 311 - Reported on the MO side for generic internal software errors; user can try again if the call still exists
- 312 - Reported on the MT side if the upgrade timer has been cancelled or cannot complete the request for some reason after notifying the user of a re-invite request
- 313 - Call origination is rejected due to a Service-Specific Access Control (SSAC) barring
- 314 - Phone was put in thermal emergency
- 315 - 1XCSFB call ended because of a soft failure
- 316 - 1XCSFB call ended because of a hard failure

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.28 qmerrno.h File Reference

## Enumerations

- enum eQCWWANError {
  eQCWWAN_ERR_ENUM_BEGIN = -1,
  eQCWWAN_ERR_NONE,
  eQCWWAN_ERR_GENERAL,
  eQCWWAN_ERR_INTERNAL,
  eQCWWAN_ERR_MEMORY,
  eQCWWAN_ERR_INVALID_ARG,
  eQCWWAN_ERR_BUFFER_SZ,
  eQCWWAN_ERR_NO_DEVICE,
  eQCWWAN_ERR_INVALID_DEVID,
  eQCWWAN_ERR_NO_CONNECTION,
  eQCWWAN_ERR_QMI_IFACE,
  eQCWWAN_ERR_QMI_CONNECT,
  eQCWWAN_ERR_QMI_REQ_SCH,
  eQCWWAN_ERR_QMI_REQ,
  eQCWWAN_ERR_QMI_RSP,
  eQCWWAN_ERR_QMI_REQ_TO,
  eQCWWAN_ERR_QMI_RSP_TO,
  eQCWWAN_ERR_MALFORMED_QMI_RSP,
  eQCWWAN_ERR_INVALID_QMI_RSP,
  eQCWWAN_ERR_INVALID_FILE,
  eQCWWAN_ERR_FILE_OPEN,
  eQCWWAN_ERR_FILE_COPY,
  eQCWWAN_ERR_OFFLINE = 27,
  eQCWWAN_ERR_RESET,
  eQCWWAN_ERR_NO_SIGNAL,
  eQCWWAN_ERR_MULTIPLE_DEVICES,
  eQCWWAN_ERR_DRIVER,
  eQCWWAN_ERR_NO_CANCELABLE_OP,
  eQCWWAN_ERR_CANCEL_OP,
  eQCWWAN_ERR_API_MUTEX_TIMEOUT,
  eQCWWAN_ERR_PDU_GENERATION,
  eQCWWAN_ERR_INVALID_XID,
  eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED,
  eQCWWAN_ERR_ENUM_END,
  eQCWWAN_ERR_QMI_OFFSET = 1000,
  eQCWWAN_ERR_QMI_MALFORMED_MSG = 1001,
  eQCWWAN_ERR_QMI_NO_MEMORY,
  eQCWWAN_ERR_QMI_INTERNAL,
  eQCWWAN_ERR_QMI_ABORTED,
  eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED,
  eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION,
  eQCWWAN_ERR_QMI_INVALID_CLIENT_ID,
  eQCWWAN_ERR_QMI_NO_THRESHOLDS,
  eQCWWAN_ERR_QMI_INVALID_HANDLE,
  eQCWWAN_ERR_QMI_INVALID_PROFILE,
  eQCWWAN_ERR_QMI_INVALID_PINID,
  eQCWWAN_ERR_QMI_INCORRECT_PIN,
  eQCWWAN_ERR_QMI_NO_NETWORK_FOUND,
  eQCWWAN_ERR_QMI_CALL_FAILED,
  eQCWWAN_ERR_QMI_OUT_OF_CALL,
  eQCWWAN_ERR_QMI_NOT_PROVISIONED,
  eQCWWAN_ERR_QMI_MISSING_ARG,
  eQCWWAN_ERR_QMI_ARG_TOO_LONG = 1019,
  eQCWWAN_ERR_QMI_INVALID_TX_ID = 1022,
  eQCWWAN_ERR_QMI_DEVICE_IN_USE,
  eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED,
  eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED,
  eQCWWAN_ERR_QMI_NO_EFFECT,
  eQCWWAN_ERR_QMI_NO_FREE_PROFILE,
  eQCWWAN_ERR_QMI_INVALID_PDP_TYPE,
  eQCWWAN_ERR_QMI_INVALID_TECH_PREF,
  eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE,

eQCWWAN_ERR_QMI_WIDTH = 0xFFFF }

- enum qm_wds_ds_profile_extended_err_codes {
  eWDS_ERR_PROFILE_REG_RESULT_FAIL = 1,
  eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL,
  eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP,
  eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE,
  eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM,
  eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT,
  eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL,
  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_INVAL_SUBS_ID,
  eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY,
  eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY = 1001,
  eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR,
  eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED,
  eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET,
  eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET,
  eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES,
  eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE = 1101,
  eWDS_ERR_PROFILE_REG_END }

## 9.28.1 Enumeration Type Documentation

### 9.28.1.1 enum eQCWWANError

**QMI Error Code Enumeration**

**Enumerator**

***eQCWWAN_ERR_ENUM_BEGIN***

***eQCWWAN_ERR_NONE*** 00 - Success

***eQCWWAN_ERR_GENERAL*** 01 - General error

***eQCWWAN_ERR_INTERNAL*** 02 - Internal error

***eQCWWAN_ERR_MEMORY*** 03 - Memory error

***eQCWWAN_ERR_INVALID_ARG*** 04 - Invalid argument

***eQCWWAN_ERR_BUFFER_SZ*** 05 - Buffer too small

***eQCWWAN_ERR_NO_DEVICE*** 06 - Unable to detect WWAN device

***eQCWWAN_ERR_INVALID_DEVID*** 07 - Invalid WWAN device ID

***eQCWWAN_ERR_NO_CONNECTION*** 08 - No connection to WWAN device

***eQCWWAN_ERR_QMI_IFACE*** 09 - Unable to obtain QMI interace

***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 intercace is not muticast

***eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE*** 1062 - Maximum requests in use

***eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE*** 1063 - Invalid muticast handle

***eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF*** 1064 - Invalid IP family preference

***eQCWWAN_ERR_QMI_SESSION_INACTIVE*** 1065 - No tracking session has been started

***eQCWWAN_ERR_QMI_SESSION_INVALID*** 1066 - Current session does not allow this operation

***eQCWWAN_ERR_QMI_SESSION_OWNERSHIP*** 1067 - Current tracking session not started by this QMI control point

***eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES*** 1068 - Device GPS service resources insufficient for request

***eQCWWAN_ERR_QMI_DISABLED*** 1069 - Support for QMI message is disabled

***eQCWWAN_ERR_QMI_INVALID_OPERATION*** 1070 - Invalid operation specified

***eQCWWAN_ERR_QMI_INVALID_QMI_CMD*** 1071 - Invalid/unknown QMI command specified

***eQCWWAN_ERR_QMI_TPDU_TYPE*** 1072 - Message contains TPDU type that cannot be read as raw message

***eQCWWAN_ERR_QMI_SMSC_ADDR*** 1073 - The SMSC address specified is invalid

***eQCWWAN_ERR_QMI_INFO_UNAVAILABLE*** 1074 - Information element is unavailable at this point

***eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG*** 1075 - Segment size too large

***eQCWWAN_ERR_QMI_SEGMENT_ORDER*** 1076 - Segment order is incorrect

***eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED*** 1077 - Bundling not supported

***eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE*** 1078 - The operation failed partially

***eQCWWAN_ERR_QMI_POLICY_MISMATCH*** 1079 - Policy mismatch

***eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND*** 1080 - SIM file not found

***eQCWWAN_ERR_QMI_EXTENDED_INTERNAL*** 1081 - Extended internal error

***eQCWWAN_ERR_QMI_ACCESS_DENIED*** 1082 - Access to a required entity is not available

***eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED*** 1083 - Selected operating mode is invalid with current hardware setting

***eQCWWAN_ERR_QMI_ACK_NOT_SENT*** 1084 - ACK not sent

***eQCWWAN_ERR_QMI_INJECT_TIMEOUT*** 1084 - Inject a timeout for the request

***eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE*** 1090 - Incompatible state

***eQCWWAN_ERR_QMI_FDN_RESTRICT*** 1091 - FDN Restrict

***eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE*** 1092 - SUPS failure cause

***eQCWWAN_ERR_QMI_NO_RADIO*** 1093 - No Radio

***eQCWWAN_ERR_QMI_NOT_SUPPORTED*** 1094 - Not Supported

***eQCWWAN_ERR_QMI_NO_SUBSCRIPTION*** 1095 - No Subscription

***eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED*** 1096 - Card call control failed

***eQCWWAN_ERR_QMI_NETWORK_ABORTED*** 1097 - Network Aborted

***eQCWWAN_ERR_QMI_MSG_BLOCKED*** 1098 - Open Error

***eQCWWAN_ERR_QMI_MAX*** Error - End of QMI specific defines

***eQCWWAN_ERR_SWICM_START*** **Vendor defines - Connection Manager error codes**

***eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED*** 0xE001 - The API is yet to be implemented

***eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED*** 0xE002 - The service is not supported

***eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED*** 0xE003 - The client is not supported

***eQCWWAN_ERR_SWICM_TIMEOUT*** 0xE004 - API Timeout

***eQCWWAN_ERR_SWICM_SOCKET_IN_USE*** 0xE005 - The communication socket is in use

***eQCWWAN_ERR_SWICM_AM_VERS_ERROR*** 0xE006 - SLQS API and SDK version mismatch

***eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS*** 0xE007 - Failed to kill SDK process

***eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS*** 0xE008 - Call in progress

***eQCWWAN_ERR_SWICM_V4DWN_V6DWN*** 0xE009 - IPV4 and IPV6 is down

***eQCWWAN_ERR_SWICM_V4DWN_V6UP*** 0xE00A - IPV4 is down and IPV6 is up

***eQCWWAN_ERR_SWICM_V4UP_V6DWN*** 0xE00B - IPV4 is up and IPV6 is down

***eQCWWAN_ERR_SWICM_V4UP_V6UP*** 0xE00C - IPV4 and IPV6 is up

***eQCWWAN_ERR_SWICM_INVALID_SESSION_ID*** 0xE00D - Invalid V4 Session ID

***eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID*** 0xE00E - Invalid V4 Session ID

***eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID*** 0xE00F - Invalid V6 Session ID

***eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS*** 0xE010 - No available Session Manager slots for additional data sessions

***eQCWWAN_ERR_SWICM_END*** 0xE011 - End of connection manager specific codes

***eQCWWAN_ERR_SWISMS_START*** **Vendor defines - SMS Error codes**

***eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG*** 0xE101 - SMS message length is long

***eQCWWAN_ERR_SWISMS_MSG_CORRUPTED*** 0xE102 - The SMS message is corrupted (encoding wrong)

***eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED*** 0xE103 - The SMS number is corrupted (incorrect number)

***eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND*** 0xE104 - The SMS bearer data is not available

***eQCWWAN_ERR_SWISM_END***

***eQCWWAN_ERR_SWIIM_START*** **Vendor defines - Image Management error codes**

***eQCWWAN_ERR_SWIIM_INVALID_PATH*** 0xE801 - Invalid directory path

***eQCWWAN_ERR_SWIIM_OPENING_DIR*** 0xE802 - Unable to open the directory

***eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND*** 0xE803 - No Firmware image present in the path

***eQCWWAN_ERR_SWIIM_OPENING_FILE*** 0xE804 - Unable to open the file

***eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE*** 0xE805 - Firmware image is corrupted

***eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED*** 0xE806 - No Firmware image download needed

***eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL*** 0xE807 - Firmware update failed

***eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH*** 0xE808 - Update success but pri/fw preference mismatch

***eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS*** 0xE809 - Update successful

***eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE*** 0xE80A - Enter Download Mode

***eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE*** 0xE80B - File transfer to modem complete

***eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT*** 0xE80C - Wait for modem to reboot

***eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE*** 0xE80D - Invalid Crash State for Firmware Download

***eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE*** 0xE80E - Same as current active image

***eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX*** 0xE80F - invalid slot index

***eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_NOT_SIGNED***

***eQCWWAN_ERR_SWIIM_END***

***eQCWWAN_ERR_SWIDCS_START*** **Vendor defines - Device Connectivity error codes**

***eQCWWAN_ERR_SWIDCS_IOCTL_ERR*** 0xE901 - IO Control error

***eQCWWAN_ERR_SWIDCS_FILEIO_ERR*** 0xE902 - file open/read/write error

***eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND*** 0xE903 - The device is not found

***eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED*** 0xE904 - Application is disconnected from SDK

***eQCWWAN_ERR_SWIDCS_END***

***eQCWWAN_ERR_QMI_CAT_START*** **QMI errors related to CAT**

***eQCWWAN_ERR_QMI_EVENT_REG_FAILED*** 62441 - CAT event registration failed

***eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP*** 62442 - Invalid terminal response

***eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD*** 62443 - Invalid envelope command

***eQCWWAN_ERR_QMI_CARD_BUSY_RSP*** 62444 - Card busy response for envelope command

***eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE*** 62445 - Envelope command failure

***eQCWWAN_ERR_QMI_CAT_END***

***eQCWWAN_ERR_NULL_TLV***

***eQCWWAN_ERR_QMI_WIDTH*** 0xFFFF - Not an error, represent the end of QMI errors

#### 9.28.1.2 enum qm_wds_ds_profile_extended_err_codes

**WDS DS profile extented error codes**

**Enumerator**

***eWDS_ERR_PROFILE_REG_RESULT_FAIL*** 1 - General Failure

***eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL*** 2 - The request contains an invalid profile handle

***eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP*** 3 - An invalid operation was requested.

***eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE*** 4 - The request contains an invalid technology type

***eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM*** 5 - The request contains an invalid profile number

***eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT*** 6 - The request contains an invalid profile identifier

***eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL*** 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_INVAL_SUBS_ID*** 11 - The request contains an invalid subscrition identifier.

***eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY*** 12 - The request contains an invalid profile family.

***eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY*** 1001 - The request contains an invalid 3GPP profile family.

***eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR*** 1002 - An error was encountered while accessing the 3GPP profiles.

***eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED*** 1003 - The given 3GPP profile doesn't have a valid context.

***eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET*** 1004 - The given 3GPP profile is marked invalid.

***eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET*** 1005 - The given 3GPP profile is marked read-only.

***eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES*** 1006 - Creation of a new 3GPP profile failed because the limit of 16 profiles has already been reached.

***eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE*** 1101 - An invalid profile identifier was received as part of the 3GPP2 profile modification request.

***eWDS_ERR_PROFILE_REG_END***

## 9.29 qos.h File Reference

### Data Structures

- struct unpack_qos_SLQSQosGetNetworkStatus_t
- struct pack_qos_SLQSQosSwiReadApnExtraParams_t
- struct unpack_qos_SLQSQosSwiReadApnExtraParams_t

- struct pack_qos_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_SLQSSetQosEventCallback_t
- struct unpack_qos_QosFlowInfoState_t
- struct unpack_qos_dataRate_t
- struct unpack_qos_tokenBucket_t
- struct unpack_qos_pktErrRate_t
- struct unpack_qos_swiQosFlow_t
- struct unpack_qos_IPv4Addr_t
- struct unpack_qos_Tos_t
- struct unpack_qos_IPv6Addr_t
- struct unpack_qos_IPv6TrafCls_t
- struct unpack_qos_Port_t
- struct unpack_qos_swiQosFilter_t
- struct unpack_qos_QosFlowInfo_t
- struct unpack_qos_SLQSSetQosEventCallback_ind_t
- struct qos_BindDataPortPeripheralEndPointID_t
- struct qos_BindDataPortMuxID_t
- struct qos_BindDataPortSIODataPort_t
- struct pack_qos_BindDataPort_t
- struct unpack_qos_BindDataPort_t

## Macros

- #define LITEQMI_MAX_QOS_FLOW_PER_APN_STATS 10
- #define LITEQMI_MAX_QOS_FILTERS 25
- #define LITEQMI_MAX_QOS_FLOWS 8

## Functions

- int pack_qos_SLQSQosGetNetworkStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_qos_SLQSQosGetNetworkStatus (uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSQosGetNetworkStatus_t *pOutput)
- int pack_qos_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, unpack_qos_SLQSSetQosEventCallback_t *pOutput)
- int unpack_qos_SLQSSetQosEventCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_qos_SLQSSetQosEventCallback_ind_t *pOutput)
- int pack_qos_BindDataPort (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_qos_BindDataPort_t req)
- int unpack_qos_BindDataPort (uint8_t *pResp, uint16_t respLen, unpack_qos_BindDataPort_t *pOutput)

## 9.29.1 Macro Definition Documentation

### 9.29.1.1 #define LITEQMI_MAX_QOS_FILTERS 25

### 9.29.1.2 #define LITEQMI_MAX_QOS_FLOW_PER_APN_STATS 10

### 9.29.1.3 #define LITEQMI_MAX_QOS_FLOWS 8

## 9.29.2 Function Documentation

### 9.29.2.1 int pack_qos_BindDataPort ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_qos_BindDataPort_t *req* )

Function to pack command to Binds a control point to a data port. This maps to PkQmiQosBindDataPort

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *req* | • See pack_qos_BindDataPort_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Timeout: 2 seconds

### 9.29.2.2 int pack_qos_SLQSQosGetNetworkStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Function to pack command to retrieve QoS status of the network. This maps to SLQSQosGetNetworkStatus

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Timeout: 2 seconds
- Technology Supported: CDMA
- PDN Specific: No

### 9.29.2.3 int pack_qos_SLQSQosSwiReadApnExtraParams ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_qos_SLQSQosSwiReadApnExtraParams_t *reqParam* )

Function to pack QMI command to query extra APN parameters This maps to SLQSQosSwiReadApnExtraParams

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_qos_SLQSQosSwiReadApnExtraParams_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Timeout: 2 seconds
  - PDN Specific: Yes

#### 9.29.2.4 int pack_qos_SLQSQosSwiReadDataStats ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_qos_SLQSQosSwiReadDataStats_t *reqParam* )

Function to pack QMI command to query APN data statistics This maps to SLQSQosSwiReadDataStats

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_qos_SLQSQosSwiReadDataStats_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Timeout: 2 seconds
  - PDN Specific: Yes

#### 9.29.2.5 int pack_qos_SLQSSetQosEventCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_qos_SLQSSetQosEventCallback_t *reqParam* )

Function to pack QMI command to enable QoS event indications This maps to SLQSSetQosEventCallback

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |

| | | |
|---|---|---|
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_qos_SLQSSetQosEventCallback_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Timeout: 2 seconds
  - PDN Specific: Yes

#### 9.29.2.6 int unpack_qos_BindDataPort ( uint8_t * *pResp,* uint16_t *respLen,* **unpack_qos_BindDataPort_t** * *pOutput* )

Function to unpack the response to Binds a control point to a data port. This maps to UpkQmiQosBindDataPort

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.29.2.7 int unpack_qos_SLQSQosGetNetworkStatus ( uint8_t * *pResp,* uint16_t *respLen,* **unpack_qos_SLQSQosGetNetworkStatus_t** * *pOutput* )

Function to unpack the response to get NW QoS status command This maps to SLQSQosGetNetworkStatus

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSQosGetNetworkStatus_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.29.2.8 int unpack_qos_SLQSQosSwiReadApnExtraParams ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSQosSwiReadApnExtraParams_t * *pOutput* )

Function to unpack the response to get NW QoS status command This maps to SLQSQosSwiReadApnExtraParams

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSQosSwiReadApnExtraParams_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.29.2.9 int unpack_qos_SLQSQosSwiReadDataStats ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSQosSwiReadDataStats_t * *pOutput* )

Function to unpack APN data statistics response This maps to SLQSQosSwiReadDataStats

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |

| | | |
|---|---|---|
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSQosSwiReadDataStats_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.29.2.10 int unpack_qos_SLQSSetQosEventCallback ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosEventCallback_t * *pOutput* )

Function to unpack enable QoS event indications command's response This maps to SLQSSetQosEventCallback

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.29.2.11 int unpack_qos_SLQSSetQosEventCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosEventCallback_ind_t * *pOutput* )

Function to unpack QoS event indications This maps to SLQSSetQosEventCallback

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSSetQosEventCallback_ind_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_NETWORK_STATUS_IND indication to identify this event from QOS service read function

### 9.29.2.12 int unpack_qos_SLQSSetQosNWStatusCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosNWStatusCallback_ind_t * *pOutput* )

Function to unpack QoS NW status indication. This maps to SLQSSetQosNWStatusCallback

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSSetQosNWStatusCallback_ind_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Technology Supported: CDMA
- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_NETWORK_STATUS_IND indication to identify this event from QOS service read function

### 9.29.2.13 int unpack_qos_SLQSSetQosPriEventCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosPriEventCallback_ind_t * *pOutput* )

Function to unpack QoS primary flow events. This maps to SLQSSetQosPriEventCallback

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSSetQosPriEventCallback_ind_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_PRIMARY_QOS_EVENT_IND indication to identify this event from QOS service read function
- This is only generated when the primary flow is modified by the host

#### 9.29.2.14 int unpack_qos_SLQSSetQosStatusCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_qos_SLQSSetQosStatusCallback_ind_t * *pOutput* )

Function to unpack QoS status indications. This maps to SLQSSetQosStatusCallback

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pOutput* | • See unpack_qos_SLQSSetQosStatusCallback_ind_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI_QOS_FLOW_STATUS_IND indication to identify this event from QOS service read function

## 9.30 rms.h File Reference

### Data Structures

- struct unpack_rms_GetSMSWake_t
- struct pack_rms_SetSMSWake_t
- struct unpack_rms_SetSMSWake_t

### Macros

- #define __LITEQMI_RMS_H__

### Functions

- int pack_rms_GetSMSWake (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_rms_GetSMSWake (uint8_t ∗pResp, uint16_t respLen, unpack_rms_GetSMSWake_t ∗pOutput)
- int pack_rms_SetSMSWake (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_rms_SetSMSWake_t ∗reqArg)
- int unpack_rms_SetSMSWake (uint8_t ∗pResp, uint16_t respLen, unpack_rms_SetSMSWake_t ∗pOutput)

### 9.30.1 Macro Definition Documentation

#### 9.30.1.1 #define __LITEQMI_RMS_H__

### 9.30.2 Function Documentation

#### 9.30.2.1 int pack_rms_GetSMSWake ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen* )

Get SMS Wake pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.30.2.2 int pack_rms_SetSMSWake ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_rms_SetSMSWake_t * reqArg )

Set SMS Wake pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.30.2.3 int unpack_rms_GetSMSWake ( uint8_t * pResp, uint16_t respLen, unpack_rms_GetSMSWake_t * pOutput )

Get SMS Wake unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.30.2.4 int unpack_rms_SetSMSWake ( uint8_t * pResp, uint16_t respLen, unpack_rms_SetSMSWake_t * pOutput )

Set SMS Wake unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.31 sar.h File Reference

### Data Structures

- struct unpack_sar_SLQSGetRfSarState_t
- struct pack_sar_SLQSSetRfSarState_t

### Typedefs

- typedef unpack_result_t unpack_sar_SLQSSetRfSarState_t

### Functions

- int pack_sar_SLQSGetRfSarState (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_sar_SLQSGetRfSarState (uint8_t ∗pResp, uint16_t respLen, unpack_sar_SLQSGetRfSarState_t ∗pOutput)
- int pack_sar_SLQSSetRfSarState (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_sar_SLQSSetRfSarState_t ∗reqArg)
- int unpack_sar_SLQSSetRfSarState (uint8_t ∗pResp, uint16_t respLen, unpack_sar_SLQSSetRfSarState_t ∗pOutput)

### 9.31.1 Typedef Documentation

#### 9.31.1.1 typedef unpack_result_t unpack_sar_SLQSSetRfSarState_t

### 9.31.2 Function Documentation

#### 9.31.2.1 int pack_sar_SLQSGetRfSarState ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen* )

Gets the specified RF SAR state pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.31.2.2 int pack_sar_SLQSSetRfSarState ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen*, pack_sar_SLQSSetRfSarState_t ∗ *reqArg* )

Sets the specified RF SAR state pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.31.2.3 int unpack_sar_SLQSGetRfSarState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sar_SLQSGetRfSarState_t * *pOutput* )

Gets the specified RF SAR state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.31.2.4 int unpack_sar_SLQSSetRfSarState ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sar_SLQSSetRfSarState_t * *pOutput* )

Sets the specified RF SAR state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.32 sms.h File Reference

### Data Structures

- struct pack_sms_SLQSGetSMS_t
- struct unpack_sms_SLQSGetSMS_t
- struct pack_sms_SLQSGetSMSList_t

- struct qmiSmsMessageList
- struct unpack_sms_SLQSGetSMSList_t
- struct pack_sms_SLQSModifySMSStatus_t
- struct unpack_sms_SLQSModifySMSStatus_t
- struct pack_sms_SLQSDeleteSMS_t
- struct unpack_sms_SLQSDeleteSMS_t
- struct pack_sms_SendSMS_t
- struct unpack_sms_SendSMS_t
- struct pack_sms_SetNewSMSCallback_t
- struct unpack_sms_SetNewSMSCallback_t
- struct sMSMTMessageInfo
- struct newMTMessageTlv
- struct sMSTransferRouteMTMessageInfo
- struct transferRouteMessageTlv
- struct sMSMessageModeInfo
- struct messageModeTlv
- struct sMSEtwsMessageInfo
- struct sMSEtwsMessageTlv
- struct sMSEtwsPlmnInfo
- struct eTWSPLMNInfoTlv
- struct sMSCAddressInfo
- struct sMSCAddressTlv
- struct sMSOnIMSInfo
- struct sMSOnIMSTlv
- struct unpack_sms_SetNewSMSCallback_ind_t
- struct unpack_sms_SLQSWmsMemoryFullCallBack_ind_t
- struct unpack_sms_GetSMSCAddress_t
- struct pack_sms_SetSMSCAddress_t
- struct unpack_sms_SetSMSCAddress_t
- struct pack_sms_SaveSMS_t
- struct unpack_sms_SaveSMS_t
- struct sms_BroadcastConfig
- struct sms_qaQmi3GPPBroadcastCfgInfo
- struct sms_CDMABroadcastConfig
- struct sms_qaQmi3GPP2BroadcastCfgInfo
- struct unpack_sms_SLQSGetSmsBroadcastConfig_t
- struct pack_sms_SLQSGetSmsBroadcastConfig_t
- struct pack_sms_SLQSSetSmsBroadcastConfig_t
- struct unpack_sms_SLQSSetSmsBroadcastConfig_t
- struct pack_sms_SLQSSetSmsBroadcastActivation_t
- struct unpack_sms_SLQSSetSmsBroadcastActivation_t
- struct sms_transLayerInfo
- struct sms_getTransLayerInfo
- struct unpack_sms_SLQSGetTransLayerInfo_t
- struct sms_getTransNWRegInfo
- struct unpack_sms_SLQSGetTransNWRegInfo_t
- struct sms_getIndicationReg
- struct unpack_sms_SLQSGetIndicationRegister_t
- struct sms_setIndicationReg
- struct pack_sms_SLQSSetIndicationRegister_t
- struct unpack_sms_SLQSSetIndicationRegister_t
- struct sms_routeEntry
- struct sms_setRoutesReq
- struct pack_sms_SLQSSmsSetRoutes_t
- struct unpack_sms_SLQSSmsSetRoutes_t

- struct sms_msgProtocolResp
- struct unpack_sms_SLQSSmsGetMessageProtocol_t
- struct sms_maxStorageSizeReq
- struct sms_maxStorageSizeResp
- struct pack_sms_SLQSSmsGetMaxStorageSize_t
- struct unpack_sms_SLQSSmsGetMaxStorageSize_t
- struct sms_messageWaitingInfoContent
- struct sms_getMsgWaitingInfo
- struct unpack_sms_SLQSGetMessageWaiting_t
- struct sms_sendAsyncsmsParams
- struct pack_sms_SLQSSendAsyncSMS_t
- struct unpack_sms_SLQSSendAsyncSMS_t
- struct pack_sms_SLQSSetSmsStorage_t
- struct unpack_sms_SLQSSetSmsStorage_t
- struct unpack_sms_SLQSSwiGetSMSStorage_t
- struct unpack_sms_SLQSTransLayerInfoCallback_ind_t
- struct unpack_sms_SLQSNWRegInfoCallback_ind_t
- struct unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t
- struct unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t

## Macros

- #define MAX_SMS_MESSAGE_SIZE 2048
- #define MAX_SMS_LIST_SIZE 255
- #define MAX_MS_TRANSFER_ROUTE_MSG 256
- #define MAX_MSE_TWS_MSG 1254
- #define MAX_MSC_ADDRESS_SIZE 256
- #define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255
- #define SMSC_TYPE_LEN 0x03
- #define SMS_CONFIG_LEN 0x05
- #define SMS_MAX_SMS_ROUTES 0x0A
- #define SMS_NUM_OF_SET 0xFF

## Enumerations

- enum eqmiCbkSetStatus {
  LITEQMI_QMI_CBK_PARAM_RESET = 0,
  LITEQMI_QMI_CBK_PARAM_SET = 1,
  LITEQMI_QMI_CBK_PARAM_NOCHANGE }

## Functions

- int pack_sms_SLQSGetSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSMS_t *reqParam)
- int unpack_sms_SLQSGetSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetSMS_t *pOutput)
- int pack_sms_SLQSGetSMSList (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSMSList_t *reqParam)
- int unpack_sms_SLQSGetSMSList (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetSMSList_t *pOutput)
- int pack_sms_SLQSModifySMSStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSModifySMSStatus_t *reqParam)
- int unpack_sms_SLQSModifySMSStatus (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSModifySMSStatus_t *pOutput)

- int pack_sms_SLQSDeleteSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSDeleteSMS_t *reqParam)
- int unpack_sms_SLQSDeleteSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSDeleteSMS_t *pOutput)
- int pack_sms_SendSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SendSMS_t *reqParam)
- int unpack_sms_SendSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SendSMS_t *pOutput)
- int pack_sms_SetNewSMSCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SetNewSMSCallback_t reqParam)
- int unpack_sms_SetNewSMSCallback (uint8_t *pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_t *Output)
- int unpack_sms_SetNewSMSCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_ind_t *pOutput)
- int unpack_sms_SLQSWmsMemoryFullCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSWmsMemoryFullCallBack_ind_t *pOutput)
- int pack_sms_GetSMSCAddress (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_GetSMSCAddress (uint8_t *pResp, uint16_t respLen, unpack_sms_GetSMSCAddress_t *pOutput)
- int pack_sms_SetSMSCAddress (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SetSMSCAddress_t *reqParam)
- int unpack_sms_SetSMSCAddress (uint8_t *pResp, uint16_t respLen, unpack_sms_SetSMSCAddress_t *pOutput)
- int pack_sms_SaveSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SaveSMS_t *reqParam)
- int unpack_sms_SaveSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SaveSMS_t *pOutput)
- int pack_sms_SLQSGetSmsBroadcastConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSGetSmsBroadcastConfig_t *reqParam)
- int unpack_sms_SLQSGetSmsBroadcastConfig (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetSmsBroadcastConfig_t *pOutput)
- int pack_sms_SLQSSetSmsBroadcastConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetSmsBroadcastConfig_t *reqParam)
- int unpack_sms_SLQSSetSmsBroadcastConfig (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSetSmsBroadcastConfig_t *pOutput)
- int pack_sms_SLQSSetSmsBroadcastActivation (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetSmsBroadcastActivation_t *reqParam)
- int unpack_sms_SLQSSetSmsBroadcastActivation (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSetSmsBroadcastActivation_t *pOutput)
- int pack_sms_SLQSGetTransLayerInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetTransLayerInfo (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetTransLayerInfo_t *pOutput)
- int pack_sms_SLQSGetTransNWRegInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetTransNWRegInfo (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetTransNWRegInfo_t *pOutput)
- int pack_sms_SLQSGetIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetIndicationRegister_t *pOutput)
- int pack_sms_SLQSSetIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetIndicationRegister_t *reqParam)
- int unpack_sms_SLQSSetIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSetIndicationRegister_t *pOutput)
- int pack_sms_SLQSSmsSetRoutes (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSmsSetRoutes_t *reqParam)
- int unpack_sms_SLQSSmsSetRoutes (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSmsSetRoutes_t *pOutput)
- int pack_sms_SLQSSmsGetMessageProtocol (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)

- int unpack_sms_SLQSSmsGetMessageProtocol (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSmsGetMessageProtocol_t *pOutput)
- int pack_sms_SLQSSmsGetMaxStorageSize (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSmsGetMaxStorageSize_t *reqParam)
- int unpack_sms_SLQSSmsGetMaxStorageSize (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSmsGetMaxStorageSize_t *pOutput)
- int pack_sms_SLQSGetMessageWaiting (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSGetMessageWaiting (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSGetMessageWaiting_t *pOutput)
- int pack_sms_SLQSSendAsyncSMS (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSendAsyncSMS_t *reqParam)
- int unpack_sms_SLQSSendAsyncSMS (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSendAsyncSMS_t *pOutput)
- int pack_sms_SLQSSetSmsStorage (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_sms_SLQSSetSmsStorage_t *reqParam)
- int unpack_sms_SLQSSetSmsStorage (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSetSmsStorage_t *pOutput)
- int pack_sms_SLQSSwiGetSMSStorage (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_sms_SLQSSwiGetSMSStorage (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSSwiGetSMSStorage_t *pOutput)
- int unpack_sms_SLQSTransLayerInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSTransLayerInfoCallback_ind_t *pOutput)
- int unpack_sms_SLQSNWRegInfoCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSNWRegInfoCallback_ind_t *pOutput)
- int unpack_sms_SLQSWmsMessageWaitingCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t *pOutput)
- int unpack_sms_SLQSWmsAsyncRawSendCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t *pOutput)

### 9.32.1 Macro Definition Documentation

#### 9.32.1.1 #define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255

#### 9.32.1.2 #define MAX_MS_TRANSFER_ROUTE_MSG 256

#### 9.32.1.3 #define MAX_MSC_ADDRESS_SIZE 256

#### 9.32.1.4 #define MAX_MSE_TWS_MSG 1254

#### 9.32.1.5 #define MAX_SMS_LIST_SIZE 255

#### 9.32.1.6 #define MAX_SMS_MESSAGE_SIZE 2048

#### 9.32.1.7 #define SMS_CONFIG_LEN 0x05

#### 9.32.1.8 #define SMS_MAX_SMS_ROUTES 0x0A

#### 9.32.1.9 #define SMS_NUM_OF_SET 0xFF

#### 9.32.1.10 #define SMSC_TYPE_LEN 0x03

### 9.32.2 Enumeration Type Documentation

#### 9.32.2.1 enum eqmiCbkSetStatus

**Enumerator**

***LITEQMI_QMI_CBK_PARAM_RESET***

***LITEQMI_QMI_CBK_PARAM_SET***

***LITEQMI_QMI_CBK_PARAM_NOCHANGE***

### 9.32.3 Function Documentation

#### 9.32.3.1 int pack_sms_GetSMSCAddress ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets the SMS center address pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.2 int pack_sms_SaveSMS ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SaveSMS_t * *reqParam* )

Saves an SMS message to device memory pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.3 int pack_sms_SendSMS ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SendSMS_t * *reqParam* )

send sms list pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.4 int pack_sms_SetNewSMSCallback ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SetNewSMSCallback_t *reqParam* )

set new sms callback pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.5 int pack_sms_SetSMSCAddress ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SetSMSCAddress_t * *reqParam* )

Set the SMS center address pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.6 int pack_sms_SLQSDeleteSMS ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSDeleteSMS_t * reqParam )

delete sms pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.7 int pack_sms_SLQSGetIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Gets registration state of different WMS indications pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.8 int pack_sms_SLQSGetMessageWaiting ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Gets the message waiting information pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.9 int pack_sms_SLQSGetSMS ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSGetSMS_t * *reqParam* )

get sms pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.10 int pack_sms_SLQSGetSmsBroadcastConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSGetSmsBroadcastConfig_t * *reqParam* )

Provides Information about the SMS BroadcastConfiguration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.11 int pack_sms_SLQSGetSMSList ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSGetSMSList_t * *reqParam* )

get sms list pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.12 int pack_sms_SLQSGetTransLayerInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets information about the transport layer pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.13 int pack_sms_SLQSGetTransNWRegInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets transport layer network registration info pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.14 int pack_sms_SLQSModifySMSStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSModifySMSStatus_t * *reqParam* )

modify sms status pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.15 int pack_sms_SLQSSendAsyncSMS ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSendAsyncSMS_t * *reqParam* )

Sends an SMS message for immediate over-the-air transmission pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.16 int pack_sms_SLQSSetIndicationRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetIndicationRegister_t * *reqParam* )

Sets the registration state of different WMS indications pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.17 int pack_sms_SLQSSetSmsBroadcastActivation ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetSmsBroadcastActivation_t * *reqParam* )

Enables or disables the reception of broadcast SMS messages pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.18 int pack_sms_SLQSSetSmsBroadcastConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetSmsBroadcastConfig_t * *reqParam* )

Sets the information about the SMS BroadcastConfiguration pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.19 int pack_sms_SLQSSetSmsStorage ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_sms_SLQSSetSmsStorage_t * *reqParam* )

Sets the SMS Storage on the device pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.20 int pack_sms_SLQSSmsGetMaxStorageSize ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSmsGetMaxStorageSize_t * reqParam )

Get the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.21 int pack_sms_SLQSSmsGetMessageProtocol ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Gets the message protocol currently in use for the WMS client pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.22 int pack_sms_SLQSSmsSetRoutes ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSmsSetRoutes_t * reqParam )

Sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| | *reqParam* | packed request |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.23 int pack_sms_SLQSSwiGetSMSStorage ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets the current SMS configuration that is applied to all incoming and outgoing messages pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.24 int unpack_sms_GetSMSCAddress ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_GetSMSCAddress_t * *pOutput* )

Gets the SMS center address unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.25 int unpack_sms_SaveSMS ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SaveSMS_t * *pOutput* )

Saves an SMS message to device memory unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.26 int unpack_sms_SendSMS ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SendSMS_t** * *pOutput* )

send sms unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.27 int unpack_sms_SetNewSMSCallback ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SetNewSMSCallback_t** * *Output* )

set new sms callback unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *Output* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.28 int unpack_sms_SetNewSMSCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SetNewSMSCallback_ind_t** * *pOutput* )

set new sms callback indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.29 int unpack_sms_SetSMSCAddress ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SetSMSCAddress_t** * *pOutput* )

Set the SMS center address unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.30 int unpack_sms_SLQSDeleteSMS ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSDeleteSMS_t** * *pOutput* )

delete sms unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.31 int unpack_sms_SLQSGetIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSGetIndicationRegister_t** * *pOutput* )

Gets registration state of different WMS indications unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.32 int unpack_sms_SLQSGetMessageWaiting ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetMessageWaiting_t * *pOutput* )

Gets the message waiting information unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.33 int unpack_sms_SLQSGetSMS ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSMS_t * *pOutput* )

get sms unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.34 int unpack_sms_SLQSGetSmsBroadcastConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSGetSmsBroadcastConfig_t * *pOutput* )

Provides Information about the SMS BroadcastConfiguration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.35 int unpack_sms_SLQSGetSMSList ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSGetSMSList_t** * *pOutput* )

get sms list unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.36 int unpack_sms_SLQSGetTransLayerInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSGetTransLayerInfo_t** * *pOutput* )

Gets information about the transport layer unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.37 int unpack_sms_SLQSGetTransNWRegInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSGetTransNWRegInfo_t** * *pOutput* )

Gets transport layer network registration info unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.38 int unpack_sms_SLQSModifySMSStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSModifySMS-Status_t * *pOutput* )

modify sms status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.39 int unpack_sms_SLQSNWRegInfoCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSNWRegInfoCallback_ind_t * *pOutput* )

Unpack indication about change in transport layer info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.40 int unpack_sms_SLQSSendAsyncSMS ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSendAsyncS-MS_t * *pOutput* )

Sends an SMS message for immediate over-the-air transmission unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.41 int unpack_sms_SLQSSetIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSetIndicationRegister_t * *pOutput* )

Sets the registration state of different WMS indications unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.42 int unpack_sms_SLQSSetSmsBroadcastActivation ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSetSmsBroadcastActivation_t * *pOutput* )

Enables or disables the reception of broadcast SMS messages unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.43 int unpack_sms_SLQSSetSmsBroadcastConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSetSmsBroadcastConfig_t * *pOutput* )

Sets the information about the SMS BroadcastConfiguration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.44 int unpack_sms_SLQSSetSmsStorage ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSetSmsStorage_t * *pOutput* )

Sets the SMS Storage on the device unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.45 int unpack_sms_SLQSSmsGetMaxStorageSize ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSmsGetMaxStorageSize_t * *pOutput* )

Get the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.46 int unpack_sms_SLQSSmsGetMessageProtocol ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSSmsGetMessageProtocol_t * *pOutput* )

Gets the message protocol currently in use for the WMS client unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.47 int unpack_sms_SLQSSmsSetRoutes ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSSmsSetRoutes_t** * *pOutput* )

Sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.48 int unpack_sms_SLQSSwiGetSMSStorage ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSSwiGetSMSStorage_t** * *pOutput* )

Gets the current SMS configuration that is applied to all incoming and outgoing messages unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.49 int unpack_sms_SLQSTransLayerInfoCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSTransLayerInfoCallback_ind_t** * *pOutput* )

Unpack indication about change in transport layer info

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.50 int unpack_sms_SLQSWmsAsyncRawSendCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t** * *pOutput* )

Unpack indication for sms async raw send

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.51 int unpack_sms_SLQSWmsMemoryFullCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_sms_SLQSWmsMemoryFullCallBack_ind_t** * *pOutput* )

sms full callback indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.32.3.52 int unpack_sms_SLQSWmsMessageWaitingCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t * *pOutput* )

Unpack indication for message waiting information

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.33 swiaudio.h File Reference

### Data Structures

- struct pack_swiaudio_SLQSGetM2MAudioProfile_t
- struct unpack_swiaudio_SLQSGetM2MAudioProfile_t
- struct pack_swiaudio_SLQSSetM2MAudioProfile_t
- struct pack_swiaudio_SLQSGetM2MAudioVolume_t
- struct unpack_swiaudio_SLQSGetM2MAudioVolume_t
- struct pack_swiaudio_SLQSSetM2MAudioVolume_t
- struct swiaudio_PCMparams
- struct pack_swiaudio_SLQSSetM2MAudioAVCFG_t
- struct pack_swiaudio_SLQSSetM2MAudioLPBK_t
- struct pack_swiaudio_SLQSGetM2MSpkrGain_t
- struct unpack_swiaudio_SLQSGetM2MSpkrGain_t
- struct pack_swiaudio_SLQSSetM2MSpkrGain_t
- struct pack_swiaudio_SLQSGetM2MAVMute_t
- struct unpack_swiaudio_SLQSGetM2MAVMute_t
- struct pack_swiaudio_SLQSSetM2MAVMute_t

### Macros

- #define SWIAUDIO_MAX_LEN_IFACE_TABLE 255

### Typedefs

- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioProfile_t
- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioVolume_t
- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioAVCFG_t
- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioLPBK_t
- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioNVDef_t
- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MSpkrGain_t
- typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAVMute_t

### Functions

- int pack_swiaudio_SLQSGetM2MAudioProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSGetM2MAudioProfile_t *pReqParam)
- int unpack_swiaudio_SLQSGetM2MAudioProfile (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSGetM2MAudioProfile_t *pOutput)
- int pack_swiaudio_SLQSSetM2MAudioProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSSetM2MAudioProfile_t *pReqParam)
- int unpack_swiaudio_SLQSSetM2MAudioProfile (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MAudioProfile_t *pOutput)
- int pack_swiaudio_SLQSGetM2MAudioVolume (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSGetM2MAudioVolume_t *pReqParam)
- int unpack_swiaudio_SLQSGetM2MAudioVolume (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSGetM2MAudioVolume_t *pOutput)
- int pack_swiaudio_SLQSSetM2MAudioVolume (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSSetM2MAudioVolume_t *pReqParam)
- int unpack_swiaudio_SLQSSetM2MAudioVolume (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MAudioVolume_t *pOutput)
- int pack_swiaudio_SLQSSetM2MAudioAVCFG (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSSetM2MAudioAVCFG_t *pReqParam)
- int unpack_swiaudio_SLQSSetM2MAudioAVCFG (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MAudioAVCFG_t *pOutput)
- int pack_swiaudio_SLQSSetM2MAudioLPBK (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSSetM2MAudioLPBK_t *pReqParam)
- int unpack_swiaudio_SLQSSetM2MAudioLPBK (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MAudioLPBK_t *pOutput)
- int pack_swiaudio_SLQSSetM2MAudioNVDef (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swiaudio_SLQSSetM2MAudioNVDef (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MAudioNVDef_t *pOutput)
- int pack_swiaudio_SLQSGetM2MSpkrGain (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSGetM2MSpkrGain_t *pReqParam)
- int unpack_swiaudio_SLQSGetM2MSpkrGain (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSGetM2MSpkrGain_t *pOutput)
- int pack_swiaudio_SLQSSetM2MSpkrGain (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSSetM2MSpkrGain_t *pReqParam)
- int unpack_swiaudio_SLQSSetM2MSpkrGain (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MSpkrGain_t *pOutput)
- int pack_swiaudio_SLQSGetM2MAVMute (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSGetM2MAVMute_t *pReqParam)
- int unpack_swiaudio_SLQSGetM2MAVMute (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSGetM2MAVMute_t *pOutput)
- int pack_swiaudio_SLQSSetM2MAVMute (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiaudio_SLQSSetM2MAVMute_t *pReqParam)
- int unpack_swiaudio_SLQSSetM2MAVMute (uint8_t *pResp, uint16_t respLen, unpack_swiaudio_SLQSSetM2MAVMute_t *pOutput)

## 9.33.1 Macro Definition Documentation

#### 9.33.1.1 #define SWIAUDIO_MAX_LEN_IFACE_TABLE 255

## 9.33.2 Typedef Documentation

#### 9.33.2.1 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioAVCFG_t

#### 9.33.2.2 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioLPBK_t

#### 9.33.2.3 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioNVDef_t

#### 9.33.2.4 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioProfile_t

#### 9.33.2.5 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioVolume_t

#### 9.33.2.6 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAVMute_t

#### 9.33.2.7 typedef unpack_result_t unpack_swiaudio_SLQSSetM2MSpkrGain_t

### 9.33.3 Function Documentation

#### 9.33.3.1 int pack_swiaudio_SLQSGetM2MAudioProfile ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSGetM2MAudioProfile_t * *pReqParam* )

Gets the profile content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.2 int pack_swiaudio_SLQSGetM2MAudioVolume ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSGetM2MAudioVolume_t * *pReqParam* )

Gets the Volume content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.3 int pack_swiaudio_SLQSGetM2MAVMute ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSGetM2MAVMute_t * *pReqParam* )

Gets the AV Mute content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.4 int pack_swiaudio_SLQSGetM2MSpkrGain ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSGetM2MSpkrGain_t * *pReqParam* )

Gets the SPKRGAIN content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.5 int pack_swiaudio_SLQSSetM2MAudioAVCFG ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSSetM2MAudioAVCFG_t * *pReqParam* )

Sets the AVCFG content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.6 int pack_swiaudio_SLQSSetM2MAudioLPBK ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAudioLPBK_t * pReqParam )

Sets the LPBK content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.7 int pack_swiaudio_SLQSSetM2MAudioNVDef ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Sets the NVDef content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.8 int pack_swiaudio_SLQSSetM2MAudioProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAudioProfile_t * pReqParam )

sets an audio profile content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.9 int pack_swiaudio_SLQSSetM2MAudioVolume ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSSetM2MAudioVolume_t * *pReqParam* )

Sets the Volume content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.10 int pack_swiaudio_SLQSSetM2MAVMute ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSSetM2MAVMute_t * *pReqParam* )

Sets the AV Mute content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.11 int pack_swiaudio_SLQSSetM2MSpkrGain ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiaudio_SLQSSetM2MSpkrGain_t * *pReqParam* )

Sets the SPKRGAIN content pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameters |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.12 int unpack_swiaudio_SLQSGetM2MAudioProfile ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_swiaudio_SLQSGetM2MAudioProfile_t** * *pOutput* )

Gets the profile content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.13 int unpack_swiaudio_SLQSGetM2MAudioVolume ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_swiaudio_SLQSGetM2MAudioVolume_t** * *pOutput* )

Gets the Volume content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.14 int unpack_swiaudio_SLQSGetM2MAVMute ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_swiaudio_SLQSGetM2MAVMute_t** * *pOutput* )

Gets the AV Mute content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.15 int unpack_swiaudio_SLQSGetM2MSpkrGain ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSGetM2MSpkrGain_t * *pOutput* )

Gets the SPKRGAIN content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.16 int unpack_swiaudio_SLQSSetM2MAudioAVCFG ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MAudioAVCFG_t * *pOutput* )

Sets the AVCFG content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.17 int unpack_swiaudio_SLQSSetM2MAudioLPBK ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MAudioLPBK_t * *pOutput* )

Sets the LPBK content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.18 int unpack_swiaudio_SLQSSetM2MAudioNVDef ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MAudioNVDef_t * *pOutput* )

Sets the NVDef content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.19 int unpack_swiaudio_SLQSSetM2MAudioProfile ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MAudioProfile_t * *pOutput* )

sets an audio profile content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.20 int unpack_swiaudio_SLQSSetM2MAudioVolume ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MAudioVolume_t * *pOutput* )

Sets the Volume content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.21 int unpack_swiaudio_SLQSSetM2MAVMute ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MAVMute_t * *pOutput* )

Sets the AV Mute content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.33.3.22 int unpack_swiaudio_SLQSSetM2MSpkrGain ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiaudio_SLQSSetM2MSpkrGain_t * *pOutput* )

Sets the SPKRGAIN content unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.34 swiavms.h File Reference

### Data Structures

- struct pack_swiavms_SLQSAVMSStartSession_t
- struct unpack_swiavms_SLQSAVMSStartSession_t
- struct pack_swiavms_SLQSAVMSStopSession_t

- struct unpack_swiavms_SLQSAVMSStopSession_t
- struct PackSwiAvmsSetSettingsPeriodInfo
- struct PackSwiAvmsSetSettingsAPNInfo
- struct PackSwiAvmsSetSettingsConnectionRetryTimers
- struct pack_swiavms_SLQSAVMSSetSettings_t
- struct unpack_swiavms_SLQSAVMSSetSettings_t
- struct pack_swiavms_SLQSAVMSSetSettings_v2_t
- struct unpack_swiavms_SLQSAVMSSetSettings_v2_t
- struct PackSwiAVMSSettingsAPNInfo
- struct PackSwiAVMSSettingsConnectionRetryTimers
- struct PackSwiAVMSSettingsPeriodsInfo
- struct unpack_swiavms_SLQSAVMSGetSettings_t
- struct unpack_swiavms_SLQSAVMSGetSettings_v2_t
- struct pack_swiavms_SLQSAVMSSendSelection_t
- struct unpack_swiavms_SLQSAVMSSendSelection_t
- struct unpack_swiavms_SLQSAvmsSetEventReport_t
- struct UnpackSwiAvmsEventReportBinaryUpdateSessionInfo
- struct UnpackSwiAvmsEventReportConfig
- struct UnpackSwiAvmsEventReportNotification
- struct UnpackSwiAvmsEventReportConnectionRequest
- struct UnpackSwiAvmsEventReportWAMSParamChange
- struct UnpackSwiAvmsEventReportPackageID
- struct UnpackSwiAvmsEventReportRegStatus
- struct UnpackSwiAvmsEventReportDataSessionStatus
- struct UnpackSwiAvmsEventReportSessionType
- struct UnpackSwiAvmsEventReportHTTPStatus
- struct unpack_swiavms_SLQSAVMSSessionGetInfo_t
- struct unpack_swiavms_SLQSAVMSEventReportInd_t

## Macros

- #define LITEQMI_MAX_SWIOMA_STR_LEN 255
- #define MAX_PACK_SWI_AVMS_SET_SETTING_APN_LENGTH 49
- #define MAX_PACK_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29
- #define MAX_PACK_SWI_AVMS_SET_SETTING_PWD_LENGTH 29
- #define MAX_PACK_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8
- #define MAX_AVMS_SETTINGS_RETRY_TIMER_NUMBER 8
- #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 128
- #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 128
- #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1024
- #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200

## Functions

- int pack_swiavms_SLQSAVMSStartSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiavms_SLQSAVMSStartSession_t reqParam)
- int unpack_swiavms_SLQSAVMSStartSession (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSStartSession_t *pResponse)
- int pack_swiavms_SLQSAVMSStopSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiavms_SLQSAVMSStopSession_t reqParam)

- int unpack_swiavms_SLQSAVMSStopSession (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSStopSession_t *pResponse)
- int unpack_swiavms_SLQSAVMSStopSession_avc2 (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSStopSession_t *pResponse)
- int pack_swiavms_SLQSAVMSSetSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiavms_SLQSAVMSSetSettings_t reqParam)
- int unpack_swiavms_SLQSAVMSSetSettings (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSSetSettings_t *pResponse)
- int pack_swiavms_SLQSAVMSSetSettings_v2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiavms_SLQSAVMSSetSettings_v2_t reqParam)
- int unpack_swiavms_SLQSAVMSSetSettings_v2 (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSSetSettings_v2_t *pResponse)
- int pack_swiavms_SLQSAVMSSetSettingsNoAutoRebootField (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiavms_SLQSAVMSSetSettings_t reqParam)
- int pack_swiavms_SLQSAVMSGetSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swiavms_SLQSAVMSGetSettings (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSGetSettings_t *pResponse)
- int pack_swiavms_SLQSAVMSGetSettings_v2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swiavms_SLQSAVMSGetSettings_v2 (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSGetSettings_v2_t *pResponse)
- int pack_swiavms_SLQSAVMSSendSelection (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swiavms_SLQSAVMSSendSelection_t reqParam)
- int unpack_swiavms_SLQSAVMSSendSelection (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSSendSelection_t *pResponse)
- int pack_swiavms_SLQSAvmsSetEventReport (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swiavms_SLQSAvmsSetEventReport (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAvmsSetEventReport_t *pResponse)
- int pack_swiavms_SLQSAVMSSessionGetInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swiavms_SLQSAVMSSessionGetInfo (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSSessionGetInfo_t *pResponse)
- int unpack_swiavms_SLQSAVMSEventReportInd (uint8_t *pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSEventReportInd_t *pResponse)

### 9.34.1 Macro Definition Documentation

#### 9.34.1.1 #define LITEQMI_MAX_SWIOMA_STR_LEN 255

#### 9.34.1.2 #define MAX_AVMS_SETTINGS_RETRY_TIMER_NUMBER 8

#### 9.34.1.3 #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH 1024

#### 9.34.1.4 #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH 128

#### 9.34.1.5 #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH 128

#### 9.34.1.6 #define MAX_PACK_SWI_AVMS_SESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200

#### 9.34.1.7 #define MAX_PACK_SWI_AVMS_SET_SETTING_APN_LENGTH 49

#### 9.34.1.8 #define MAX_PACK_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8

#### 9.34.1.9 #define MAX_PACK_SWI_AVMS_SET_SETTING_PWD_LENGTH 29

#### 9.34.1.10 #define MAX_PACK_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29

### 9.34.2 Function Documentation

#### 9.34.2.1 int pack_swiavms_SLQSAVMSGetSettings ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Function to pack AVMS Get settings command This maps to SLQSAVMSSetSettings

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.2 int pack_swiavms_SLQSAVMSGetSettings_v2 ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Function to pack AVMS Get settings command This maps to SLQSAVMSSetSettings_v2 (For AVC2 service)

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.3 int pack_swiavms_SLQSAVMSSendSelection ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiavms_SLQSAVMSSendSelection_t *reqParam* )

Function to pack AVMS send selection command This maps to SLQSAVMSSendSelection

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_swiavms_SLQSAVMSSendSelection_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.4 int pack_swiavms_SLQSAVMSSessionGetInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Function to pack AVMS get session info command This maps to SLQSAVMSSessionGetInfo

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |

| | | |
|---|---|---|
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.5 int pack_swiavms_SLQSAvmsSetEventReport ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Function to pack AVMS set event report command This maps to SLQSAVMSSetEventReport

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.6 int pack_swiavms_SLQSAVMSSetSettings ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiavms_SLQSAVMSSetSettings_t *reqParam* )

Function to pack AVMS Set settings command This maps to SLQSAVMSSetSettings

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_swiavms_SLQSAVMSSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.7 int pack_swiavms_SLQSAVMSSetSettings_v2 ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiavms_SLQSAVMSSetSettings_v2_t *reqParam* )

Function to pack AVMS Set settings command This maps to SLQSAVMSSetSettings_v2 (For AVC2 service)

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_swiavms_SLQSAVMSSetSettings_v2_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.8 int pack_swiavms_SLQSAVMSSetSettingsNoAutoRebootField ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiavms_SLQSAVMSSetSettings_t *reqParam* )

Function to pack AVMS Set settings without Auto Reobot field command This maps to SLQSAVMSSetSettings-IgnoreAutoReboot

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_swiavms_SLQSAVMSSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.9 int pack_swiavms_SLQSAVMSStartSession ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiavms_SLQSAVMSStartSession_t *reqParam* )

Function to pack Start AVMS session command This maps to SLQSAVMSStartSession

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |

| | | |
|---|---|---|
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_swiavms_SLQSAVMSStartSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.10 int pack_swiavms_SLQSAVMSStopSession ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swiavms_SLQSAVMSStopSession_t *reqParam* )

Function to pack cancel AVMS session command This maps to SLQSAVMSStopSession

**Parameters**

| | | |
|---|---|---|
| out | *pCtx* | • See pack_qmi_t for more information |
| in,out | *pReqBuf* | • Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| in,out | *pLen* | • On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| in | *reqParam* | • See pack_swiavms_SLQSAVMSStopSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 2 seconds

#### 9.34.2.11 int unpack_swiavms_SLQSAVMSEventReportInd ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSEventReportInd_t * *pResponse* )

Function to unpack AVMS event report Indication command This maps to SLQSAVMSSetEventReport

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSEventReportInd_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.12 int unpack_swiavms_SLQSAVMSGetSettings ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSGetSettings_t * *pResponse* )

Function to unpack AVMS set settings command This maps to SLQSAVMSGetSettings

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.13 int unpack_swiavms_SLQSAVMSGetSettings_v2 ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSGetSettings_v2_t * *pResponse* )

Function to unpack AVMS set settings command This maps to SLQSAVMSGetSettings_v2 (For AVC2 service)

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.14 int unpack_swiavms_SLQSAVMSSendSelection ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSSendSelection_t * *pResponse* )

Function to unpack AVMS send selection command This maps to SLQSAVMSSendSelection

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSSendSelection_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.15 int unpack_swiavms_SLQSAVMSSessionGetInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSSessionGetInfo_t * *pResponse* )

Function to unpack AVMS event report Indication command This maps to SLQSAVMSSetEventReport

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSSessionGetInfo_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.16 int unpack_swiavms_SLQSAvmsSetEventReport ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAvmsSetEventReport_t * *pResponse* )

Function to unpack AVMS set event report command This maps to SLQSAVMSSetEventReport

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAvmsSetEventReport_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.17 int unpack_swiavms_SLQSAVMSSetSettings ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSSetSettings_t * *pResponse* )

Function to unpack AVMS Set settings command This maps to SLQSAVMSSetSettings

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |

| | | |
|---|---|---|
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.18 int unpack_swiavms_SLQSAVMSSetSettings_v2 ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSSetSettings_v2_t * *pResponse* )

Function to unpack AVMS Set settings command This maps to SLQSAVMSSetSettings_v2 (For AVC2 service)

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSSetSettings_v2_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.34.2.19 int unpack_swiavms_SLQSAVMSStartSession ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSStartSession_t * *pResponse* )

Function to unpack Start AVMS session response from modem This maps to SLQSAVMSStartSession

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |

| | | |
|---|---|---|
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSStartSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.34.2.20 int unpack_swiavms_SLQSAVMSStopSession ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSStopSession_t * *pResponse* )

Function to pack stop AVMS session command This maps to SLQSAVMSStopSession

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSStopSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.34.2.21 int unpack_swiavms_SLQSAVMSStopSession_avc2 ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiavms_SLQSAVMSStopSession_t * *pResponse* )

Function to pack stop AVMS session command This maps to SLQSAVMSStopSession Note: this targets WP760x as the stop response doesn't contains mandatory session_type(0x01) TLV

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | • Response from modem |
| in | *respLen* | • Length of pResp from modem |
| out | *pResponse* | • See unpack_swiavms_SLQSAVMSStopSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.35 SwiDataTypes.h File Reference

SWI data types.

### Macros

- #define SWI_API
- #define QMI_NO_LTE_FW_SUPPORT 0
- #define QMI_TLV_PLACEHOLDER 0x8F
- #define UNUSEDPARAM(x) (void)x

### Typedefs

- typedef unsigned int ULONG
- typedef unsigned long long ULONGLONG
- typedef signed char INT8
- typedef unsigned char BYTE
- typedef char CHAR
- typedef unsigned short WORD
- typedef unsigned short USHORT
- typedef const char * LPCSTR
- typedef int BOOL
- typedef signed short SHORT
- typedef signed int INT32
- typedef float FLOAT

### 9.35.1 Detailed Description

SWI data types.

### 9.35.2 Macro Definition Documentation

#### 9.35.2.1 #define QMI_NO_LTE_FW_SUPPORT 0

#### 9.35.2.2 #define QMI_TLV_PLACEHOLDER 0x8F

#### 9.35.2.3 #define SWI_API

#### 9.35.2.4 #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.

### 9.35.3 Typedef Documentation

#### 9.35.3.1 typedef int BOOL

#### 9.35.3.2 typedef unsigned char BYTE

#### 9.35.3.3 typedef char CHAR

#### 9.35.3.4 typedef float FLOAT

#### 9.35.3.5 typedef signed int INT32

#### 9.35.3.6 typedef signed char INT8

#### 9.35.3.7 typedef const char* LPCSTR

#### 9.35.3.8 typedef signed short SHORT

#### 9.35.3.9 typedef unsigned int ULONG

#### 9.35.3.10 typedef unsigned long long ULONGLONG

#### 9.35.3.11 typedef unsigned short USHORT

#### 9.35.3.12 typedef unsigned short WORD

## 9.36 swidms.h File Reference

### Data Structures

- struct unpack_swidms_SLQSSwiDmsGetUsbNetNum_t
- struct pack_swidms_SLQSSwiDmsSetUsbNetNum_t
- struct pack_swidms_SLQSSwiDmsSetMTU_t
- struct unpack_swidms_SLQSSwiDmsSetMTU_t
- struct swidms_mtuSize3gppTlv
- struct swidms_hrpdMTUSizeTlv
- struct swidms_ehrpdMTUSizeTlv
- struct swidms_usbMTUSizeTlv
- struct unpack_swidms_SLQSSwiDmsGetMTU_t

### Typedefs

- typedef unpack_result_t unpack_swidms_SLQSSwiDmsSetUsbNetNum_t

### Functions

- int pack_swidms_SLQSSwiDmsGetUsbNetNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swidms_SLQSSwiDmsGetUsbNetNum (uint8_t *pResp, uint16_t respLen, unpack_swidms_SLQSSwiDmsGetUsbNetNum_t *pOutput)
- int pack_swidms_SLQSSwiDmsSetUsbNetNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swidms_SLQSSwiDmsSetUsbNetNum_t *pReqParam)
- int unpack_swidms_SLQSSwiDmsSetUsbNetNum (uint8_t *pResp, uint16_t respLen, unpack_swidms_SLQSSwiDmsSetUsbNetNum_t *pOutput)

- int pack_swidms_SLQSSwiDmsSetMTU (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swidms_SLQSSwiDmsSetMTU_t *reqArg)
- int unpack_swidms_SLQSSwiDmsSetMTU (uint8_t *pResp, uint16_t respLen, unpack_swidms_SLQSSwiDmsSetMTU_t *pOutput)
- int pack_swidms_SLQSSwiDmsGetMTU (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_swidms_SLQSSwiDmsGetMTU (uint8_t *pResp, uint16_t respLen, unpack_swidms_SLQSSwiDmsGetMTU_t *pOutput)

## 9.36.1 Typedef Documentation

### 9.36.1.1 typedef unpack_result_t unpack_swidms_SLQSSwiDmsSetUsbNetNum_t

## 9.36.2 Function Documentation

### 9.36.2.1 int pack_swidms_SLQSSwiDmsGetMTU ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Get the preferred MTU Size of the 3GPP, HRPD, EHRPD and USB descriptor Interfaces pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.36.2.2 int pack_swidms_SLQSSwiDmsGetUsbNetNum ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

pack function to get usb net numbers for QMAP configuration

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.36.2.3 int pack_swidms_SLQSSwiDmsSetMTU ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swidms_SLQSSwiDmsSetMTU_t * *reqArg* )

Sets the preferred MTU size for 3GPP, HRPD, EHRPD Interfaces pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.36.2.4 int pack_swidms_SLQSSwiDmsSetUsbNetNum ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swidms_SLQSSwiDmsSetUsbNetNum_t * *pReqParam* )

pack function to set usb net numbers for QMAP configuration

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request params |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.36.2.5 int unpack_swidms_SLQSSwiDmsGetMTU ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swidms_SLQSSwiDmsGetMTU_t * *pOutput* )

Get the MTU Size of the 3GPP, HRPD, EHRPD and USB descriptor Interfaces unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.36.2.6 int unpack_swidms_SLQSSwiDmsGetUsbNetNum ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swidms_SLQSSwiDmsGetUsbNetNum_t * *pOutput* )

unpack function to get usb net numbers for QMAP configuration

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.36.2.7 int unpack_swidms_SLQSSwiDmsSetMTU ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swidms_SLQSSwiDmsSetMTU_t * *pOutput* )

Sets preferred MTU size for 3GPP, HRPD, EHRPD Interfaces unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.36.2.8 int unpack_swidms_SLQSSwiDmsSetUsbNetNum ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swidms_SLQSSwiDmsSetUsbNetNum_t * *pOutput* )

unpack function to set usb net numbers for QMAP configuration

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.37 swiloc.h File Reference

### Data Structures

- struct unpack_swiloc_SwiLocGetAutoStart_t
- struct pack_swiloc_SwiLocSetAutoStart_t

### Typedefs

- typedef unpack_result_t unpack_swiloc_SwiLocSetAutoStart_t

### Functions

- int pack_swiloc_SwiLocGetAutoStart (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_swiloc_SwiLocGetAutoStart (uint8_t ∗pResp, uint16_t respLen, unpack_swiloc_SwiLocGetAutoStart_t ∗pOutput)
- int pack_swiloc_SwiLocSetAutoStart (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_swiloc_SwiLocSetAutoStart_t ∗reqArg)
- int unpack_swiloc_SwiLocSetAutoStart (uint8_t ∗pResp, uint16_t respLen, unpack_swiloc_SwiLocSetAutoStart_t ∗pOutput)

### 9.37.1 Typedef Documentation

#### 9.37.1.1 typedef unpack_result_t unpack_swiloc_SwiLocSetAutoStart_t

### 9.37.2 Function Documentation

#### 9.37.2.1 int pack_swiloc_SwiLocGetAutoStart ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen* )

Get Auto Start pack

**Parameters**

| | | |
|---|---|---|
| in | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.37.2.2 int pack_swiloc_SwiLocSetAutoStart ( pack_qmi_t ∗ *pCtx*, uint8_t ∗ *pReqBuf*, uint16_t ∗ *pLen*, pack_swiloc_SwiLocSetAutoStart_t ∗ *reqArg* )

Set Auto Start pack

**Parameters**

| | | |
|---|---|---|
| in | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.37.2.3 int unpack_swiloc_SwiLocGetAutoStart ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiloc_SwiLocGetAutoStart_t * *pOutput* )

Get Auto Start unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.37.2.4 int unpack_swiloc_SwiLocSetAutoStart ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swiloc_SwiLocSetAutoStart_t * *pOutput* )

Set Auto Start unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.38 swioma.h File Reference

### Data Structures

- struct pack_swioma_SLQSOMADMStartSession_t
- struct unpack_swioma_SLQSOMADMStartSession_t
- struct pack_swioma_SLQSOMADMCancelSession_t
- struct unpack_swioma_SLQSOMADMGetSettings_t
- struct pack_swioma_SLQSOMADMSetSettings_t
- struct pack_swioma_SLQSOMADMSendSelection_t
- struct pack_swioma_SLQSOMADMGetSessionInfo_t
- struct unpack_swioma_SLQSOMADMGetSessionInfo_t
- struct unpack_omaDmFotaTlv_t
- struct unpack_omaDmConfigTlv_t
- struct unpack_omaDmNotificationsTlv_t
- struct unpack_swioma_SLQSOMADMAlertCallback_ind_t

### Macros

- #define LITEQMI_MAX_SWIOMA_STR_LEN 255

### Functions

- int pack_swioma_SLQSOMADMStartSession (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_swioma_SLQSOMADMStartSession_t reqParam)
- int unpack_swioma_SLQSOMADMStartSession (uint8_t ∗pResp, uint16_t respLen, unpack_swioma_SLQSOMADMStartSession_t ∗pOutput)
- int pack_swioma_SLQSOMADMCancelSession (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_swioma_SLQSOMADMCancelSession_t reqParam)
- int unpack_swioma_SLQSOMADMCancelSession (uint8_t ∗pResp, uint16_t respLen)
- int pack_swioma_SLQSOMADMGetSettings (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_swioma_SLQSOMADMGetSettings (uint8_t ∗pResp, uint16_t respLen, unpack_swioma_SLQSOMADMGetSettings_t ∗pOutput)
- int pack_swioma_SLQSOMADMSetSettings (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_swioma_SLQSOMADMSetSettings_t reqParam)
- int unpack_swioma_SLQSOMADMSetSettings (uint8_t ∗pResp, uint16_t respLen)
- int pack_swioma_SLQSOMADMSendSelection (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_swioma_SLQSOMADMSendSelection_t reqParam)
- int unpack_swioma_SLQSOMADMSendSelection (uint8_t ∗pResp, uint16_t respLen)
- int pack_swioma_SLQSOMADMGetSessionInfo (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen, pack_swioma_SLQSOMADMGetSessionInfo_t reqParam)
- int unpack_swioma_SLQSOMADMGetSessionInfo (uint8_t ∗pResp, uint16_t respLen, unpack_swioma_SLQSOMADMGetSessionInfo_t ∗pOutput)
- int pack_swioma_SLQSOMADMAlertCallback (pack_qmi_t ∗pCtx, uint8_t ∗pReqBuf, uint16_t ∗pLen)
- int unpack_swioma_SLQSOMADMAlertCallback (uint8_t ∗pResp, uint16_t respLen)
- int unpack_swioma_SLQSOMADMAlertCallback_ind (uint8_t ∗pResp, uint16_t respLen, unpack_swioma_SLQSOMADMAlertCallback_ind_t ∗pOutput)

## 9.38.1 Macro Definition Documentation

### 9.38.1.1 #define LITEQMI_MAX_SWIOMA_STR_LEN 255

## 9.38.2 Function Documentation

### 9.38.2.1 int pack_swioma_SLQSOMADMAlertCallback ( 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 SetSLQSOMADMAlertCallback

**Parameters**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.2 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**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On output, number of bytes actually packed |
| *reqParam* | [in]<br>• See pack_swioma_SLQSOMADMCancelSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.3 int pack_swioma_SLQSOMADMGetSessionInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swioma_SLQSOMADMGetSessionInfo_t *reqParam* )

Function to pack QMI command to return information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

**Parameters**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [in]<br>• See pack_swioma_SLQSOMADMGetSessionInfo_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.4 int pack_swioma_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 SLQSOMADMGet-Settings2

**Parameters**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |

| | |
|---|---|
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On output, number of bytes actually packed |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.5 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**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [in]<br>• See pack_swioma_SLQSOMADMSendSelection_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.6 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**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [in]<br>• See pack_swioma_SLQSOMADMSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.7 int pack_swioma_SLQSOMADMStartSession ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swioma_SLQSOMADMStartSession_t *reqParam* )

Function to pack Start OMA-DM session command This maps to SLQSOMADMStartSession2

**Parameters**

| | |
|---|---|
| *pCtx* | [out]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [in,out]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [in,out]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [in]<br>• See pack_swioma_SLQSOMADMStartSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.38.2.8 int unpack_swioma_SLQSOMADMAlertCallback ( 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 SetSLQSOMADMAlertCallback

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

- Please use eQMI_SWIOMA_EVENT_IND indication to identify this event from SWIOMA service read function

#### 9.38.2.9 int unpack_swioma_SLQSOMADMAlertCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMAlertCallback_ind_t * *pOutput* )

Function to unpack SWIOMADM alert indications This maps to SetSLQSOMADMAlertCallback

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |
| *pOutput* | [out]<br>• See unpack_swioma_SLQSOMADMAlertCallback_ind_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.38.2.10 int unpack_swioma_SLQSOMADMCancelSession ( uint8_t * *pResp*, uint16_t *respLen* )

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.38.2.11 int unpack_swioma_SLQSOMADMGetSessionInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMGetSessionInfo_t * *pOutput* )

Function to unpack information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |
| *pOutput* | [out]<br>• See unpack_swioma_SLQSOMADMGetSessionInfo_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.38.2.12 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**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |
| *pOutput* | [out]<br>• See unpack_swioma_SLQSOMADMGetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.38.2.13 int unpack_swioma_SLQSOMADMSendSelection ( uint8_t * *pResp*, uint16_t *respLen* )

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.38.2.14 int unpack_swioma_SLQSOMADMSetSettings ( uint8_t * *pResp*, uint16_t *respLen* )

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |

| | |
|---|---|
| *respLen* | [in]<br>• Length of pResp from modem |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.38.2.15 int unpack_swioma_SLQSOMADMStartSession ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMStartSession_t * *pOutput* )

Function to unpack Start OMA-DM session response from modem This maps to SLQSOMADMStartSession2

**Parameters**

| | |
|---|---|
| *pResp* | [in]<br>• Response from modem |
| *respLen* | [in]<br>• Length of pResp from modem |
| *pOutput* | [out]<br>• See unpack_swioma_SLQSOMADMStartSession_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.39 swiomaext.h File Reference

### Data Structures

- struct pack_swioma_SLQSOMADMStartSessionExt_t
- struct pack_swioma_SLQSOMADMCancelSessionExt_t
- struct pack_swioma_SLQSOMADMSetSettingsExt_t
- struct pack_swioma_SLQSOMADMSendSelectionExt_t

### Macros

- #define LITEQMI_MAX_SWIOMA_STR_LEN 255

### Typedefs

- typedef unpack_result_t unpack_swioma_SLQSOMADMStartSessionExt_t
- typedef unpack_result_t unpack_swioma_SLQSOMADMCancelSessionExt_t
- typedef unpack_result_t unpack_swioma_SLQSOMADMSetSettingsExt_t
- typedef unpack_result_t unpack_swioma_SLQSOMADMSendSelectionExt_t

### Functions

- int pack_swioma_SLQSOMADMStartSessionExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swioma_SLQSOMADMStartSessionExt_t reqParam)
- int unpack_swioma_SLQSOMADMStartSessionExt (uint8_t *pResp, uint16_t respLen, unpack_swioma_SLQSOMADMStartSessionExt_t *pOutput)
- int pack_swioma_SLQSOMADMCancelSessionExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swioma_SLQSOMADMCancelSessionExt_t reqParam)
- int unpack_swioma_SLQSOMADMCancelSessionExt (uint8_t *pResp, uint16_t respLen, unpack_swioma_SLQSOMADMCancelSessionExt_t *pOutput)
- int pack_swioma_SLQSOMADMSetSettingsExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swioma_SLQSOMADMSetSettingsExt_t reqParam)
- int unpack_swioma_SLQSOMADMSetSettingsExt (uint8_t *pResp, uint16_t respLen, unpack_swioma_SLQSOMADMSetSettingsExt_t *pOutput)
- int pack_swioma_SLQSOMADMSendSelectionExt (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_swioma_SLQSOMADMSendSelectionExt_t reqParam)
- int unpack_swioma_SLQSOMADMSendSelectionExt (uint8_t *pResp, uint16_t respLen, unpack_swioma_SLQSOMADMSendSelectionExt_t *pOutput)

## 9.39.1 Macro Definition Documentation

#### 9.39.1.1 #define LITEQMI_MAX_SWIOMA_STR_LEN 255

## 9.39.2 Typedef Documentation

#### 9.39.2.1 typedef unpack_result_t unpack_swioma_SLQSOMADMCancelSessionExt_t

#### 9.39.2.2 typedef unpack_result_t unpack_swioma_SLQSOMADMSendSelectionExt_t

#### 9.39.2.3 typedef unpack_result_t unpack_swioma_SLQSOMADMSetSettingsExt_t

#### 9.39.2.4 typedef unpack_result_t unpack_swioma_SLQSOMADMStartSessionExt_t

## 9.39.3 Function Documentation

#### 9.39.3.1 int pack_swioma_SLQSOMADMCancelSessionExt ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swioma_SLQSOMADMCancelSessionExt_t *reqParam* )

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSessionExt

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |

| | |
|---|---|
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [IN]<br>• See pack_swioma_SLQSOMADMCancelSessionExt_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.39.3.2 int pack_swioma_SLQSOMADMSendSelectionExt ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swioma_SLQSOMADMSendSelectionExt_t *reqParam* )

Function to pack OMA-DM send selection command This maps to SLQSOMADMSendSelectionExt

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [IN]<br>• See pack_swioma_SLQSOMADMSendSelectionExt_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.39.3.3 int pack_swioma_SLQSOMADMSetSettingsExt ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swioma_SLQSOMADMSetSettingsExt_t *reqParam* )

Function to pack OMA-DM set settings command This maps to SLQSOMADMSetSettingsExt

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [IN]<br>• See pack_swioma_SLQSOMADMSetSettings_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.39.3.4 int pack_swioma_SLQSOMADMStartSessionExt ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_swioma_SLQSOMADMStartSessionExt_t *reqParam* )

Function to pack Start OMA-DM session command This maps to SLQSOMADMStartSessionExt

**Parameters**

| | |
|---|---|
| *pCtx* | [OUT]<br>• See pack_qmi_t for more information |
| *pReqBuf* | [IN/OUT]<br>• Buffer for packed QMI command to be provided by the host application<br>• Minimum expected size is 2048 bytes |
| *pLen* | [IN/OUT]<br>• On input, size of pReqBuf<br>• On ouptut, number of bytes actually packed |
| *reqParam* | [IN]<br>• See pack_swioma_SLQSOMADMStartSessionExt_t for more information |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

Timeout: 20 seconds

#### 9.39.3.5 int unpack_swioma_SLQSOMADMCancelSessionExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMCancelSessionExt_t * *pOutput* )

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSessionExt

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.39.3.6 int unpack_swioma_SLQSOMADMSendSelectionExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMSendSelectionExt_t * *pOutput* )

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelectionExt

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.39.3.7 int unpack_swioma_SLQSOMADMSetSettingsExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMSetSettingsExt_t * *pOutput* )

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettingsExt

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.39.3.8 int unpack_swioma_SLQSOMADMStartSessionExt ( uint8_t * *pResp*, uint16_t *respLen*, unpack_swioma_SLQSOMADMStartSessionExt_t * *pOutput* )

Function to unpack Start OMA-DM session response from modem This maps to SLQSOMADMStartSessionExt

**Parameters**

| | |
|---|---|
| *pResp* | [IN]<br>• Response from modem |
| *respLen* | [IN]<br>• Length of pResp from modem |
| *pOutput* | [OUT]<br>• response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.40 switype_256bit.h File Reference

### Data Structures

- struct swi_uint256_t

### Macros

- #define SWI_UINT256_WORD_COUNT 16
- #define SWI_UINT256_BITS_PER_WORD 16
- #define SWI_UINT256_WORD_OFFSET(val) ((val) / SWI_UINT256_BITS_PER_WORD)
- #define SWI_UINT256_BIT_OFFSET(val) ((val) % SWI_UINT256_BITS_PER_WORD)
- #define SWI_UINT256_INT_VALUE {{0}}

### Functions

- void swi_uint256_set_bit (swi_uint256_t *pMask, uint8_t bit)
- void swi_uint256_clear_bit (swi_uint256_t *pMask, uint8_t bit)
- uint8_t swi_uint256_get_bit (swi_uint256_t Mask, uint8_t bit)
- void swi_uint256_print_mask (swi_uint256_t mask)

### 9.40.1 Macro Definition Documentation

#### 9.40.1.1 #define SWI_UINT256_BIT_OFFSET( *val* ) ((val) % SWI_UINT256_BITS_PER_WORD)

#### 9.40.1.2 #define SWI_UINT256_BITS_PER_WORD 16

#### 9.40.1.3 #define SWI_UINT256_INT_VALUE {{0}}

#### 9.40.1.4 #define SWI_UINT256_WORD_COUNT 16

#### 9.40.1.5 #define SWI_UINT256_WORD_OFFSET( *val* ) ((val) / SWI_UINT256_BITS_PER_WORD)

### 9.40.2 Function Documentation

#### 9.40.2.1 void swi_uint256_clear_bit ( swi_uint256_t * *pMask*, uint8_t *bit* )

Clear relevant bit in provided mask

**Parameters**

| | | |
|---|---|---|
| in | *pMask* | Pointer to bitmask |
| in | *bit* | Bit to clear |

#### 9.40.2.2 uint8_t swi_uint256_get_bit ( swi_uint256_t *Mask*, uint8_t *bit* )

Get relevant bit from provided mask

**Parameters**

| | | |
|---|---|---|
| in | *Mask* | Bit mask |
| in | *bit* | Bit to extract |

**Returns**

Bit value

- 0 - bit is not set
- 1 - bit is set

#### 9.40.2.3 void swi_uint256_print_mask ( swi_uint256_t *mask* )

Print the log mask to syslog. Use only for debugging

**Parameters**

| | | |
|---|---|---|
| in | *mask* | Bit mask |

#### 9.40.2.4 void swi_uint256_set_bit ( swi_uint256_t * *pMask*, uint8_t *bit* )

Set relevant bit in provided mask

**Parameters**

| | | |
|---|---|---|
| in | *pMask* | Pointer to bit mask |
| in | *bit* | Bit to set |

## 9.41 SWIWWANCMAPI.h File Reference

## 9.42 tmd.h File Reference

### Data Structures

- struct tmd_mitigationDevList
- struct unpack_tmd_SLQSTmdGetMitigationDevList_t
- struct pack_tmd_SLQSTmdGetMitigationLvl_t
- struct unpack_tmd_SLQSTmdGetMitigationLvl_t
- struct pack_tmd_SLQSTmdRegNotMitigationLvl_t
- struct unpack_tmd_SLQSTmdRegNotMitigationLvl_t
- struct pack_tmd_SLQSTmdDeRegNotMitigationLvl_t
- struct unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t
- struct unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t

### Macros

- #define TMD_MAX_DEV_LIST 255
- #define MAX_MITIGATION_DEV_LIST_LEN 255
- #define MAX_MITIGATION_DEV_ID_LEN 255

## Functions

- int pack_tmd_SLQSTmdGetMitigationDevList (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_tmd_SLQSTmdGetMitigationDevList (uint8_t *pResp, uint16_t respLen, unpack_tmd_SLQSTmdGetMitigationDevList_t *pOutput)
- int pack_tmd_SLQSTmdGetMitigationLvl (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_tmd_SLQSTmdGetMitigationLvl_t *reqArg)
- int unpack_tmd_SLQSTmdGetMitigationLvl (uint8_t *pResp, uint16_t respLen, unpack_tmd_SLQSTmdGetMitigationLvl_t *pOutput)
- int pack_tmd_SLQSTmdRegNotMitigationLvl (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_tmd_SLQSTmdRegNotMitigationLvl_t *reqArg)
- int unpack_tmd_SLQSTmdRegNotMitigationLvl (uint8_t *pResp, uint16_t respLen, unpack_tmd_SLQSTmdRegNotMitigationLvl_t *pOutput)
- int pack_tmd_SLQSTmdDeRegNotMitigationLvl (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_tmd_SLQSTmdDeRegNotMitigationLvl_t *reqArg)
- int unpack_tmd_SLQSTmdDeRegNotMitigationLvl (uint8_t *pResp, uint16_t respLen, unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t *pOutput)
- int unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t *pOutput)

### 9.42.1 Macro Definition Documentation

#### 9.42.1.1 #define MAX_MITIGATION_DEV_ID_LEN 255

#### 9.42.1.2 #define MAX_MITIGATION_DEV_LIST_LEN 255

#### 9.42.1.3 #define TMD_MAX_DEV_LIST 255

### 9.42.2 Function Documentation

#### 9.42.2.1 int pack_tmd_SLQSTmdDeRegNotMitigationLvl ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_tmd_SLQSTmdDeRegNotMitigationLvl_t * *reqArg* )

To DeRegister notification mitigation level pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *reqArg* | request argument. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.2 int pack_tmd_SLQSTmdGetMitigationDevList ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

To get mitigation device list pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.3 int pack_tmd_SLQSTmdGetMitigationLvl ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_tmd_SLQSTmdGetMitigationLvl_t * *reqArg* )

To get mitigation level pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *reqArg* | request argument. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.4 int pack_tmd_SLQSTmdRegNotMitigationLvl ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_tmd_SLQSTmdRegNotMitigationLvl_t * *reqArg* )

To Register notification mitigation level pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| out | *reqArg* | request argument. |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.5 int unpack_tmd_SLQSTmdDeRegNotMitigationLvl ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t** * *pOutput* )

To De-Register notification mitigation level unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.6 int unpack_tmd_SLQSTmdGetMitigationDevList ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_tmd_SLQSTmdGetMitigationDevList_t** * *pOutput* )

To mitigation device list unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.7 int unpack_tmd_SLQSTmdGetMitigationLvl ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_tmd_SLQSTmdGetMitigationLvl_t** * *pOutput* )

To get mitigation level unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.8 int unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t * *pOutput* )

Mitigiation Level Report Indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | indication unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.42.2.9 int unpack_tmd_SLQSTmdRegNotMitigationLvl ( uint8_t * *pResp*, uint16_t *respLen*, unpack_tmd_SLQSTmdRegNotMitigationLvl_t * *pOutput* )

To Register notification mitigation level unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.43 uim.h File Reference

### Data Structures

- struct uim_appStatus
- struct uim_slotInfo
- struct uim_cardStatus
- struct uim_hotSwapStatus
- struct unpack_uim_GetCardStatus_t
- struct uim_encryptedPIN1
- struct uim_remainingRetries

- struct uim_sessionInformation
- struct uim_verifyUIMPIN
- struct uim_unblockUIMPIN
- struct uim_cardResult
- struct uim_setPINProtection
- struct uim_changeUIMPIN
- struct uim_fileInfo
- struct uim_UIMSessionInformation
- struct uim_readTransparentInfo
- struct uim_readResult
- struct pack_uim_VerifyPin_t
- struct unpack_uim_VerifyPin_t
- struct pack_uim_UnblockPin_t
- struct unpack_uim_UnblockPin_t
- struct pack_uim_SetPinProtection_t
- struct unpack_uim_SetPinProtection_t
- struct pack_uim_ChangePin_t
- struct unpack_uim_ChangePin_t
- struct pack_uim_ReadTransparent_t
- struct unpack_uim_ReadTransparent_t
- struct pack_uim_SLQSUIMEventRegister_t
- struct unpack_uim_SLQSUIMEventRegister_t
- struct appStats
- struct slotInf
- struct unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t
- struct slot_t
- struct slots_t
- struct unpack_uim_SLQSUIMGetSlotsStatus_t
- struct pack_uim_SLQSUIMSwitchSlot_t
- struct unpack_uim_SetUimSlotStatusChangeCallback_ind_t
- struct pack_uim_SLQSUIMPowerUp_t
- struct pack_uim_SLQSUIMPowerDown_t
- struct pack_uim_SLQSUIMRefreshOK_t
- struct uim_registerRefresh
- struct pack_uim_SLQSUIMRefreshRegister_t
- struct pack_uim_SLQSUIMRefreshComplete_t
- struct uim_refreshevent
- struct pack_uim_SLQSUIMRefreshGetLastEvent_t
- struct unpack_uim_SLQSUIMRefreshGetLastEvent_t
- struct pack_uim_SLQSUIMGetFileAttributes_t
- struct uim_fileAttributes
- struct unpack_uim_SLQSUIMGetFileAttributes_t
- struct uim_depersonalizationInformation
- struct pack_uim_SLQSUIMDepersonalization_t
- struct unpack_uim_SLQSUIMDepersonalization_t
- struct uim_authenticationData
- struct pack_uim_SLQSUIMAuthenticate_t
- struct uim_authenticateResult
- struct unpack_uim_SLQSUIMAuthenticate_t
- struct uim_personalizationStatus
- struct pack_uim_SLQSUIMGetConfiguration_t
- struct unpack_uim_SLQSUIMGetConfiguration_t
- struct unpack_uim_SLQSUIMRefreshCallback_Ind_t

## Macros

- #define UIM_UINT8_MAX_STRING_SZ 255
- #define UIM_MAX_DESCRIPTION_LENGTH 255
- #define UIM_MAX_NO_OF_SLOTS 5
- #define UIM_MAX_NO_OF_APPLICATIONS 10
- #define MAX_NO_OF_SLOTS 5
- #define MAX_NO_OF_APPLICATIONS 10
- #define MAX_DESCRIPTION_LENGTH 255
- #define MAX_SLOTS_STATUS 255
- #define MAX_ICCID_LENGTH 255
- #define UIM_MAX_CONTENT_LENGTH 1024
- #define UIM_MAX_ACTIVE_PERS_FEATURES 12

## 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)

- int pack_uim_SLQSUIMReset (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_uim_SLQSUIMReset (uint8_t *pResp, uint16_t respLen)
- int pack_uim_SLQSUIMRefreshOK (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMRefreshOK_t *reqArg)
- int unpack_uim_SLQSUIMRefreshOK (uint8_t *pResp, uint16_t respLen)
- int pack_uim_SLQSUIMRefreshRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMRefreshRegister_t *reqArg)
- int unpack_uim_SLQSUIMRefreshRegister (uint8_t *pResp, uint16_t respLen)
- int pack_uim_SLQSUIMRefreshComplete (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMRefreshComplete_t *reqArg)
- int unpack_uim_SLQSUIMRefreshComplete (uint8_t *pResp, uint16_t respLen)
- int pack_uim_SLQSUIMRefreshGetLastEvent (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMRefreshGetLastEvent_t *reqArg)
- int unpack_uim_SLQSUIMRefreshGetLastEvent (uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMRefreshGetLastEvent_t *pOutput)
- int pack_uim_SLQSUIMGetFileAttributes (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMGetFileAttributes_t *reqArg)
- int unpack_uim_SLQSUIMGetFileAttributes (uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMGetFileAttributes_t *pOutput)
- int pack_uim_SLQSUIMDepersonalization (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMDepersonalization_t *reqArg)
- int unpack_uim_SLQSUIMDepersonalization (uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMDepersonalization_t *pOutput)
- int pack_uim_SLQSUIMAuthenticate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMAuthenticate_t *reqArg)
- int unpack_uim_SLQSUIMAuthenticate (uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMAuthenticate_t *pOutput)
- int pack_uim_SLQSUIMGetConfiguration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_uim_SLQSUIMGetConfiguration_t *reqArg)
- int unpack_uim_SLQSUIMGetConfiguration (uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMGetConfiguration_t *pOutput)
- int unpack_uim_SLQSUIMRefreshCallback_Ind (uint8_t *pResp, uint16_t respLen, unpack_uim_SLQSUIMRefreshCallback_Ind_t *pOutput)

## 9.43.1 Macro Definition Documentation

### 9.43.1.1 #define MAX_DESCRIPTION_LENGTH 255

### 9.43.1.2 #define MAX_ICCID_LENGTH 255

### 9.43.1.3 #define MAX_NO_OF_APPLICATIONS 10

### 9.43.1.4 #define MAX_NO_OF_SLOTS 5

### 9.43.1.5 #define MAX_SLOTS_STATUS 255

### 9.43.1.6 #define UIM_MAX_ACTIVE_PERS_FEATURES 12

### 9.43.1.7 #define UIM_MAX_CONTENT_LENGTH 1024

### 9.43.1.8 #define UIM_MAX_DESCRIPTION_LENGTH 255

### 9.43.1.9 #define UIM_MAX_NO_OF_APPLICATIONS 10

### 9.43.1.10 #define UIM_MAX_NO_OF_SLOTS 5

#### 9.43.1.11 #define UIM_UINT8_MAX_STRING_SZ 255

### 9.43.2 Function Documentation

#### 9.43.2.1 int pack_uim_ChangePin ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_ChangePin_t * *reqArg* )

Change Pin pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.2 int pack_uim_GetCardStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, void * *reqArg* )

Get Card Status pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.3 int pack_uim_ReadTransparent ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_ReadTransparent_t * *reqArg* )

SLQS ReadTransparent pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.4 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,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.5 int pack_uim_SLQSUIMAuthenticate ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMAuthenticate_t * *reqArg* )

Send a security command to the card pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.6 int pack_uim_SLQSUIMDepersonalization ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMDepersonalization_t * *reqArg* )

De-activates or unblocks the personalization on the phone pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.7 int pack_uim_SLQSUIMEventRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMEventRegister_t * *reqArg* )

UIM Status Change callback enable pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.8 int pack_uim_SLQSUIMGetConfiguration ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMGetConfiguration_t * *reqArg* )

Gets the modem configuration for the UIM module pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.9 int pack_uim_SLQSUIMGetFileAttributes ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMGetFileAttributes_t * *reqArg* )

Retrieves the file attributes for any EF(Elementary File) or DF(Dedicated File) in the card and provides access by the path pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.10 int pack_uim_SLQSUIMGetSlotsStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Pack get slots status.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.11 int pack_uim_SLQSUIMPowerDown ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMPowerDown_t * *reqArg* )

Powers down the card pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.12 int pack_uim_SLQSUIMPowerUp ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMPowerUp_t * *reqArg* )

Powers up the card pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.13 int pack_uim_SLQSUIMRefreshComplete ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshComplete_t * *reqArg* )

Enables the terminal response to be sent to the card When the client finished Refresh procedure pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.14 int pack_uim_SLQSUIMRefreshGetLastEvent ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshGetLastEvent_t * *reqArg* )

Retrieve the last refresh event pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.15 int pack_uim_SLQSUIMRefreshOK ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshOK_t * *reqArg* )

Enables the client to indicate whether it is OK to start the Refresh procedure pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.16 int pack_uim_SLQSUIMRefreshRegister ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_SLQSUIMRefreshRegister_t * *reqArg* )

Registers for file change notifications triggered by the card pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.17 int pack_uim_SLQSUIMReset ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Resets the issuing control points state kept by the service pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.43.2.18 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.43.2.19 int pack_uim_UnblockPin ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_UnblockPin_t * *reqArg* )

Unblock Pin pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.20 int pack_uim_VerifyPin ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_uim_VerifyPin_t * *reqArg* )

Verify Pin Status pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.21 int unpack_uim_ChangePin ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_ChangePin_t * *pOutput* )

Change Pin unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.22 int unpack_uim_GetCardStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_GetCardStatus_t * *pOutput* )

Get Card Status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.23 int unpack_uim_ReadTransparent ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_uim_ReadTransparent_t** * *pOutput* )

SLQS ReadTransparent unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.24 int unpack_uim_SetPinProtection ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_uim_SetPinProtection_t** * *pOutput* )

Set Pin Protection unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.25 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 | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

use pack_uim_SLQSUIMEventRegister to subscribe

#### 9.43.2.26 int unpack_uim_SLQSUIMAuthenticate ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMAuthenticate_t * *pOutput* )

Unpack the status code received from the card when card responded to the read request

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.27 int unpack_uim_SLQSUIMDepersonalization ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMDepersonalization_t * *pOutput* )

De-activates or unblocks the personalization on the phone unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.28 int unpack_uim_SLQSUIMEventRegister ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMEventRegister_t * *pOutput* )

UIM Status Change callback enable unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.29 int unpack_uim_SLQSUIMGetConfiguration ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMGetConfiguration_t * *pOutput* )

Gets the modem configuration for the UIM module unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.30 int unpack_uim_SLQSUIMGetFileAttributes ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMGetFileAttributes_t * *pOutput* )

Retrieves the file attributes for any EF(Elementary File) or DF(Dedicated File) in the card and provides access by the path unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.31 int unpack_uim_SLQSUIMGetSlotsStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMGetSlotsStatus_t * *pOutput* )

get slot status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.32 int unpack_uim_SLQSUIMPowerDown ( uint8_t * *pResp*, uint16_t *respLen* )

Powers down the card unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.33 int unpack_uim_SLQSUIMPowerUp ( uint8_t * *pResp*, uint16_t *respLen* )

Powers up the card unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.34 int unpack_uim_SLQSUIMRefreshCallback_Ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMRefreshCallback_Ind_t * *pOutput* )

Unpack UIM refresh event indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.35 int unpack_uim_SLQSUIMRefreshComplete ( uint8_t * *pResp*, uint16_t *respLen* )

Enables the terminal response to be sent to the card When the client finished Refresh procedure unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.36 int unpack_uim_SLQSUIMRefreshGetLastEvent ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMRefreshGetLastEvent_t * *pOutput* )

Retrieve the last refresh event unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.37 int unpack_uim_SLQSUIMRefreshOK ( uint8_t * *pResp*, uint16_t *respLen* )

Enables the client to indicate whether it is OK to start the Refresh procedure unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.38 int unpack_uim_SLQSUIMRefreshRegister ( uint8_t * *pResp*, uint16_t *respLen* )

Registers for file change notifications triggered by the card unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.39 int unpack_uim_SLQSUIMReset ( uint8_t * *pResp*, uint16_t *respLen* )

Resets the issuing control points state kept by the service unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.40 int unpack_uim_SLQSUIMSetStatusChangeCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t * *pOutput* )

UIM Status Change indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

use pack_uim_SLQSUIMEventRegister to subscribe

#### 9.43.2.41 int unpack_uim_SLQSUIMSwitchSlot ( uint8_t * *pResp*, uint16_t *respLen* )

switch slot unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.42 int unpack_uim_UnblockPin ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_uim_UnblockPin_t** * *pOutput* )

Unblock Pin unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.43.2.43 int unpack_uim_VerifyPin ( uint8_t * *pResp*, uint16_t *respLen*, unpack_uim_VerifyPin_t * *pOutput* )

Verify Pin unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | qmi response length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.44 voice.h File Reference

### Data Structures

- struct voice_USSInfo
- struct pack_voice_OriginateUSSD_t
- struct pack_voice_AnswerUSSD_t
- struct voice_UUSInfo
- struct voice_CUGInfo
- struct voice_calledPartySubAdd
- struct pack_voice_SLQSVoiceDialCall_t
- struct voice_alphaIDInfo
- struct voice_ccSUPSType
- struct unpack_voice_SLQSVoiceDialCall_t
- struct pack_voice_SLQSVoiceEndCall_t
- struct unpack_voice_SLQSVoiceEndCall_t
- struct voice_callFwdTypeAndPlan
- struct pack_voice_SLQSVoiceSetSUPSService_t
- struct unpack_voice_SLQSVoiceSetSUPSService_t
- struct voice_airTimer
- struct voice_roamTimer
- struct voice_prefVoiceSO
- struct pack_voice_SLQSVoiceSetConfig_t
- struct unpack_voice_SLQSVoiceSetConfig_t
- struct pack_voice_SLQSVoiceAnswerCall_t
- struct unpack_voice_SLQSVoiceAnswerCall_t
- struct voice_CLIRResp
- struct unpack_voice_SLQSVoiceGetCLIR_t
- struct voice_CLIPResp
- struct unpack_voice_SLQSVoiceGetCLIP_t
- struct pack_voice_SLQSVoiceGetCallWaiting_t
- struct unpack_voice_SLQSVoiceGetCallWaiting_t
- struct pack_voice_SLQSVoiceGetCallBarring_t
- struct unpack_voice_SLQSVoiceGetCallBarring_t
- struct pack_voice_SLQSVoiceGetCallForwardingStatus_t
- struct voice_callFWInfo

- struct voice_callFWExtInfo
- struct voice_getCallFWInfo
- struct voice_getCallFWExtInfo
- struct unpack_voice_SLQSVoiceGetCallForwardingStatus_t
- struct pack_voice_SLQSVoiceSetCallBarringPassword_t
- struct unpack_voice_SLQSVoiceSetCallBarringPassword_t
- struct pack_voice_SLQSVoiceGetCallInfo_t
- struct voice_callInfo
- struct voice_remotePartyNum
- struct voice_remotePartyName
- struct voice_connectNumInfo
- struct voice_diagInfo
- struct unpack_voice_SLQSVoiceGetCallInfo_t
- struct voice_getAllCallInformation
- struct voice_getAllCallRmtPtyNum
- struct voice_getAllCallRmtPtyName
- struct voice_allCallsUUSInfo
- struct voice_allCallsAlphaIDInfo
- struct voice_allCallsDiagInfo
- struct voice_peerNumberInfo
- struct voice_arrCallInfo
- struct voice_arrRemotePartyNum
- struct voice_arrRemotePartyName
- struct voice_arrAlertingType
- struct voice_arrUUSInfo
- struct voice_arrSvcOption
- struct voice_arrCallEndReason
- struct voice_arrAlphaID
- struct voice_arrConnectPartyNum
- struct voice_arrDiagInfo
- struct voice_arrCalledPartyNum
- struct voice_arrRedirPartyNum
- struct voice_arrAlertingPattern
- struct unpack_voice_SLQSVoiceGetAllCallInfo_t
- struct pack_voice_SLQSVoiceManageCalls_t
- struct unpack_voice_SLQSVoiceManageCalls_t
- struct voice_burstDTMFInfo
- struct voice_DTMFLengths
- struct pack_voice_SLQSVoiceBurstDTMF_t
- struct unpack_voice_SLQSVoiceBurstDTMF_t
- struct pack_voice_SLQSVoiceStartContDTMF_t
- struct unpack_voice_SLQSVoiceStartContDTMF_t
- struct pack_voice_SLQSVoiceStopContDTMF_t
- struct unpack_voice_SLQSVoiceStopContDTMF_t
- struct pack_voice_SLQSVoiceSendFlash_t
- struct unpack_voice_SLQSVoiceSendFlash_t
- struct pack_voice_SLQSVoiceSetPreferredPrivacy_t
- struct pack_voice_SLQSVoiceIndicationRegister_t
- struct pack_voice_SLQSVoiceGetConfig_t
- struct voice_curAMRConfig
- struct unpack_voice_SLQSVoiceGetConfig_t
- struct pack_voice_SLQSVoiceOrigUSSDNoWait_t
- struct pack_voice_SLQSVoiceBindSubscription_t
- struct pack_voice_SLQSVoiceALSSetLineSwitching_t
- struct pack_voice_SLQSVoiceALSSelectLine_t

- struct voice_COLPResp
- struct unpack_voice_SLQSVoiceGetCOLP_t
- struct voice_COLRResp
- struct unpack_voice_SLQSVoiceGetCOLR_t
- struct voice_CNAPResp
- struct unpack_voice_SLQSVoiceGetCNAP_t
- struct pack_voice_SLQSOriginateUSSD_t
- struct unpack_voice_SLQSOriginateUSSD_t
- struct voice_USSDNotificationNetworkInfo
- struct unpack_voice_USSDNotificationCallback_ind_t
- struct voice_SUPSInfo
- struct voice_newPwdData
- struct unpack_voice_SLQSVoiceSUPSCallback_ind_t
- struct unpack_voice_allCallStatusCallback_ind_t
- struct unpack_voice_voicePrivacyChangeCallback_ind_t
- struct voice_DTMFInfo
- struct unpack_voice_DTMFEventCallback_ind_t
- struct voice_ECTNum
- struct unpack_voice_SUPSNotificationCallback_ind_t
- struct unpack_voice_OTASPStatusCallback_ind_t
- struct voice_signalInfo
- struct voice_callerIDInfo
- struct voice_callingPartyInfo
- struct voice_calledPartyInfo
- struct voice_redirNumInfo
- struct voice_NSSAudioCtrl
- struct voice_lineCtrlInfo
- struct voice_extDispRecInfo
- struct unpack_voice_VoiceInfoRecCallback_ind_t

## Macros

- #define MAXVOICEUSSDLENGTH 182
- #define MAX_VOICE_CALL_NO_LEN 81
- #define MAX_VOICE_DESCRIPTION_LENGTH 255
- #define VOICE_MAX_NO_OF_CALLS 20
- #define BARRING_PASSWORD_LENGTH 4

## Enumerations

- enum liteServiceClassInformation {
  LITE_VOICE_SUPS_SRV_CLASS_NONE = 0x00,
  LITE_VOICE_SUPS_SRV_CLASS_VOICE = 0x01,
  LITE_VOICE_SUPS_SRV_CLASS_DATA = 0x02,
  LITE_VOICE_SUPS_SRV_CLASS_FAX = 0x04,
  LITE_VOICE_SUPS_SRV_CLASS_SMS = 0x08,
  LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC = 0x10,
  LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC = 0x20,
  LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS = 0x40,
  LITE_VOICE_SUPS_SRV_CLASS_PADACCESS = 0x80 }

## Functions

- int pack_voice_OriginateUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_OriginateUSSD_t *reqArg)
- int unpack_voice_OriginateUSSD (uint8_t *pResp, uint16_t respLen)
- int pack_voice_AnswerUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_AnswerUSSD_t *reqArg)
- int unpack_voice_AnswerUSSD (uint8_t *pResp, uint16_t respLen)
- int pack_voice_CancelUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_CancelUSSD (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceDialCall (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceDialCall_t *pReq)
- int unpack_voice_SLQSVoiceDialCall (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceDialCall_t *pOutput)
- int pack_voice_SLQSVoiceEndCall (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceEndCall_t *pReq)
- int unpack_voice_SLQSVoiceEndCall (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceEndCall_t *pOutput)
- int pack_voice_SLQSVoiceSetSUPSService (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetSUPSService_t *pReq)
- int unpack_voice_SLQSVoiceSetSUPSService (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetSUPSService_t *pOutput)
- int pack_voice_SLQSVoiceSetConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetConfig_t *pReq)
- int unpack_voice_SLQSVoiceSetConfig (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetConfig_t *pOutput)
- int pack_voice_SLQSVoiceAnswerCall (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceAnswerCall_t *pReq)
- int unpack_voice_SLQSVoiceAnswerCall (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceAnswerCall_t *pOutput)
- int pack_voice_SLQSVoiceGetCLIR (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCLIR (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCLIR_t *pOutput)
- int pack_voice_SLQSVoiceGetCLIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCLIP (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCLIP_t *pOutput)
- int pack_voice_SLQSVoiceGetCallWaiting (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallWaiting_t *pReq)
- int unpack_voice_SLQSVoiceGetCallWaiting (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallWaiting_t *pOutput)
- int pack_voice_SLQSVoiceGetCallBarring (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallBarring_t *pReq)
- int unpack_voice_SLQSVoiceGetCallBarring (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallBarring_t *pOutput)
- int pack_voice_SLQSVoiceGetCallForwardingStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallForwardingStatus_t *pReq)
- int unpack_voice_SLQSVoiceGetCallForwardingStatus (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallForwardingStatus_t *pOutput)
- int pack_voice_SLQSVoiceSetCallBarringPassword (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetCallBarringPassword_t *pReq)
- int unpack_voice_SLQSVoiceSetCallBarringPassword (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetCallBarringPassword_t *pOutput)
- int pack_voice_SLQSVoiceGetCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetCallInfo_t *pReq)
- int unpack_voice_SLQSVoiceGetCallInfo (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCallInfo_t *pOutput)

- int pack_voice_SLQSVoiceGetAllCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetAllCallInfo (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetAllCallInfo_t *pOutput)
- int pack_voice_SLQSVoiceManageCalls (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceManageCalls_t *pReq)
- int unpack_voice_SLQSVoiceManageCalls (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceManageCalls_t *pOutput)
- int pack_voice_SLQSVoiceBurstDTMF (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceBurstDTMF_t *pReq)
- int unpack_voice_SLQSVoiceBurstDTMF (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceBurstDTMF_t *pOutput)
- int pack_voice_SLQSVoiceStartContDTMF (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceStartContDTMF_t *pReq)
- int unpack_voice_SLQSVoiceStartContDTMF (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceStartContDTMF_t *pOutput)
- int pack_voice_SLQSVoiceStopContDTMF (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceStopContDTMF_t *pReq)
- int unpack_voice_SLQSVoiceStopContDTMF (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceStopContDTMF_t *pOutput)
- int pack_voice_SLQSVoiceSendFlash (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSendFlash_t *pReq)
- int unpack_voice_SLQSVoiceSendFlash (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSendFlash_t *pOutput)
- int pack_voice_SLQSVoiceSetPreferredPrivacy (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceSetPreferredPrivacy_t *reqArg)
- int unpack_voice_SLQSVoiceSetPreferredPrivacy (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceIndicationRegister_t *reqArg)
- int unpack_voice_SLQSVoiceIndicationRegister (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceGetConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceGetConfig_t *reqArg)
- int unpack_voice_SLQSVoiceGetConfig (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetConfig_t *pOutput)
- int pack_voice_SLQSVoiceOrigUSSDNoWait (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceOrigUSSDNoWait_t *reqArg)
- int unpack_voice_SLQSVoiceOrigUSSDNoWait (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceBindSubscription (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceBindSubscription_t *reqArg)
- int unpack_voice_SLQSVoiceBindSubscription (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceALSSetLineSwitching (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceALSSetLineSwitching_t *reqArg)
- int unpack_voice_SLQSVoiceALSSetLineSwitching (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceALSSelectLine (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSVoiceALSSelectLine_t *reqArg)
- int unpack_voice_SLQSVoiceALSSelectLine (uint8_t *pResp, uint16_t respLen)
- int pack_voice_SLQSVoiceGetCOLP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCOLP (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCOLP_t *pOutput)
- int pack_voice_SLQSVoiceGetCOLR (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCOLR (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCOLR_t *pOutput)
- int pack_voice_SLQSVoiceGetCNAP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_voice_SLQSVoiceGetCNAP (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCNAP_t *pOutput)
- int pack_voice_SLQSOriginateUSSD (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_voice_SLQSOriginateUSSD_t *reqArg)

- int unpack_voice_SLQSOriginateUSSD (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSOriginateUSSD_t *pOutput)
- int unpack_voice_USSDNotificationCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_USSDNotificationCallback_ind_t *pOutput)
- int unpack_voice_SLQSVoiceSUPSCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_SLQSVoiceSUPSCallback_ind_t *pOutput)
- int unpack_voice_allCallStatusCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_allCallStatusCallback_ind_t *pOutput)
- int unpack_voice_voicePrivacyChangeCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_voicePrivacyChangeCallback_ind_t *pOutput)
- int unpack_voice_DTMFEventCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_DTMFEventCallback_ind_t *pOutput)
- int unpack_voice_SUPSNotificationCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_SUPSNotificationCallback_ind_t *pOutput)
- int unpack_voice_OTASPStatusCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_OTASPStatusCallback_ind_t *pOutput)
- int unpack_voice_VoiceInfoRecCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_voice_VoiceInfoRecCallback_ind_t *pOutput)

### 9.44.1 Detailed Description

### 9.44.2 S1

- 3GPP Mobile Radio Interface Layer 3 Specification : Core Network Protocols; Stage 3 (Release 7)
- 3GPP TS 24.008 V7.0.0 (2005-06)

### 9.44.3 Macro Definition Documentation

#### 9.44.3.1 #define BARRING_PASSWORD_LENGTH 4

#### 9.44.3.2 #define MAX_VOICE_CALL_NO_LEN 81

#### 9.44.3.3 #define MAX_VOICE_DESCRIPTION_LENGTH 255

#### 9.44.3.4 #define MAXVOICEUSSDLENGTH 182

#### 9.44.3.5 #define VOICE_MAX_NO_OF_CALLS 20

### 9.44.4 Enumeration Type Documentation

#### 9.44.4.1 enum liteServiceClassInformation

Service Class information

**Enumerator**

***LITE_VOICE_SUPS_SRV_CLASS_NONE***
***LITE_VOICE_SUPS_SRV_CLASS_VOICE***
***LITE_VOICE_SUPS_SRV_CLASS_DATA***
***LITE_VOICE_SUPS_SRV_CLASS_FAX***
***LITE_VOICE_SUPS_SRV_CLASS_SMS***
***LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC***
***LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC***
***LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS***
***LITE_VOICE_SUPS_SRV_CLASS_PADACCESS***

### 9.44.5 Function Documentation

#### 9.44.5.1 int pack_voice_AnswerUSSD ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_AnswerUSSD_t * reqArg )

Responds to a USSD request from the network pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.2 int pack_voice_CancelUSSD ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

Cancels an in-progress USSD operation pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.3 int pack_voice_OriginateUSSD ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_OriginateUSSD_t * reqArg )

Initiates a USSD operation pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.4 int pack_voice_SLQSOriginateUSSD ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSOriginateUSSD_t * *reqArg* )

Initiates a USSD session pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.5 int pack_voice_SLQSVoiceALSSelectLine ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceALSSelectLine_t * *reqArg* )

allows the user to select the preferred line pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.6 int pack_voice_SLQSVoiceALSSetLineSwitching ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceALSSetLineSwitching_t * *reqArg* )

sets the line switch setting on the card pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.7 int pack_voice_SLQSVoiceAnswerCall ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceAnswerCall_t * *pReq* )

Answers an incoming voice call pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.8 int pack_voice_SLQSVoiceBindSubscription ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceBindSubscription_t * *reqArg* )

binds a subscription type to a specific voice client ID pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.9 int pack_voice_SLQSVoiceBurstDTMF ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceBurstDTMF_t * *pReq* )

Sends a burst Dual-Tone Multi frequency (DTMF) pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.10 int pack_voice_SLQSVoiceDialCall ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceDialCall_t * *pReq* )

Originates a voice call (MO call) pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.11 int pack_voice_SLQSVoiceEndCall ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceEndCall_t * *pReq* )

Ends a voice call pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.12 int pack_voice_SLQSVoiceGetAllCallInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

information associated with all the calls originating or terminating pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.13 int pack_voice_SLQSVoiceGetCallBarring ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetCallBarring_t * *pReq* )

Status of Call Barring Supplementary Service pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.14 int pack_voice_SLQSVoiceGetCallForwardingStatus ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetCallForwardingStatus_t * *pReq* )

Status of Call Forwarding Supplementary Service pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.15 int pack_voice_SLQSVoiceGetCallInfo ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetCallInfo_t * *pReq* )

information associated with a call pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.16 int pack_voice_SLQSVoiceGetCallWaiting ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceGetCallWaiting_t * *pReq* )

Status of Call Waiting Supplementary Service pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.17 int pack_voice_SLQSVoiceGetCLIP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

status of the Calling Line Identification Presentation (CLIP) pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.18 int pack_voice_SLQSVoiceGetCLIR ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

status of the Calling Line Identification Restriction (CLIR) pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.19 int pack_voice_SLQSVoiceGetCNAP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

status of the Calling Name Presentation(CNAP) pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.20 int pack_voice_SLQSVoiceGetCOLP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

status of the Connected Line Identification Presentation pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.21 int pack_voice_SLQSVoiceGetCOLR ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )

status of the Connected Line Identification Restriction pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.22 int pack_voice_SLQSVoiceGetConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetConfig_t * reqArg )

retrieves various configuration parameters that control the modem behavior pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.23 int pack_voice_SLQSVoiceIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceIndicationRegister_t * reqArg )

Sets the registration state for different QMI_VOICE indications pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.24 int pack_voice_SLQSVoiceManageCalls ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceManageCalls_t * pReq )

Manages the calls by using the supplementary service pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.25 int pack_voice_SLQSVoiceOrigUSSDNoWait ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceOrigUSSDNoWait_t * reqArg )

initiates a USSD operation No Wait pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.26 int pack_voice_SLQSVoiceSendFlash ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSendFlash_t * *pReq* )

sends a simple flash message pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.27 int pack_voice_SLQSVoiceSetCallBarringPassword ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetCallBarringPassword_t * *pReq* )

Sets a Call Barring Password pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.28 int pack_voice_SLQSVoiceSetConfig ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetConfig_t * *pReq* )

configuration parameters that control the modem behavior pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.44.5.29 int pack_voice_SLQSVoiceSetPreferredPrivacy ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetPreferredPrivacy_t * *reqArg* )

sets the voice privacy preference pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.44.5.30 int pack_voice_SLQSVoiceSetSUPSService ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_voice_SLQSVoiceSetSUPSService_t * *pReq* )

call-independent supplementary services pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.31 int pack_voice_SLQSVoiceStartContDTMF ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceStartContDTMF_t * pReq )

Starts a continuous DTMF pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.32 int pack_voice_SLQSVoiceStopContDTMF ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceStopContDTMF_t * pReq )

Stops a continuous DTMF pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.33 int unpack_voice_allCallStatusCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_voice_allCallStatusCallback_ind_t * pOutput )

Unpack all call status callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | all call status callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.34 int unpack_voice_AnswerUSSD ( uint8_t * *pResp*, uint16_t *respLen* )

Responds to a USSD request from the network unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.35 int unpack_voice_CancelUSSD ( uint8_t * *pResp*, uint16_t *respLen* )

Cancels an in-progress USSD operation unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.36 int unpack_voice_DTMFEventCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_DTMFEventCallback_ind_t * *pOutput* )

Unpack DTMF event callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | DTMF event callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.37 int unpack_voice_OriginateUSSD ( uint8_t * *pResp*, uint16_t *respLen* )

Initiates a USSD operation unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.38 int unpack_voice_OTASPStatusCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_OTASPStatusCallback_ind_t * *pOutput* )

Unpack voice OTASP status callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | voice OTASP status callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.39 int unpack_voice_SLQSOriginateUSSD ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSOriginateUSSD_t * *pOutput* )

Initiates a USSD session unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.40 int unpack_voice_SLQSVoiceALSSelectLine ( uint8_t * *pResp*, uint16_t *respLen* )

allows the user to select the preferred line unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.41 int unpack_voice_SLQSVoiceALSSetLineSwitching ( uint8_t * *pResp*, uint16_t *respLen* )

sets the line switch setting on the card unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.42 int unpack_voice_SLQSVoiceAnswerCall ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceAnswerCall_t * *pOutput* )

Answers an incoming voice call unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.43 int unpack_voice_SLQSVoiceBindSubscription ( uint8_t * *pResp*, uint16_t *respLen* )

binds a subscription type to a specific voice client ID unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.44 int unpack_voice_SLQSVoiceBurstDTMF ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceBurstDTMF_t * *pOutput* )

Sends a burst Dual-Tone Multi frequency (DTMF) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.45 int unpack_voice_SLQSVoiceDialCall ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceDialCall_t * *pOutput* )

Originates a voice call (MO call) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.46 int unpack_voice_SLQSVoiceEndCall ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_voice_SLQSVoiceEndCall_t** * *pOutput* )

Ends a voice call unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.47 int unpack_voice_SLQSVoiceGetAllCallInfo ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_voice_SLQSVoiceGetAllCallInfo_t** * *pOutput* )

information associated with all the calls originating or terminating unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.48 int unpack_voice_SLQSVoiceGetCallBarring ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_voice_SLQSVoiceGetCallBarring_t** * *pOutput* )

Status of Call Barring Supplementary Service unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.49 int unpack_voice_SLQSVoiceGetCallForwardingStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCallForwardingStatus_t * *pOutput* )

Status of Call Forwarding Supplementary Service unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.50 int unpack_voice_SLQSVoiceGetCallInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCallInfo_t * *pOutput* )

information associated with a call unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.51 int unpack_voice_SLQSVoiceGetCallWaiting ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCallWaiting_t * *pOutput* )

Status of Call Waiting Supplementary Service unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.52 int unpack_voice_SLQSVoiceGetCLIP ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCLIP_t * *pOutput* )

status of the Calling Line Identification Presentation (CLIP) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.53 int unpack_voice_SLQSVoiceGetCLIR ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCLIR_t * *pOutput* )

status of the Calling Line Identification Restriction (CLIR) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.54 int unpack_voice_SLQSVoiceGetCNAP ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCNAP_t * *pOutput* )

status of the Calling Name Presentation(CNAP) unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.55 int unpack_voice_SLQSVoiceGetCOLP ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCOLP_t * *pOutput* )

status of the Connected Line Identification Presentation unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.56 int unpack_voice_SLQSVoiceGetCOLR ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetCOLR_t * *pOutput* )

status of the Connected Line Identification Restriction unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.57 int unpack_voice_SLQSVoiceGetConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceGetConfig_t * *pOutput* )

retrieves various configuration parameters that control the modem behavior unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.58 int unpack_voice_SLQSVoiceIndicationRegister ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the registration state for different QMI_VOICE indications unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.59 int unpack_voice_SLQSVoiceManageCalls ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceManageCalls_t * *pOutput* )

Manages the calls by using the supplementary service unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.60 int unpack_voice_SLQSVoiceOrigUSSDNoWait ( uint8_t * *pResp*, uint16_t *respLen* )

initiates a USSD operation No Wait unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.61 int unpack_voice_SLQSVoiceSendFlash ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceSendFlash_t * *pOutput* )

sends a simple flash message unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.62 int unpack_voice_SLQSVoiceSetCallBarringPassword ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceSetCallBarringPassword_t * *pOutput* )

Sets a Call Barring Password unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.63 int unpack_voice_SLQSVoiceSetConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceSetConfig_t * *pOutput* )

configuration parameters that control the modem behavior unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.64 int unpack_voice_SLQSVoiceSetPreferredPrivacy ( uint8_t * *pResp*, uint16_t *respLen* )

sets the voice privacy preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.65 int unpack_voice_SLQSVoiceSetSUPSService ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceSetSUPSService_t * *pOutput* )

call-independent supplementary services unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.66 int unpack_voice_SLQSVoiceStartContDTMF ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceStartContDTMF_t * *pOutput* )

Starts a continuous DTMF unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.44.5.67 int unpack_voice_SLQSVoiceStopContDTMF ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceStopContDTMF_t * *pOutput* )

Stops a continuous DTMF unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**Note**

return eQCWWAN_ERR_NULL_TLV then callID value is invalid.

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.44.5.68 int unpack_voice_SLQSVoiceSUPSCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SLQSVoiceSUPSCallback_ind_t * *pOutput* )

Unpack voice sups callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | voice susp callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

### 9.44.5.69 int unpack_voice_SUPSNotificationCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_SUPSNotificationCallback_ind_t * *pOutput* )

Unpack SUPS notification callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | SUPS notification callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.70 int unpack_voice_USSDNotificationCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_USSDNotificationCallback_ind_t * *pOutput* )

Unpack USSD notification callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | USSD notification unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.71 int unpack_voice_VoiceInfoRecCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_VoiceInfoRecCallback_ind_t * *pOutput* )

Unpack SUPS notification callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | SUPS notification callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.44.5.72 int unpack_voice_voicePrivacyChangeCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_voice_voicePrivacyChangeCallback_ind_t * *pOutput* )

Unpack voice privacy change callback indication.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | voice privacy change callback unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

## 9.45 wds.h File Reference

### Data Structures

- struct LibPackQosClassID
- struct LibPackTFTIDParams
- struct LibPackGPRSRequestedQoS
- struct LibPackUMTSQoS
- struct LibPackUMTSReqQoSSigInd
- struct pack_wds_SLQSStartDataSession_t
- struct unpack_wds_SLQSStartDataSession_t
- struct unpack_wds_SLQSSetPacketSrvStatusCallback_t
- struct pack_wds_SLQSStopDataSession_t
- struct wds_ProfileIdentifier
- struct wds_GPRSQoS
- struct wds_PCSCFIPv4ServerAddressList
- struct wds_PCSCFFQDNAddress
- struct wds_PCSCFFQDNAddressList
- struct wds_Domain
- struct wds_DomainNameList
- struct wds_IPV6AddressInfo
- struct wds_IPV6GWAddressInfo
- struct unpack_wds_SLQSGetRuntimeSettings_t
- struct wds_currNetworkInfo
- struct unpack_wds_SLQSSetWdsEventCallback_ind_t
- struct pack_wds_SLQSSetWdsEventCallback_t
- struct pack_wds_SLQSGetRuntimeSettings_t
- struct wds_UMTSMinQoS
- struct LibPackprofile_3GPP
- struct LibPackprofile_3GPP2
- union wds_profileInfo
- struct pack_wds_SLQSCreateProfile_t
- struct PackCreateProfileOut
- struct unpack_wds_SLQSCreateProfile_t
- struct pack_wds_SLQSModifyProfile_t

- struct unpack_wds_SLQSModifyProfile_t
- struct pack_wds_SLQSGetProfileSettings_t
- struct LibpackProfile3GPP
- struct LibpackProfile3GPP2
- union unpackWdsProfileParam
- struct UnPackGetProfileSettingOut
- struct unpack_wds_SLQSGetProfileSettings_t
- struct unpack_wds_GetSessionState_t
- struct pack_wds_GetDefaultProfile_t
- struct unpack_wds_GetDefaultProfile_t
- struct unpack_wds_GetConnectionRate_t
- struct pack_wds_GetPacketStatus_t
- struct unpack_wds_GetPacketStatus_t
- struct unpack_wds_GetSessionDuration_t
- struct pack_wds_GetSessionDuration_t
- struct unpack_wds_GetDormancyState_t
- struct pack_wds_GetDormancyState_t
- struct pack_wds_SLQSDeleteProfile_t
- struct unpack_wds_SLQSDeleteProfile_t
- struct pack_wds_SetDefaultProfile_t
- struct unpack_wds_SLQSGet3GPPConfigItem_t
- struct pack_wds_SLQSSet3GPPConfigItem_t
- struct unpack_wds_GetMobileIP_t
- struct pack_wds_GetMobileIP_t
- struct pack_wds_GetMobileIPProfile_t
- struct unpack_wds_GetMobileIPProfile_t
- struct currNetworkInfo
- struct unpack_wds_SLQSGetCurrDataSystemStat_t
- struct pack_wds_SLQSGetCurrDataSystemStat_t
- struct unpack_wds_GetLastMobileIPError_t
- struct pack_wds_GetLastMobileIPError_t
- struct rmTrasnferStaticsReq
- struct pack_wds_RMSetTransferStatistics_t
- struct unpack_wds_RMSetTransferStatistics_t
- struct pack_wds_SetMobileIPProfile_t
- struct unpack_wds_SetMobileIPProfile_t
- struct pack_wds_SLQSWdsSwiPDPRuntimeSettings_t
- struct ipv6AddressInfo
- struct unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t
- struct wds_transferStatInd
- struct pack_wds_SLQSGetDUNCallInfo_t
- struct connectionStatus
- struct dunchannelRate
- struct unpack_wds_SLQSGetDUNCallInfo_t
- struct qmiWDSDataBearerTechnology
- struct unpack_wds_SLQSGetDataBearerTechnology_t
- struct pack_wds_SLQSGetDataBearerTechnology_t
- struct pack_wds_SLQSSetIPFamilyPreference_t
- struct unpack_wds_SLQSSetIPFamilyPreference_t
- struct pack_wds_SetDefaultProfileNum_t
- struct pack_wds_GetDefaultProfileNum_t
- struct unpack_wds_GetDefaultProfileNum_t
- struct wdsDhcpv4ProfileId
- struct wdsDhcpv4HwConfig
- struct wdsDhcpv4Option

- struct wdsDhcpv4OptionList
- struct pack_wds_SLQSSGetDHCPv4ClientConfig_t
- struct unpack_wds_SLQSSGetDHCPv4ClientConfig_t
- struct pack_wds_GetPacketStatistics_t
- struct unpack_wds_GetPacketStatistics_t
- struct unpack_wds_GetByteTotals_t
- struct unpack_wds_SLQSGetCurrentChannelRate_t
- struct unpack_wds_SLQSSGetLoopback_t
- struct pack_wds_SLQSSSetLoopback_t
- struct wds_DataULongTlv
- struct wds_DataULongLongTlv
- struct unpack_RMTransferStatistics_ind_t
- struct pack_wds_DHCPv4ClientLeaseChange_t
- struct wds_DHCPProfileIdTlv
- struct wds_DHCPLeaseStateTlv
- struct wds_IPv4AdTlv
- struct wds_DHCPOpt
- struct wds_DHCPLeaseOptTlv
- struct unpack_wds_DHCPv4ClientLease_ind_t
- struct pack_wds_SetMobileIP_t
- struct pack_wds_SetMobileIPParameters_t
- struct pack_wds_SetAutoconnect_t
- struct unpack_wds_GetAutoconnect_t
- struct wds_TrStatInd
- struct pack_wds_SLQSWdsSetEventReport_t
- struct wds_DHCPv4ProfileId
- struct wds_DHCPv4HWConfig
- struct wds_DHCPv4Option
- struct wds_DHCPv4OptionList
- struct pack_wds_SLQSSSetDHCPv4ClientConfig_t
- struct unpack_wds_GetDataBearerTechnology_t
- struct wds_ConnStatusTlv
- struct wds_LastMdmCallEndRsnTlv
- struct wds_TXBytesOKTlv
- struct wds_RXBytesOKTlv
- struct wds_DormStatTlv
- struct wds_DataBearTechTlv
- struct wds_channelRate
- struct wds_ChannelRateTlv
- struct unpack_wds_SLQSDUNCallInfoCallBack_ind_t

## Macros

- #define IPV6_ADDRESS_ARRAY_SIZE 8
- #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24
- #define PACK_WDS_IPV4 4
- #define PACK_WDS_IPV6 6
- #define BYT_STAT_STAT_MASK 0X000000C0
- #define WDS_DHCP_MAX_NUM_OPTIONS 30
- #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /∗ current max size of raw message in SDK process is 2048 ∗/

## Typedefs

- typedef union unpackWdsProfileParam UnpackQmiProfileInfo

## Enumerations

- enum liteQmiDataBearerMasks {
  QMI_LITE_WDS_CURRENT_CALL_DB_MASK = 0x01,
  QMI_LITE_WDS_LAST_CALL_DB_MASK = 0x02 }

## Functions

- int pack_wds_SLQSStartDataSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSStartDataSession_t *reqArg)
- int unpack_wds_SLQSStartDataSession (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSStartDataSession_t *pOutput)
- int unpack_wds_SLQSSetPacketSrvStatusCallback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetPacketSrvStatusCallback_t *pOutput)
- int pack_wds_SLQSStopDataSession (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSStopDataSession_t *reqArg)
- int unpack_wds_SLQSStopDataSession (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_SLQSGetRuntimeSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetRuntimeSettings_t *pOutput)
- int unpack_wds_SLQSSetWdsEventCallback_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetWdsEventCallback_ind_t *pOutput)
- int unpack_wds_SLQSSetWdsEventCallback (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSSetWdsEventCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetWdsEventCallback_t *reqArg)
- int pack_wds_SLQSGetRuntimeSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetRuntimeSettings_t *reqArg)
- int pack_wds_SLQSCreateProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSCreateProfile_t *reqArg)
- int unpack_wds_SLQSCreateProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSCreateProfile_t *pOutput)
- int pack_wds_SLQSModifyProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSModifyProfile_t *reqArg)
- int unpack_wds_SLQSModifyProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSModifyProfile_t *pOutput)
- int pack_wds_SLQSGetProfileSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetProfileSettings_t *reqArg)
- int unpack_wds_SLQSGetProfileSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetProfileSettings_t *pOutput)
- int pack_wds_GetSessionState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetSessionState (uint8_t *pResp, uint16_t respLen, unpack_wds_GetSessionState_t *pOutput)
- int pack_wds_GetDefaultProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfile_t *reqParam)
- int unpack_wds_GetDefaultProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfile_t *pOutput)
- int pack_wds_GetConnectionRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetConnectionRate (uint8_t *pResp, uint16_t respLen, unpack_wds_GetConnectionRate_t *pOutput)
- int pack_wds_GetPacketStatus (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatus_t *reqParam)

- int unpack_wds_GetPacketStatus (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatus_t *pOutput)
- int pack_wds_GetSessionDuration (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetSessionDuration_t *reqParam)
- int unpack_wds_GetSessionDuration (uint8_t *pResp, uint16_t respLen, unpack_wds_GetSessionDuration_t *pOutput)
- int pack_wds_GetDormancyState (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDormancyState_t *reqParam)
- int unpack_wds_GetDormancyState (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDormancyState_t *pOutput)
- int pack_wds_SLQSDeleteProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSDeleteProfile_t *reqParam)
- int unpack_wds_SLQSDeleteProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSDeleteProfile_t *pOutput)
- int pack_wds_SetDefaultProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfile_t *reqParam)
- int unpack_wds_SetDefaultProfile (uint8_t *pResp, uint16_t respLen)
- int pack_wds_SLQSGet3GPPConfigItem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSGet3GPPConfigItem (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGet3GPPConfigItem_t *pOutput)
- int pack_wds_SLQSSet3GPPConfigItem (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSet3GPPConfigItem_t *reqParam)
- int unpack_wds_SLQSSet3GPPConfigItem (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetMobileIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIP_t *pReqParam)
- int unpack_wds_GetMobileIP (uint8_t *pResp, uint16_t respLen, unpack_wds_GetMobileIP_t *pOutput)
- int pack_wds_GetMobileIPProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetMobileIPProfile_t *reqParam)
- int unpack_wds_GetMobileIPProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_GetMobileIPProfile_t *pOutput)
- int pack_wds_SLQSGetCurrDataSystemStat (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetCurrDataSystemStat_t *pReqParam)
- int unpack_wds_SLQSGetCurrDataSystemStat (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrDataSystemStat_t *pOutput)
- int pack_wds_GetLastMobileIPError (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetLastMobileIPError_t *pReqParam)
- int unpack_wds_GetLastMobileIPError (uint8_t *pResp, uint16_t respLen, unpack_wds_GetLastMobileIPError_t *pOutput)
- int pack_wds_RMSetTransferStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_RMSetTransferStatistics_t *reqParam)
- int unpack_wds_RMSetTransferStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_RMSetTransferStatistics_t *pOutput)
- int pack_wds_SetMobileIPProfile (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPProfile_t *reqParam)
- int unpack_wds_SetMobileIPProfile (uint8_t *pResp, uint16_t respLen, unpack_wds_SetMobileIPProfile_t *pOutput)
- int pack_wds_SLQSWdsSwiPDPRuntimeSettings (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t *reqParam)
- int unpack_wds_SLQSWdsSwiPDPRuntimeSettings (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t *pOutput)
- int pack_wds_SLQSGetDUNCallInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDUNCallInfo_t *reqParam)
- int unpack_wds_SLQSGetDUNCallInfo (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDUNCallInfo_t *pOutput)
- int pack_wds_SLQSGetDataBearerTechnology (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetDataBearerTechnology_t *pReqParam)

- int unpack_wds_SLQSGetDataBearerTechnology (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetDataBearerTechnology_t *pOutput)
- int pack_wds_SLQSSetIPFamilyPreference (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetIPFamilyPreference_t *pReqParam)
- int unpack_wds_SLQSSetIPFamilyPreference (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetIPFamilyPreference_t *pOutput)
- int pack_wds_SetDefaultProfileNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetDefaultProfileNum_t *pReqParam)
- int unpack_wds_SetDefaultProfileNum (uint8_t *pResp, uint16_t respLen)
- int pack_wds_GetDefaultProfileNum (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfileNum_t *pReqParam)
- int unpack_wds_GetDefaultProfileNum (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfileNum_t *pOutput)
- int pack_wds_SLQSSGetDHCPv4ClientConfig (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSGetDHCPv4ClientConfig_t *pReq)
- int unpack_wds_SLQSSGetDHCPv4ClientConfig (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSGetDHCPv4ClientConfig_t *pOutput)
- int pack_wds_GetPacketStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatistics_t *pReq)
- int unpack_wds_GetPacketStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatistics_t *pOutput)
- int pack_wds_GetByteTotals (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_GetByteTotals (uint8_t *pResp, uint16_t respLen, unpack_wds_GetByteTotals_t *pOutput)
- int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSGetCurrentChannelRate (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrentChannelRate_t *pOutput)
- int pack_wds_SLQSSGetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSSGetLoopback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSGetLoopback_t *pOutput)
- int pack_wds_SLQSSSetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSSetLoopback_t *reqArg)
- int unpack_wds_SLQSSSetLoopback (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_RMTransferStatistics_ind (uint8_t *pResp, uint16_t respLen, unpack_RMTransferStatistics_ind_t *pOutput)
- int pack_wds_DHCPv4ClientLeaseChange (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_DHCPv4ClientLeaseChange_t *reqArg)
- int unpack_wds_DHCPv4ClientLeaseChange (uint8_t *pResp, uint16_t respLen)
- 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 *pOutput)
- int pack_wds_SLQSWdsSetEventReport (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSWdsSetEventReport_t *reqArg)
- int unpack_wds_SLQSWdsSetEventReport (uint8_t *pResp, uint16_t respLen)
- 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_SLQSResetPacketStatics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)
- int unpack_wds_SLQSResetPacketStatics (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_GetDataBearerTechnology_t *pOutput)
- int pack_wds_SetMuxID (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, uint8_t *pMuxID)
- int unpack_wds_SetMuxID (uint8_t *pResp, uint16_t respLen)
- int unpack_wds_SLQSDUNCallInfoCallBack_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSDUNCallInfoCallBack_ind_t *pOutput)

### 9.45.1 Detailed Description

### 9.45.2 SO Mask

- SO mask to indicate the service option or type of application.
  - An SO mask value of zero indicates that this field is ignored.
  - Values:
    - 0x00 - DONT_CARE
  - CDMA 1X SO mask:
    - 0x01 - CDMA_1X_IS95
    - 0x02 - CDMA_1X_IS2000
    - 0x04 - CDMA_1X_IS2000_REL_A
  - CDMA EV-DO Rev 0 SO mask:
    - 0x01 - DPA
  - CDMA EV-DO Rev A SO mask:
    - 0x01 - DPA
    - 0x02 - MFPA
    - 0x04 - EMPA
    - 0x08 - EMPA_EHRPD
  - CDMA EV-DO Rev B SO mask:
    - 0x01 - DPA
    - 0x02 - MFPA
    - 0x04 - EMPA
    - 0x08 - EMPA_EHRPD
    - 0x10 - MMPA
    - 0x20 - MMPA_EHRPD

### 9.45.3 RAT Mask

- RAT mask to indicate the type of technology. A RAT mask value of zero indicates that this field is ignored.

  Values:
  - 0x00 - DONT_CARE
  - 0x8000 - NULL_BEARER

- CDMA RAT mask:
  - 0x01 - CDMA_1X
  - 0x02 - EVDO_REV0
  - 0x04 - EVDO_REVA
  - 0x08 - EVDO_REVB
  - 0x10 - EHRPD
  - 0x20 - FMC
- UMTS RAT mask:
  - 0x01 - WCDMA
  - 0x02 - GPRS
  - 0x04 - HSDPA
  - 0x08 - HSUPA
  - 0x10 - EDGE
  - 0x20 - LTE
  - 0x40 - HSDPA+
  - 0x80 - DC_HSDPA+
  - 0x100 - 64_QAM
  - 0x200 - TD-SCDMA

## 9.45.4 Macro Definition Documentation

### 9.45.4.1 #define BYT_STAT_STAT_MASK 0X000000C0

### 9.45.4.2 #define IPV6_ADDRESS_ARRAY_SIZE 8

### 9.45.4.3 #define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24

### 9.45.4.4 #define PACK_WDS_IPV4 4

### 9.45.4.5 #define PACK_WDS_IPV6 6

### 9.45.4.6 #define WDS_DHCP_MAX_NUM_OPTIONS 30

### 9.45.4.7 #define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /∗ current max size of raw message in SDK process is 2048 ∗/

## 9.45.5 Typedef Documentation

### 9.45.5.1 typedef union unpackWdsProfileParam UnpackQmiProfileInfo

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

**Parameters**

| | |
|---|---|
| *SlqsProfile3GPP* | 3GPP profile<br>• See LibpackProfile3GPP |
| *SlqsProfile3GPP2* | 3GPP2 profile<br>• See LibpackProfile3GPP2 |

### 9.45.6 Enumeration Type Documentation

#### 9.45.6.1 enum liteQmiDataBearerMasks

Bit mask values to indicate the presence of data bearer information for the current and last data calls

**Enumerator**

***QMI_LITE_WDS_CURRENT_CALL_DB_MASK***

***QMI_LITE_WDS_LAST_CALL_DB_MASK***

### 9.45.7 Function Documentation

#### 9.45.7.1 int pack_wds_DHCPv4ClientLeaseChange ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_DHCPv4ClientLeaseChange_t * *reqArg* )

DHCPv4 lease state changes pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.2 int pack_wds_GetAutoconnect ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets auto connect data session setting pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.3 int pack_wds_GetByteTotals ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get Rx/Tx byte counts since the start of the last packet data session pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.4 int pack_wds_GetConnectionRate ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get connection rate pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: No

#### 9.45.7.5 int pack_wds_GetDataBearerTechnology ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get current data bearer technology pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.6 int pack_wds_GetDefaultProfile ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetDefaultProfile_t * *reqParam* )

get default profile pack.

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.7 int pack_wds_GetDefaultProfileNum ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetDefaultProfileNum_t * *pReqParam* )

get default profile number pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.8 int pack_wds_GetDormancyState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_GetDormancyState_t * *reqParam* )

get dormancy state pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: No

#### 9.45.7.9 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: No

#### 9.45.7.10 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: No

#### 9.45.7.11 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.12 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.13 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.14 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: No

#### 9.45.7.15 int pack_wds_GetSessionState ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get session state pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.16 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Note**

PDN Specific: No

#### 9.45.7.17 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

### 9.45.7.18 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

### 9.45.7.19 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.20 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.21 int pack_wds_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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.22 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.23 int pack_wds_SetMuxID ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pMuxID )

Set MUX ID pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pMuxID* | MUX ID.<br>• 0x80 to 0x88 |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.24 int pack_wds_SLQSCreateProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSCreateProfile_t * reqArg )

Create Profile pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.25 int pack_wds_SLQSDeleteProfile ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSDeleteProfile_t * *reqParam* )

delete stored profile pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.26 int pack_wds_SLQSGet3GPPConfigItem ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

get 3Gpp config items pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.27 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.28 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.29 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.30 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.31 int pack_wds_SLQSGetProfileSettings ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetProfileSettings_t * *reqArg* )

Get Profile Settings pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.32 int pack_wds_SLQSGetRuntimeSettings ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSGetRuntimeSettings_t * *reqArg* )

get runtime settings pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.33 int pack_wds_SLQSModifyProfile ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen*, pack_wds_SLQSModifyProfile_t * *reqArg* )

Modify Profile pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.34 int pack_wds_SLQSResetPacketStatics ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Reset packet data transfer statistics pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.35 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.36 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.37 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: No

#### 9.45.7.38 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *pReq* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.39 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.40 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

see qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.41 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.42 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.43 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.44 int pack_wds_SLQSWdsGoActive ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets the device into Active state pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.45 int pack_wds_SLQSWdsGoDormant ( pack_qmi_t * *pCtx*, uint8_t * *pReqBuf*, uint16_t * *pLen* )

Gets the device into dormant state pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.46 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 | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqArg* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.47 int pack_wds_SLQSWdsSwiPDPRuntimeSettings ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSWdsSwiPDPRuntimeSettings_t * reqParam )

swi pdp runtime settings pack

**Parameters**

| | | |
|---|---|---|
| in,out | *pCtx* | qmi request context |
| out | *pReqBuf* | qmi request buffer |
| out | *pLen* | qmi request length |
| in | *reqParam* | request parameter |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

**Note**

PDN Specific: Yes

#### 9.45.7.48 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 | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.49 int unpack_wds_DHCPv4ClientLeaseChange ( uint8_t * *pResp*, uint16_t *respLen* )

DHCPv4 lease state changes unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.50 int unpack_wds_GetAutoconnect ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetAutoconnect_t * *pOutput* )

Gets auto connect data session setting unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.51 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 | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.52 int unpack_wds_GetConnectionRate ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_GetConnectionRate_t** * *pOutput* )

get connection rate unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.53 int unpack_wds_GetDataBearerTechnology ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_GetDataBearerTechnology_t** * *pOutput* )

get current data bearer technology unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.54 int unpack_wds_GetDefaultProfile ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_GetDefaultProfile_t** * *pOutput* )

get default profile unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.55 int unpack_wds_GetDefaultProfileNum ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_GetDefaultProfileNum_t** * *pOutput* )

get default profile number unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.56 int unpack_wds_GetDormancyState ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_GetDormancyState_t** * *pOutput* )

get dormancy state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.57 int unpack_wds_GetLastMobileIPError ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetLastMobileIP-Error_t * *pOutput* )

get current data system unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.58 int unpack_wds_GetMobileIP ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetMobileIP_t * *pOutput* )

get mobile ip mode unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.59 int unpack_wds_GetMobileIPProfile ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetMobileIPProfile_t * *pOutput* )

get mobile ip profile unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.60 int unpack_wds_GetPacketStatistics ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatistics_t * *pOutput* )

gets current packet transfer counter values unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.61 int unpack_wds_GetPacketStatus ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetPacketStatus_t * *pOutput* )

get packet status unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.62 int unpack_wds_GetSessionDuration ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_GetSessionDuration_t * *pOutput* )

get session duration unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.63 int unpack_wds_GetSessionState ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_GetSessionState_t** * *pOutput* )

get session state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.64 int unpack_wds_RMSetTransferStatistics ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_RMSetTransferStatistics_t** * *pOutput* )

RM set transfer statistics unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.65 int unpack_wds_RMTransferStatistics_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_RMTransferStatistics_ind_t** * *pOutput* )

RM transfer statistics indication unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.66 int unpack_wds_SetAutoconnect ( uint8_t * *pResp*, uint16_t *respLen* )

Auto connect data session parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.67 int unpack_wds_SetDefaultProfile ( uint8_t * *pResp*, uint16_t *respLen* )

set default profile unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.68 int unpack_wds_SetDefaultProfileNum ( uint8_t * *pResp*, uint16_t *respLen* )

set default profile number unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.69 int unpack_wds_SetMobileIP ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the current mobile IP setting unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.70 int unpack_wds_SetMobileIPParameters ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the specified mobile IP parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.71 int unpack_wds_SetMobileIPProfile ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_SetMobileIPProfile_t** * *pOutput* )

set mobile ip profile unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.72 int unpack_wds_SetMuxID ( uint8_t * *pResp*, uint16_t *respLen* )

Set MUX ID unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.73 int unpack_wds_SLQSCreateProfile ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSCreateProfile_t * *pOutput* )

Create Profile unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response structure to fill |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.74 int unpack_wds_SLQSDeleteProfile ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSDeleteProfile_t * *pOutput* )

delete stored profile unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.75 int unpack_wds_SLQSDUNCallInfoCallBack_ind ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSDUNCallInfoCallBack_ind_t * *pOutput* )

unpack DUN call info indication

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.76 int unpack_wds_SLQSGet3GPPConfigItem ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGet3GPPConfigItem_t * *pOutput* )

get 3GPP config items unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.77 int unpack_wds_SLQSGetCurrDataSystemStat ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetCurrDataSystemStat_t * *pOutput* )

get current data system unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.78 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 | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.79 int unpack_wds_SLQSGetDataBearerTechnology ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetDataBearerTechnology_t * *pOutput* )

get data bearer technology unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.80 int unpack_wds_SLQSGetDUNCallInfo ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSGetDUNCallInfo_t * *pOutput* )

get dun call info unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.81 int unpack_wds_SLQSGetProfileSettings ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_SLQSGetProfileSettings_t** * *pOutput* )

get session state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response structure to fill |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.82 int unpack_wds_SLQSGetRuntimeSettings ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_SLQSGetRuntimeSettings_t** * *pOutput* )

get runtime settings unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.83 int unpack_wds_SLQSModifyProfile ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_SLQSModifyProfile_t** * *pOutput* )

Modify Profile unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response structure to fill |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.84 int unpack_wds_SLQSResetPacketStatics ( uint8_t * *pResp*, uint16_t *respLen* )

Reset packet data transfer statistics unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.85 int unpack_wds_SLQSSet3GPPConfigItem ( uint8_t * *pResp*, uint16_t *respLen* )

set 3GPP config items unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

#### 9.45.7.86 int unpack_wds_SLQSSetIPFamilyPreference ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSetIPFamilyPreference_t * *pOutput* )

Set IP Family Preference unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.87 int unpack_wds_SLQSSetPacketSrvStatusCallback ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_SLQSSetPacketSrvStatusCallback_t** * *pOutput* )

set packet srv status callback unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response |
| in | *respLen* | length |
| out | *pOutput* | unpacked response |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.88 int unpack_wds_SLQSSetWdsEventCallback ( uint8_t * *pResp*, uint16_t *respLen* )

set event callback unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.89 int unpack_wds_SLQSSetWdsEventCallback_ind ( uint8_t * *pResp*, uint16_t *respLen*, **unpack_wds_SLQSSetWdsEventCallback_ind_t** * *pOutput* )

set event callback unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.90 int unpack_wds_SLQSSGetDHCPv4ClientConfig ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSSGetDHCPv4ClientConfig_t * *pOutput* )

get DHCPv4 Client Config unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.91 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 | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.92 int unpack_wds_SLQSSSetDHCPv4ClientConfig ( uint8_t * *pResp*, uint16_t *respLen* )

Gets the DHCP Client V4 Configuration unpack.

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.93 int unpack_wds_SLQSSSetLoopback ( uint8_t * *pResp*, uint16_t *respLen* )

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.94 int unpack_wds_SLQSStartDataSession ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSStartDataSession_t * *pOutput* )

start data session unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | response unpacked |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.95 int unpack_wds_SLQSStopDataSession ( uint8_t * *pResp*, uint16_t *respLen* )

stop data session unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.96 int unpack_wds_SLQSWdsGoActive ( uint8_t * *pResp*, uint16_t *respLen* )

Gets the device into Active state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.97 int unpack_wds_SLQSWdsGoDormant ( uint8_t * *pResp*, uint16_t *respLen* )

Gets the device into dormant state unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.98 int unpack_wds_SLQSWdsSetEventReport ( uint8_t * *pResp*, uint16_t *respLen* )

Sets the event report parameters unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |

**Returns**

eQCWWAN_ERR_NONE on success, eQCWWAN_xxx error value otherwise

**See Also**

See qmerrno.h for eQCWWAN_xxx error values

#### 9.45.7.99 int unpack_wds_SLQSWdsSwiPDPRuntimeSettings ( uint8_t * *pResp*, uint16_t *respLen*, unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t * *pOutput* )

get current data system unpack

**Parameters**

| | | |
|---|---|---|
| in | *pResp* | qmi response from modem |
| in | *respLen* | qmi response length |
| out | *pOutput* | 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, 897
_litefw_FirmwareInfo_, 33
    szCarrier_str, 33
    szCarrierPriversion_str, 34
    szFwversion_str, 34
    szModelid_str, 34
    szPackageid_str, 34
    szSku_str, 34

AAASPI
    unpack_wds_GetMobileIPProfile_t, 888
AAAState
    unpack_wds_GetMobileIPProfile_t, 888
ACT_CODE_MAX_SIZE
    dms.h, 1023
ALS
    voice_getAllCallInformation, 961
AMSSString
    unpack_dms_GetFirmwareRevision_t, 568
    unpack_dms_GetFirmwareRevisions_t, 569
AMTlv
    unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
AOATlv
    unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
APNName
    unpack_wds_SLQSGetRuntimeSettings_t, 905
AWMTlv
    unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
AWOATlv
    unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
accelTemp
    pack_loc_SLQSLOCInjectSensorData_t, 341
acceleroData
    pack_loc_SLQSLOCInjectSensorData_t, 341
acceleroTimeSrc
    pack_loc_SLQSLOCInjectSensorData_t, 341
AccessMac
    nas_protocolSubtypeElement, 232
accolc
    pack_nas_SetACCOLC_t, 347
accuracy
    pack_pds_SetPDSDefaults_t, 385
ackIndicator
    sMSTransferRouteMTMessageInfo, 515
acqOrdeLen
    nas_acqOrderPref, 141
    nas_AcqOrderPrefTlv, 141
acroamsetting
    pack_wds_SetAutoconnect_t, 463
acsetting
    pack_wds_SetAutoconnect_t, 463
actCode
    pack_dms_ActivateAutomatic_t, 303
ActSetCnt
    nas_NetworkStat1x, 215
ActSetPilotPN
    nas_ActPilotPNElement, 142
ActSetPilotPNStrength
    nas_ActPilotPNElement, 142
action
    pack_nas_InitiateDomainAttach_t, 346
activated_ind
    sms_qaQmi3GPP2BroadcastCfgInfo, 504
    sms_qaQmi3GPPBroadcastCfgInfo, 505
activationState
    pack_dms_SetActivationStatusCallback_t, 304
activationStatus
    dms_ActivationStatusTlv, 52
ActivationStatusTlv
    unpack_dms_SetEventReport_ind_t, 584
activeBand
    nas_RfBandInfoExtFormatElements, 239
activeBandClass
    nas_RFBandInfoElements, 238
    nas_RFInfoTlv, 244
    RFBandInfoElements, 492
activeChannel
    nas_RFBandInfoElements, 238
    nas_RfBandInfoExtFormatElements, 239
    nas_RFInfoTlv, 244
    RFBandInfoElements, 492
activeInd
    sms_messageWaitingInfoContent, 503
ActiveStatus
    voice_CLIPResp, 951
    voice_CLIRResp, 952
    voice_CNAPResp, 953
    voice_COLPResp, 953
    voice_COLRResp, 954
ActiveTechPref
    unpack_nas_GetNetworkPreference_t, 679
ActiveTimer
    unpack_dms_PSMCfgChange_ind_t, 581
activeTimer
    dms_PSMActiveTimerTlv, 54
ActiveTimerInd
    dms_PSMActiveTimerIndTlv, 54
addr

unpack_qos_IPv4Addr_t, 757
unpack_qos_IPv6Addr_t, 758
address
loc_IPv4Info, 130
loc_IPv6Info, 131
loc_urlAddr, 138
unpack_wds_GetMobileIPProfile_t, 888
addressSize
unpack_sms_GetSMSCAddress_t, 782
aid
uim_refreshevent, 539
uim_sessionInformation, 541
uim_UIMSessionInformation, 545
aidLength
appStats, 37
uim_appStatus, 524
uim_refreshevent, 539
uim_sessionInformation, 541
uim_UIMSessionInformation, 545
aidVal
appStats, 37
uim_appStatus, 524
aidingIndicatorMask
loc_sensorDataUsage, 136
airTimerValue
voice_airTimer, 929
alertPitch
voice_signalInfo, 973
alertingPattern
voice_arrAlertingPattern, 932
AlertingType
voice_arrAlertingType, 933
alertmsg
unpack_omaDmConfigTlv_t, 739
alertmsglength
unpack_omaDmConfigTlv_t, 739
allCallsAlphaIDInfoArr
voice_arrAlphaID, 933
AllCallsUUSInfo
voice_arrUUSInfo, 939
alphaDcs
voice_alphaIDInfo, 931
AlphaID
cat_AlPhaIdentifierTlv, 45
AlphaIDInfo
voice_allCallsAlphaIDInfo, 929
alphaIDLen
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 799
AlphaIDLength
cat_AlPhaIdentifierTlv, 45
alphaLen
voice_alphaIDInfo, 931
alphaText
voice_alphaIDInfo, 931
altSrcInfo_t, 34
coverage, 35
linkage, 35
source, 35
Altitude
unpack_pds_SLQSGetGPSStateInfo_t, 755
altitudeAssumed
unpack_loc_GnssSvInfo_Ind_t, 655
altitudeSrcInfo
pack_loc_SLQSLOCInjectPosition_t, 339
altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 339
altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 339
ambr_dl
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
ambr_dl_ext
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
ambr_dl_ext2
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
ambr_ul
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
ambr_ul_ext
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
ambr_ul_ext2
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
amrMode
ims_AMRModeInfo, 70
amrOctAlgn
ims_AMROctAlgnInfo, 70
amrWBEnable
ims_EnabAMRWBInfo, 72
amrWBMode
ims_AMRWBModeInfo, 71
amrWBOctAlgn
ims_AMRWBOctAlgnInfo, 72
amssSize
unpack_dms_GetFirmwareRevision_t, 568
unpack_dms_GetFirmwareRevisions_t, 569
apdoxypages.c, 1001
apnId
pack_qos_SLQSQosSwiReadApnExtraParams_t, 395
pack_qos_SLQSQosSwiReadDataStats_t, 395
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
unpack_qos_SLQSQosSwiReadDataStats_t, 765
apnName
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
apnname
unpack_wds_GetDefaultProfile_t, 884
apnsize
unpack_wds_GetDefaultProfile_t, 884
appNameLength

loc_LocApplicationInfo, 132
appProviderLength
loc_LocApplicationInfo, 132
appState
appStats, 37
uim_appStatus, 524
appStats, 35
aidLength, 37
aidVal, 37
appState, 37
appType, 37
persoFeature, 37
persoRetries, 37
persoState, 37
persoUnblockRetries, 37
pin1Retries, 38
pin1State, 38
pin2Retries, 38
pin2State, 38
puk1Retries, 38
puk2Retries, 38
univPin, 38
AppStatus
slotInf, 496
uim_slotInfo, 544
appType
appStats, 37
uim_appStatus, 524
appVersionLength
loc_LocApplicationInfo, 132
appVersionValid
loc_LocApplicationInfo, 132
Application
unpack_nas_GetCDMANetworkParameters_t, 674
appversion_str
unpack_dms_GetFirmwareInfo_t, 566
arfcn
nas_GERANInfo, 170
nas_gsmCellInfo, 172
arrCallInfomation
unpack_voice_allCallStatusCallback_ind_t, 839
arrfileInfo
uim_refreshevent, 539
uim_registerRefresh, 540
audio.h, 1001
pack_audio_SLQSGetAudioPathConfig, 1002
pack_audio_SLQSGetAudioProfile, 1002
pack_audio_SLQSGetAudioVolTLBConfig, 1003
pack_audio_SLQSSetAudioPathConfig, 1003
pack_audio_SLQSSetAudioProfile, 1004
pack_audio_SLQSSetAudioVolTLBConfig, 1004
unpack_audio_SLQSGetAudioPathConfig, 1005
unpack_audio_SLQSGetAudioProfile, 1005
unpack_audio_SLQSGetAudioVolTLBConfig, 1005
unpack_audio_SLQSSetAudioPathConfig, 1006
unpack_audio_SLQSSetAudioProfile, 1006
unpack_audio_SLQSSetAudioVolTLBConfig, 1006
audio_RXAGCList, 38
pRXAIG, 38
pRXComprSlope, 39
pRXComprThres, 39
pRXExpSlope, 39
pRXExpThres, 39
pRXStaticGain, 39
audio_RXAVCList, 39
pAVRXAVCHeadroom, 39
pAVRXAVCSens, 39
audio_RXPCMIIRFltr, 39
pFlag, 41
pStage0Val, 41
pStage1Val, 41
pStage2Val, 41
pStage3Val, 41
pStage4Val, 41
pStageCnt, 41
audio_TXAGCList, 41
pTXAIG, 41
pTXComprSlope, 42
pTXComprThres, 42
pTXExpSlope, 42
pTXExpThres, 42
pTXStaticGain, 42
audio_TXPCMIIRFltr, 42
pFlag, 43
pStage0Val, 43
pStage1Val, 43
pStage2Val, 43
pStage3Val, 43
pStage4Val, 43
pStageCnt, 43
auth
unpack_wds_GetDefaultProfile_t, 884
authData
pack_uim_SLQSUIMAuthenticate_t, 431
AuthProt
nas_protocolSubtypeElement, 232
Authentication
unpack_wds_SLQSGetRuntimeSettings_t, 905
authentication
pack_wds_SetDefaultProfile_t, 464
AutoConnect
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 416
unpack_swiavms_SLQSAVMSGetSettings_t, 807
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
AutoReboot
pack_swiavms_SLQSAVMSSetSettings_t, 415
unpack_swiavms_SLQSAVMSGetSettings_t, 807
Autosdm
unpack_swioma_SLQSOMADMGetSettings_t, 823
avgPeriod
nas_LTESigRptCfg, 203
nas_LTESigRptConfig, 204
azimuth
loc_satelliteInfo, 135

bAPNLength
PackSwiAvmsSetSettingsAPNInfo, 485
PackSwiAVMSSettingsAPNInfo, 487
bAuto
pack_pds_SetPortAutomaticTracking_t, 386
pack_pds_SetServiceAutomaticTracking_t, 386
bBinaryType
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 920
bEnable
pack_dms_UIMSetPINProtection_t, 318
pack_nas_SLQSSetSignalStrengthsCallback_t, 366
bEnabled
pack_pds_SetXTRAAutomaticDownload_t, 387
bForceDownload
pack_fms_SetImagesPreference_t, 321
bICCID
slot_t, 495
bICCIDLength
slot_t, 495
bLogicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 437
slot_t, 495
bNotification
UnpackSwiAvmsEventReportNotification, 925
bNumberOfPhySlots
unpack_uim_SetUimSlotStatusChangeCallback_-ind_t, 830
BOOL
SwiDataTypes.h, 1311
BPTlv
NASQmiCbkNasSystemSelPrefInd, 290
bPWDLength
PackSwiAvmsSetSettingsAPNInfo, 485
PackSwiAVMSSettingsAPNInfo, 487
bPackageID
UnpackSwiAvmsEventReportPackageID, 926
bRegStatus
UnpackSwiAvmsEventReportRegStatus, 926
bResetStatistics
rmTrasnferStaticsReq, 493
bSerity
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 920
bState
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 920
UnpackSwiAvmsEventReportConfig, 922
bType
UnpackSwiAvmsEventReportDataSessionStatus, 923
UnpackSwiAvmsEventReportSessionType, 927
bUnameLength
PackSwiAvmsSetSettingsAPNInfo, 486
PackSwiAVMSSettingsAPNInfo, 487
bUserInputRequest
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 920
UnpackSwiAvmsEventReportConfig, 922
UnpackSwiAvmsEventReportConnectionRequest, 922
BYT_STAT_STAT_MASK
wds.h, 1393
BYTE
SwiDataTypes.h, 1311
band
nas_LTEInfo, 191
band1900
nas_gsmCellInfo, 172
band_pref
NASBandPreferenceTlv, 280
BandCapability
unpack_dms_GetBandCapability_t, 555
bandCapability
unpack_dms_SLQSGetBandCapability_t, 597
BandPref
nas_BandPrefTlv, 145
bandPref
pack_nas_SLQSSetBandPreference_t, 365
bandwidth
nas_LTEInfo, 191
nas_RfBandwidthInfoElements, 241
baseId
nas_CDMAInfo, 148
nas_CDMASysInfo, 152
baseLat
nas_CDMAInfo, 148
nas_CDMASysInfo, 152
baseLong
nas_CDMAInfo, 148
nas_CDMASysInfo, 152
BasestationID
unpack_nas_SLQSGetServingSystem_t, 701
BasestationLatitude
unpack_nas_SLQSGetServingSystem_t, 701
BasestationLongitude
unpack_nas_SLQSGetServingSystem_t, 701
batchPerSec
loc_accelAcceptReady, 121
loc_accelTempAcceptReady, 122
loc_gyroAcceptReady, 128
loc_gyroTempAcceptReady, 128
BearerID
unpack_qos_QosFlowInfo_t, 760
bearerID
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 909
bearerId
unpack_QosFlowStat_t, 778
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 918
bits_129_192
nas_lteBandPrefExt, 186
nas_nr5gBandPref, 219

nas_NR5gBandPrefTlv, 220
bits_193_256
nas_lteBandPrefExt, 186
nas_nr5gBandPref, 219
nas_NR5gBandPrefTlv, 220
bits_1_64
nas_lteBandPrefExt, 186
nas_nr5gBandPref, 219
nas_NR5gBandPrefTlv, 220
bits_65_128
nas_lteBandPrefExt, 186
nas_nr5gBandPref, 219
nas_NR5gBandPrefTlv, 220
bootSize
unpack_dms_GetFirmwareRevisions_t, 569
BootString
unpack_dms_GetFirmwareRevisions_t, 569
bootversion_str
unpack_dms_GetFirmwareInfo_t, 566
Broadcast
unpack_nas_GetCDMANetworkParameters_t, 674
broadcastActivate
pack_sms_SLQSSetSmsBroadcastActivation_t, 404
broadcastConfig
sms_qaQmi3GPPBroadcastCfgInfo, 505
bsInfoValid
nas_CDMASysInfo, 152
bsPRev
nas_CDMASysInfo, 152
bsPRevValid
nas_CDMASysInfo, 152
bsic
nas_GERANInfo, 170
bsicId
nas_gsmCellInfo, 172
bucketSz
unpack_qos_tokenBucket_t, 777
buildID
FMSImageIdElement, 66
image_info_t, 69
buildIDLen
image_info_t, 69
buildIDLength
FMSImageIdElement, 66
buildId
FMSImageElement, 65
buildIdLength
FMSImageElement, 65
BurstDTMFInfo
pack_voice_SLQSVoiceBurstDTMF_t, 443
ByteLoopbackMode
unpack_wds_SLQSSGetLoopback_t, 915
ByteLoopbackMultiplier
unpack_wds_SLQSSGetLoopback_t, 915

CCETlv
unpack_cat_SetCatEventCallback_ind_t, 551
CDMA_P_Rev
unpack_nas_SLQSGetServingSystem_t, 701
CDMABroadcastConfig
sms_qaQmi3GPP2BroadcastCfgInfo, 504
CDMAECIOThreshListLen
nas_CDMAECIOThresh, 147
CDMARSSIThreshListLen
nas_CDMARSSIThresh, 149
CDMASSInfo
unpack_nas_SLQSNasGetSigInfo_t, 722
CDMASystemInfoExt
unpack_nas_SLQSGetServingSystem_t, 701
CHAR
SwiDataTypes.h, 1311
CK_MAX_SIZE
dms.h, 1023
CQIValueCW0
unpack_nas_SLQSSwiGetLteCQI_t, 732
CQIValueCW1
unpack_nas_SLQSSwiGetLteCQI_t, 732
CRTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
CSDomain
unpack_nas_GetServingNetwork_t, 682
CSTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
CUGIndex
voice_CUGInfo, 956
CallBarStatus
unpack_nas_SLQSGetServingSystem_t, 701
callDuration
unpack_wds_GetSessionDuration_t, 893
CallEndReason
wds_LastMdmCallEndRsnTlv, 989
callEndReason
unpack_wds_SLQSGetDUNCallInfo_t, 902
voice_arrCallEndReason, 934
CallFWExtInfo
voice_getCallFWExtInfo, 963
CallFWInfo
voice_getCallFWInfo, 963
callID
pack_voice_SLQSVoiceGetCallInfo_t, 447
pack_voice_SLQSVoiceStopContDTMF_t, 458
unpack_voice_OTASPStatusCallback_ind_t, 841
unpack_voice_SLQSVoiceStopContDTMF_t, 872
unpack_voice_SUPSNotificationCallback_ind_t, 876
unpack_voice_VoiceInfoRecCallback_ind_t, 878
unpack_voice_voicePrivacyChangeCallback_ind_t, 879
voice_allCallsAlphaIDInfo, 929
voice_allCallsDiagInfo, 930
voice_allCallsUUSInfo, 930
voice_arrAlertingPattern, 932
voice_arrAlertingType, 933
voice_arrCallEndReason, 934

voice_arrSvcOption, 938
voice_callInfo, 948
voice_DTMFInfo, 958
voice_getAllCallRmtPtyName, 962
voice_getAllCallRmtPtyNum, 962
voice_peerNumberInfo, 966
callNumber
pack_voice_SLQSVoiceDialCall_t, 445
callState
voice_callInfo, 948
callType
voice_callInfo, 948
CalledPartyNum
voice_arrCalledPartyNum, 934
callerID
voice_callerIDInfo, 943
voice_connectNumInfo, 955
callerIDLen
voice_callerIDInfo, 943
voice_connectNumInfo, 955
callerName
voice_remotePartyName, 971
CallInfo
voice_getAllCallInformation, 961
cardState
slotInf, 496
uim_slotInfo, 544
carrier
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603
carrier_str
unpack_dms_GetFirmwareInfo_t, 566
CarrierImage_t, 43
m_FwBuildId, 44
m_FwImageId, 44
m_PriBuildId, 44
m_PriImageId, 45
m_nCarrierId, 44
m_nFolderId, 44
m_nStorage, 44
cat.h, 1007
pack_cat_CATSendEnvelopeCommand, 1008
pack_cat_CATSendTerminalResponse, 1008
pack_cat_SetCATEventCallback, 1008
unpack_cat_CATSendEnvelopeCommand, 1009
unpack_cat_CATSendEnvelopeCommand_t, 1008
unpack_cat_CATSendTerminalResponse, 1009
unpack_cat_CATSendTerminalResponse_t, 1008
unpack_cat_SetCATEventCallback, 1010
unpack_cat_SetCatEventCallback_ind, 1010
cat_AlPhaIdentifierTlv, 45
AlphaID, 45
AlphaIDLength, 45
ReferenceID, 45
cat_EndProactiveSessionTlv, 47
EndProactiveSession, 47
cat_EventIDDataTlv, 47
Data, 47
DataLength, 47
ReferenceID, 47
cat_EventListTlv, 47
SetupEventList, 48
cat_RefreshTlv, 48
RefreshMode, 48
RefreshStage, 48
cat_commonEventTlv, 45
CatEvent, 46
EventID, 46
EventLength, 46
TlvPresent, 46
cat_currentCatEvent, 46
CatAlphaIdtfr, 46
CatEndPS, 46
CatEvIDData, 46
CatEventLst, 46
CatRefresh, 46
CatAlphaIdtfr
cat_currentCatEvent, 46
CatEndPS
cat_currentCatEvent, 46
CatEvIDData
cat_currentCatEvent, 46
CatEvent
cat_commonEventTlv, 46
CatEventLst
cat_currentCatEvent, 46
CatRefresh
cat_currentCatEvent, 46
causeCode
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 799
ccsSupported
nas_CDMASysInfo, 152
ccsSupportedValid
nas_CDMASysInfo, 152
cdmaSSInfo, 48
ecio, 49
rssi, 49
cdmaSysIdValid
nas_CDMASysInfo, 152
cell_resel_priority
nas_infoInterFreq, 184
cellBroadcastCap
nas_AddSysInfo, 143
CellID
unpack_nas_SLQSGetServingSystem_t, 701
cellID
nas_GERANInfo, 170
nas_UMTSInfo, 270
cellId
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_WCDMASysInfo, 278
cellIdValid
nas_gsmCellInfo, 172
nas_GSMSysInfo, 176
nas_LTESysInfo, 208

nas_WCDMASysInfo, 279
cellInterFreqParams
nas_infoInterFreq, 184
cellIsTDD
nas_umtsLTENbrCell, 272
CellParams
nas_LTEInfoIntrafreq, 194
cellReselPriority
nas_lteGsmCellInfo, 189
nas_LTEInfoIntrafreq, 194
nas_lteWcdmaCellInfo, 209
cells_len
nas_infoInterFreq, 184
nas_lteGsmCellInfo, 189
cellsLen
nas_LTEInfoIntrafreq, 194
nas_lteWcdmaCellInfo, 209
chaddr
wds_DHCPv4HWConfig, 983
wdsDhcpv4HwConfig, 997
chaddrLen
wds_DHCPv4HWConfig, 983
wdsDhcpv4HwConfig, 997
changePIN
pack_uim_ChangePin_t, 428
ChannelRate
wds_ChannelRateTlv, 977
channelRate
unpack_wds_SLQSGetDUNCallInfo_t, 902
Chipset
nas_DeviceConfigDetail, 165
ciotAcqOrderLen
nas_ciotAcqOrderPref, 155
nas_CiotAcqOrderPrefTlv, 156
ciotLteOpModePref
nas_CiotLteOpModePrefTlv, 156
ckLen
uim_depersonalizationInformation, 530
ckVal
uim_depersonalizationInformation, 530
clear
pack_dms_SLQSSwiGetCrashInfo_t, 312
cmdID
pack_cat_CATSendEnvelopeCommand_t, 301
codingScheme
nas_PLMNNetworkNameData, 230
voice_remotePartyName, 971
common.h
eAUDIO, 1014
eCAT, 1014
eCTL, 1013
eDMS, 1013
eIMS, 1014
eIMSA, 1014
eIND, 1014
eLOC, 1014
eLOG_DEBUG, 1013
eLOG_FATAL, 1013
eLOG_INFO, 1013
eLOG_WARN, 1013
eNAS, 1013
ePDS, 1013
eQOS, 1013
eREQ, 1014
eRMS, 1014
eRSP, 1014
eSAR, 1014
eSMS, 1013
eSWIAUDIO, 1014
eSWIDMS, 1014
eSWILOC, 1014
eSWIM2MCMD, 1014
eSWIM2MCMD_AVC2, 1014
eSWIOMA, 1014
eSWIOMAEXT, 1014
eTIMEOUT_10_S, 1014
eTIMEOUT_20_S, 1014
eTIMEOUT_2_S, 1014
eTIMEOUT_300_S, 1014
eTIMEOUT_30_S, 1014
eTIMEOUT_5_S, 1014
eTIMEOUT_60_S, 1014
eTIMEOUT_8_S, 1014
eTIMEOUT_DEFAULT, 1014
eTMD, 1014
eUIM, 1014
eVOICE, 1013
eWDS, 1013
common.h, 1010
eLOG_LEVEL, 1013
eQMI_SVC, 1013
eTimeout, 1014
fill_pack_ctx, 1015
fill_sdu_hdr, 1015
get_version, 1015
glog, 1016
gloglvl, 1016
helper_get_error_code, 1015
helper_get_error_reason, 1015
helper_get_req_str, 1015
helper_get_resp_ctx, 1015
helper_get_xid, 1015
helper_set_log_func, 1015
helper_set_log_lvl, 1015
liteqmi_GetVersion, 1015
liteqmi_helper_decode7bitAsciiEncString, 1016
liteqmi_log, 1016
logger, 1013
MINREQBKLEN, 1012
MSGID_AND_LEN, 1012
MSGID_DONT_CARE, 1012
msgtype, 1014
SDU_HDR_LEN, 1013
UNUSEDPARAM, 1013
unpack_result_code_only, 1016
commonInfo

unpack_nas_SLQSNasSwiModemStatus_t, 727
ConcSvcInfo
unpack_nas_SLQSGetServingSystem_t, 702
confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 343
conn_status
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 909
ConnectedPartyNum
voice_arrConnectPartyNum, 936
connectionStatus, 49
MDMCallDuration, 49
MDMConnStatus, 49
unpack_wds_GetSessionState_t, 893
unpack_wds_SLQSGetDUNCallInfo_t, 902
content
uim_authenticateResult, 525
uim_readResult, 537
contentLen
uim_authenticateResult, 525
uim_readResult, 537
context
uim_authenticationData, 526
contextId
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 483
unpack_wds_SLQSWdsSwiPDPRuntimeSettings-_t, 918
contextType
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 483
ControlMac
nas_protocolSubtypeElement, 232
count0
nas_RankIndicatorTlv, 236
count1
nas_RankIndicatorTlv, 236
countryInitials
nas_PLMNNetworkNameData, 230
coverage
altSrcInfo_t, 35
cpich_ecno
nas_wcdmaCellInfo, 274
cpich_rscp
nas_wcdmaCellInfo, 274
cradleMountConfigStatus
unpack_loc_CradleMountCallback_Ind_t, 648
crashAction
pack_dms_SetCrashAction_t, 305
crashData
crashInformation, 51
crashId
crashInformation, 51
crashInfo
crashInfoParams, 50
crashInfoParam
unpack_dms_SLQSSwiGetCrashInfo_t, 601
crashInfoParams, 50
crashInfo, 50
crashStatus, 50
crashInformation, 50
crashData, 51
crashId, 51
crashString, 51
crashStrlen, 51
gcdumpString, 51
gcdumpStrlen, 51
numCrashes, 51
crashStatus
crashInfoParams, 50
crashString
crashInformation, 51
crashStrlen
crashInformation, 51
csAttachState
nas_servSystem, 254
NASServingSystemInfo, 293
csBarStatus
nas_CallBarringSysInfo, 145
nas_callBarStatus, 146
cscfPortName
ims_CSCFPortNameInfo, 72
cur_carr_name
unpack_dms_GetFirmwareInfo_t, 567
cur_carr_rev
unpack_dms_GetFirmwareInfo_t, 567
curDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 900
curProfile
pack_wds_SLQSModifyProfile_t, 475
UnPackGetProfileSettingOut, 919
CurrChanRxRate
dunchannelRate, 63
wds_channelRate, 977
CurrChanTxRate
dunchannelRate, 63
wds_channelRate, 977
currDBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
currNWInfo
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
currNetworkInfo, 51
NetworkType, 52
RATMask, 52
SOMask, 52
unpack_wds_SLQSGetCurrDataSystemStat_t, 898
current_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 899
current_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 899

currentChannelRXRate
unpack_wds_GetConnectionRate_t, 881
currentChannelTXRate
unpack_wds_GetConnectionRate_t, 881
currentDataBearer
pack_wds_SLQSSetWdsEventCallback_t, 478
currentNetwork
qmiWDSDataBearerTechnology, 490
CurrentPLMN
unpack_nas_SLQSGetServingSystem_t, 702
CurrentmitigationLvl
unpack_tmd_SLQSTmdGetMitigationLvl_t, 825
cust_attr
DMScustSettingInfo, 60
cust_id
DMScustSettingInfo, 60
DMSgetCustomInput, 62
pack_dms_GetCustFeaturesV2_t, 303
pack_dms_SetCustFeaturesV2_t, 308
cust_value
DMScustSettingInfo, 60
pack_dms_SetCustFeaturesV2_t, 308
custSetting
DMScustSettingList, 61
CustomSCP
unpack_nas_GetCDMANetworkParameters_t, 674
CwtMute
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 801
unpack_swiaudio_SLQSGetM2MAVMute_t, 803

DBTTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
dBTechAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
dBTechExtRatValue
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
dBTechExtSoMask
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
dBTechnology
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
dBtechExtAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
dBtechnologyExt
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
DHCPRelayEnabled
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
DHCPv4LeaseOptTlv
unpack_wds_DHCPv4ClientLease_ind_t, 880
DHCPv4LeaseStateTlv
unpack_wds_DHCPv4ClientLease_ind_t, 880
dLatitude
unpack_pds_SetEventReport_Ind_t, 749
dLongitude
unpack_pds_SetEventReport_Ind_t, 749
DMS_IMGDETAILS_LEN
dms.h, 1023
DMS_PM_FACTORY
dms.h, 1023
DMS_PM_LOW
dms.h, 1024
DMS_PM_OFFLINE
dms.h, 1024
DMS_PM_ONLINE
dms.h, 1024
DMS_PM_RESET
dms.h, 1024
DMS_PM_SHUT_DOWN
dms.h, 1024
DMS_VALID_FSN_LEN
dms.h, 1024
DMScustSettingInfo, 60
cust_attr, 60
cust_id, 60
cust_value, 60
id_length, 60
value_length, 61
DMScustSettingList, 61
custSetting, 61
list_type, 61
num_instances, 61
DMSgetCustomFeatureV2, 61
pCustSettingInfo, 62
pCustSettingList, 62
pGetCustomInput, 62
DMSgetCustomInput, 62
cust_id, 62
list_type, 62
DRCCover
nas_DRCParams, 166
DRCValue
nas_DRCParams, 166
DSTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
DTMFEvent
voice_DTMFInfo, 958
DTMFInformation
unpack_voice_DTMFEventCallback_ind_t, 840
DTMFInterdigitInterval
voice_DTMFLengths, 959
DTMFPulseWidth
voice_DTMFLengths, 959
DTMFdigit
pack_voice_SLQSVoiceStartContDTMF_t, 458
DTMInd
unpack_nas_SLQSGetServingSystem_t, 702
Data
cat_EventIDDataTlv, 47

data
sMSCAddressInfo, 511
sMSEtwsMessageInfo, 512
sMSTransferRouteMTMessageInfo, 515
uim_authenticationData, 526
data_buf
NASOTAMessageTlv, 284
data_len
NASOTAMessageTlv, 284
dataBearer
pack_wds_SLQSSetWdsEventCallback_t, 478
dataBearerMask
unpack_wds_SLQSGetDataBearerTechnology_t, 900
DataBearerTech
wds_DataBearTechTlv, 979
dataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 902
dataBearerTechExt
pack_wds_SLQSSetWdsEventCallback_t, 478
dataCapabilities
nas_dataSrvCapabilities, 161
dataCapabilitiesLen
nas_dataSrvCapabilities, 161
DataCaps
unpack_nas_GetServingNetwork_t, 682
unpack_nas_GetServingNetworkCapabilities_t, 684
dataCaps
unpack_nas_SetDataCapabilitiesCallback_ind_t, 687
DataCapsLen
unpack_nas_GetServingNetwork_t, 682
unpack_nas_GetServingNetworkCapabilities_t, 684
dataCapsSize
unpack_nas_SetDataCapabilitiesCallback_ind_t, 687
dataLen
pack_cat_CATSendEnvelopeCommand_t, 301
pack_cat_CATSendTerminalResponse_t, 302
uim_authenticationData, 526
DataLength
cat_EventIDDataTlv, 47
DataRate
unpack_qos_swiQosFlow_t, 775
dataRateMax
unpack_qos_dataRate_t, 757
dataServiceCaCapability
unpack_dms_GetDeviceCapabilities_t, 561
DataServiceCapability
unpack_dms_GetDeviceCap_t, 560
DataSrvCapabilities
unpack_nas_SLQSGetServingSystem_t, 702
dataSysStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 913
dataSystemStatus
pack_wds_SLQSSetWdsEventCallback_t, 478
Date
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
DateLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
day
nas_timeInfo, 267
nas_UniversalTime, 273
dayLtSavingAdj
nas_timeInfo, 267
dayOfWeek
nas_timeInfo, 267
nas_UniversalTime, 273
daylightSavings
nas_qaQmi3Gpp2TimeZone, 233
dedicatedBand
nas_RfDedicatedBandInfoElements, 243
defaultPDNEnabled
unpack_wds_SLQSGet3GPPConfigItem_t, 897
DefaultRoamInd
unpack_nas_SLQSGetServingSystem_t, 702
delayClass
LibPackGPRSRequestedQoS, 91
wds_GPRSQoS, 987
deliveryErrSDU
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 996
depersonilisationInfo
pack_uim_SLQSUIMDepersonalization_t, 432
description
unpack_omaDmFotaTlv_t, 741
descriptionlength
unpack_omaDmFotaTlv_t, 741
Desription
nas_QmiNas3GppNetworkInfo, 234
destPortRangeEnd
LibPackTFTIDParams, 116
destPortRangeStart
LibPackTFTIDParams, 116
DetailedSvcInfo
unpack_nas_SLQSGetServingSystem_t, 702
DevCrashState
unpack_dms_GetCrashAction_t, 555
Device
pack_swiaudio_SLQSSetM2MAudioAVCFG_t, 409
deviceID
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, 826
deviceIdLen
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, 826
DiagInfo
voice_allCallsDiagInfo, 930
voice_arrDiagInfo, 936
diagInfoLen
voice_diagInfo, 957

diagnosticInfo
voice_diagInfo, 957
digitBuff
voice_DTMFInfo, 958
digitCnt
voice_burstDTMFInfo, 940
voice_DTMFInfo, 958
dirNum
nas_dirNum, 165
dirNumLen
nas_dirNum, 165
direction
voice_callInfo, 948
DisableIMSI
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
dispType
voice_extDispRecInfo, 960
displayCondition
nas_serviceProviderName, 252
dl_bw_value
nas_PhyCaAggPcellInfo, 223
nas_PhyCaAggScellDlBw, 223
nas_PhyCaAggScellInfo, 227
NASPhyCaAggPcellInfo, 285
NASPhyCaAggScellDlBw, 286
NASPhyCaAggScellInfo, 288
dms.h, 1016
ACT_CODE_MAX_SIZE, 1023
CK_MAX_SIZE, 1023
DMS_IMGDETAILS_LEN, 1023
DMS_PM_FACTORY, 1023
DMS_PM_LOW, 1024
DMS_PM_OFFLINE, 1024
DMS_PM_ONLINE, 1024
DMS_PM_RESET, 1024
DMS_PM_SHUT_DOWN, 1024
DMS_VALID_FSN_LEN, 1024
ERI_DATA_MAX_SIZE, 1024
MAX_BUILD_ID_LEN, 1024
MEID_MAX_SIZE, 1024
pack_dms_ActivateAutomatic, 1025
pack_dms_GetActivationState, 1025
pack_dms_GetBandCapability, 1025
pack_dms_GetCrashAction, 1026
pack_dms_GetCustFeature, 1026
pack_dms_GetCustFeaturesV2, 1026
pack_dms_GetDeviceCap, 1027
pack_dms_GetDeviceCapabilities, 1027
pack_dms_GetDeviceHardwareRev, 1028
pack_dms_GetDeviceMfr, 1028
pack_dms_GetDeviceSerialNumbers, 1028
pack_dms_GetFSN, 1030
pack_dms_GetFirmwareInfo, 1029
pack_dms_GetFirmwareRevision, 1029
pack_dms_GetFirmwareRevisions, 1029
pack_dms_GetHardwareRevision, 1030
pack_dms_GetIMSI, 1030
pack_dms_GetManufacturer, 1031
pack_dms_GetModelID, 1031
pack_dms_GetNetworkTime, 1031
pack_dms_GetOfflineReason, 1032
pack_dms_GetPRLVersion, 1032
pack_dms_GetPower, 1032
pack_dms_GetSerialNumbers, 1033
pack_dms_GetUSBComp, 1033
pack_dms_GetVoiceNumber, 1033
pack_dms_ResetToFactoryDefaults, 1034
pack_dms_SLQSDmsSwiGetPCInfo, 1038
pack_dms_SLQSDmsSwiGetResetInfo, 1038
pack_dms_SLQSDmsSwiIndicationRegister, 1038
pack_dms_SLQSGetBandCapability, 1039
pack_dms_SLQSGetERIFile, 1039
pack_dms_SLQSGetPowerSaveModeConfig, 1039
pack_dms_SLQSSetPowerSaveModeConfig, 1040
pack_dms_SLQSSwiClearDyingGaspStatistics, 1040
pack_dms_SLQSSwiGetCrashInfo, 1040
pack_dms_SLQSSwiGetDyingGaspCfg, 1041
pack_dms_SLQSSwiGetDyingGaspStatistics, 1041
pack_dms_SLQSSwiGetFirmwareCurr, 1041
pack_dms_SLQSSwiGetFwUpdateStatus, 1042
pack_dms_SLQSSwiGetHostDevInfo, 1042
pack_dms_SLQSSwiGetOSInfo, 1042
pack_dms_SLQSSwiGetSerialNoExt, 1043
pack_dms_SLQSSwiSetDyingGaspCfg, 1043
pack_dms_SLQSSwiSetHostDevInfo, 1043
pack_dms_SLQSSwiSetOSInfo, 1044
pack_dms_SLQSUIMGetState, 1044
pack_dms_SetActivationStatusCallback, 1034
pack_dms_SetCrashAction, 1035
pack_dms_SetCustFeature, 1035
pack_dms_SetCustFeaturesV2, 1035
pack_dms_SetEventReport, 1036
pack_dms_SetFirmwarePreference, 1036
pack_dms_SetIndicationRegister, 1036
pack_dms_SetPower, 1037
pack_dms_SetUSBComp, 1037
pack_dms_SwiSetEventReport, 1045
pack_dms_UIMChangePIN, 1045
pack_dms_UIMGetControlKeyStatus, 1045
pack_dms_UIMGetICCID, 1046
pack_dms_UIMGetPINStatus, 1046
pack_dms_UIMSetControlKeyProtection, 1046
pack_dms_UIMSetPINProtection, 1047
pack_dms_UIMUnblockControlKey, 1047
pack_dms_UIMUnblockPIN, 1048
pack_dms_UIMVerifyPIN, 1048
pack_dms_ValidateSPC, 1048
SPC_SIZE, 1025
UNIQUE_ID_LEN, 1025
unpack_dms_ActivateAutomatic, 1049
unpack_dms_GetActivationState, 1049
unpack_dms_GetBandCapability, 1050
unpack_dms_GetCrashAction, 1050

unpack_dms_GetCustFeature, 1050
unpack_dms_GetCustFeaturesV2, 1051
unpack_dms_GetDeviceCap, 1051
unpack_dms_GetDeviceCapabilities, 1051
unpack_dms_GetDeviceHardwareRev, 1052
unpack_dms_GetDeviceMfr, 1052
unpack_dms_GetDeviceSerialNumbers, 1052
unpack_dms_GetFSN, 1054
unpack_dms_GetFirmwareInfo, 1053
unpack_dms_GetFirmwareRevision, 1053
unpack_dms_GetFirmwareRevisions, 1053
unpack_dms_GetHardwareRevision, 1054
unpack_dms_GetIMSI, 1054
unpack_dms_GetManufacturer, 1055
unpack_dms_GetModelID, 1055
unpack_dms_GetNetworkTime, 1055
unpack_dms_GetOfflineReason, 1056
unpack_dms_GetPRLVersion, 1056
unpack_dms_GetPower, 1056
unpack_dms_GetSerialNumbers, 1057
unpack_dms_GetUSBComp, 1057
unpack_dms_GetVoiceNumber, 1057
unpack_dms_PSMCfgChange_ind, 1058
unpack_dms_ResetToFactoryDefaults, 1058
unpack_dms_SLQSDmsSwiGetPCInfo, 1062
unpack_dms_SLQSDmsSwiGetResetInfo, 1062
unpack_dms_SLQSDmsSwiGetResetInfo_Ind, 1063
unpack_dms_SLQSDmsSwiIndicationRegister, 1063
unpack_dms_SLQSGetBandCapability, 1063
unpack_dms_SLQSGetERIFile, 1064
unpack_dms_SLQSGetPowerSaveModeConfig, 1064
unpack_dms_SLQSSetPowerSaveModeConfig, 1064
unpack_dms_SLQSSwiClearDyingGaspStatistics, 1065
unpack_dms_SLQSSwiGetCrashInfo, 1065
unpack_dms_SLQSSwiGetDyingGaspCfg, 1066
unpack_dms_SLQSSwiGetDyingGaspStatistics, 1066
unpack_dms_SLQSSwiGetFirmwareCurr, 1066
unpack_dms_SLQSSwiGetFwUpdateStatus, 1067
unpack_dms_SLQSSwiGetHostDevInfo, 1067
unpack_dms_SLQSSwiGetOSInfo, 1067
unpack_dms_SLQSSwiGetSerialNoExt, 1068
unpack_dms_SLQSSwiSetDyingGaspCfg, 1068
unpack_dms_SLQSSwiSetHostDevInfo, 1068
unpack_dms_SLQSSwiSetOSInfo, 1069
unpack_dms_SLQSUIMGetState, 1069
unpack_dms_SetActivationStatusCallback, 1058
unpack_dms_SetCrashAction, 1059
unpack_dms_SetCustFeature, 1059
unpack_dms_SetCustFeaturesV2, 1059
unpack_dms_SetEventReport, 1060
unpack_dms_SetEventReport_ind, 1060
unpack_dms_SetFirmwarePreference, 1060
unpack_dms_SetIndicationRegister, 1061
unpack_dms_SetPower, 1061
unpack_dms_SetUSBComp, 1061
unpack_dms_SwiEventReportCallBack_ind, 1069
unpack_dms_SwiSetEventReport, 1070
unpack_dms_UIMChangePIN, 1070
unpack_dms_UIMGetControlKeyStatus, 1070
unpack_dms_UIMGetICCID, 1071
unpack_dms_UIMGetPINStatus, 1071
unpack_dms_UIMSetControlKeyProtection, 1071
unpack_dms_UIMSetPINProtection, 1072
unpack_dms_UIMUnblockControlKey, 1072
unpack_dms_UIMUnblockPIN, 1073
unpack_dms_UIMVerifyPIN, 1073
unpack_dms_ValidateSPC, 1073
dms_ActivationStatusTlv, 52
activationStatus, 52
TlvPresent, 52
dms_OperatingModeTlv, 52
operatingMode, 53
TlvPresent, 53
dms_PSMActiveTimerIndTlv, 53
ActiveTimerInd, 54
TlvPresent, 54
dms_PSMActiveTimerTlv, 54
activeTimer, 54
TlvPresent, 54
dms_PSMDurationDueToOOSTlv, 54
durationDueToOOS, 55
TlvPresent, 55
dms_PSMDurationThresholdTlv, 55
durationThreshold, 55
TlvPresent, 55
dms_PSMEarlyWakeupTimeTlv, 55
earlyWakeupTime, 56
TlvPresent, 56
dms_PSMEnableStateIndTlv, 56
EnableStateInd, 56
TlvPresent, 56
dms_PSMEnableStateTlv, 56
enableState, 57
TlvPresent, 57
dms_PSMPeriodicUpdateTimerIndTlv, 57
PeriodicUpdateTimerInd, 57
TlvPresent, 57
dms_PSMPeriodicUpdateTimerTlv, 57
periodicUpdateTimer, 58
TlvPresent, 58
dms_PSMRandomizationWindowTlv, 58
randomizationWindow, 58
TlvPresent, 58
dms_TemperatureTlv, 58
TempStat, 59
Temperature, 59
TlvPresent, 59
dms_VoltageTlv, 59
TlvPresent, 60
VoltStat, 60

Voltage, 60
domain
wds_DomainNameList, 986
domainLen
wds_Domain, 985
DomainList
unpack_wds_SLQSGetRuntimeSettings_t, 905
domainName
wds_Domain, 985
DormancyStat
wds_DormStatTlv, 986
dormancyStatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
dormancyState
unpack_wds_GetDormancyState_t, 886
dormancyStatus
pack_wds_SLQSSetWdsEventCallback_t, 478
unpack_wds_SLQSGetDUNCallInfo_t, 903
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
downLink
voice_NSSAudioCtrl, 965
dtmSupp
nas_GSMSysInfo, 176
dtmSuppValid
nas_GSMSysInfo, 176
dunchannelRate, 62
CurrChanRxRate, 63
CurrChanTxRate, 63
MaxChanRxRate, 63
MaxChanTxRate, 63
Duration
pack_nas_SetNetworkPreference_t, 349
unpack_nas_GetNetworkPreference_t, 679
durationDueToOOS
dms_PSMDurationDueToOOSTlv, 55
durationThreshold
dms_PSMDurationThresholdTlv, 55

eAUDIO
common.h, 1014
eCAT
common.h, 1014
eCTL
common.h, 1013
eDMS
common.h, 1013
eFW_TYPE_CWE
lite-fw.h, 1098
eFW_TYPE_CWE_NVU
lite-fw.h, 1098
eFW_TYPE_INVALID
lite-fw.h, 1098
eFW_TYPE_MBN
lite-fw.h, 1098
eFW_TYPE_MBN_GOBI
lite-fw.h, 1098
eFW_TYPE_NVU
lite-fw.h, 1098
eFW_TYPE_SPK
lite-fw.h, 1098
eIMS
common.h, 1014
eIMSA
common.h, 1014
eIND
common.h, 1014
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED
nas.h, 1138
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED
nas.h, 1137
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED
nas.h, 1137
eLOC
common.h, 1014
eLOG_DEBUG
common.h, 1013
eLOG_FATAL
common.h, 1013
eLOG_INFO
common.h, 1013
eLOG_WARN
common.h, 1013
eModel_9X15
lite-fw.h, 1099
eModel_9X30
lite-fw.h, 1099
eModel_9x06
lite-fw.h, 1099
eModel_9x07
lite-fw.h, 1099
eModel_Unknown
lite-fw.h, 1099
eModel_WP9X15
lite-fw.h, 1099
eNAS
common.h, 1013
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100
nas.h, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15
nas.h, 1138

eNAS_LTE_CPHY_CA_BW_NRB_LITE_25
nas.h, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50
nas.h, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6
nas.h, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75
nas.h, 1138
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE
nas.h, 1138
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE
nas.h, 1138
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE
nas.h, 1138
ePDS
common.h, 1013
eQCWWAN_ERR_API_MUTEX_TIMEOUT
qmerrno.h, 1247
eQCWWAN_ERR_BUFFER_SZ
qmerrno.h, 1246
eQCWWAN_ERR_CANCEL_OP
qmerrno.h, 1247
eQCWWAN_ERR_DRIVER
qmerrno.h, 1247
eQCWWAN_ERR_ENUM_BEGIN
qmerrno.h, 1246
eQCWWAN_ERR_ENUM_END
qmerrno.h, 1247
eQCWWAN_ERR_FILE_COPY
qmerrno.h, 1246
eQCWWAN_ERR_FILE_OPEN
qmerrno.h, 1246
eQCWWAN_ERR_GENERAL
qmerrno.h, 1246
eQCWWAN_ERR_INTERNAL
qmerrno.h, 1246
eQCWWAN_ERR_INVALID_ARG
qmerrno.h, 1246
eQCWWAN_ERR_INVALID_DEVID
qmerrno.h, 1246
eQCWWAN_ERR_INVALID_FILE
qmerrno.h, 1246
eQCWWAN_ERR_INVALID_QMI_RSP
qmerrno.h, 1246
eQCWWAN_ERR_INVALID_XID
qmerrno.h, 1247
eQCWWAN_ERR_MALFORMED_QMI_RSP
qmerrno.h, 1246
eQCWWAN_ERR_MEMORY
qmerrno.h, 1246
eQCWWAN_ERR_MULTIPLE_DEVICES
qmerrno.h, 1247
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED
qmerrno.h, 1247
eQCWWAN_ERR_NO_CANCELABLE_OP
qmerrno.h, 1247
eQCWWAN_ERR_NO_CONNECTION
qmerrno.h, 1246
eQCWWAN_ERR_NO_DEVICE
qmerrno.h, 1246
eQCWWAN_ERR_NO_SIGNAL
qmerrno.h, 1247
eQCWWAN_ERR_NONE
qmerrno.h, 1246
eQCWWAN_ERR_NULL_TLV
qmerrno.h, 1250
eQCWWAN_ERR_OFFLINE
qmerrno.h, 1247
eQCWWAN_ERR_PDU_GENERATION
qmerrno.h, 1247
eQCWWAN_ERR_QMI_ABORTED
qmerrno.h, 1247
eQCWWAN_ERR_QMI_ACCESS_DENIED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_ACK_NOT_SENT
qmerrno.h, 1249
eQCWWAN_ERR_QMI_ARG_TOO_LONG
qmerrno.h, 1247
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK
qmerrno.h, 1248
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_CALL_FAILED
qmerrno.h, 1247
eQCWWAN_ERR_QMI_CARD_BUSY_RSP
qmerrno.h, 1250
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_CAT_END
qmerrno.h, 1250
eQCWWAN_ERR_QMI_CAT_START
qmerrno.h, 1250
eQCWWAN_ERR_QMI_CAUSE_CODE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED
qmerrno.h, 1247
eQCWWAN_ERR_QMI_CONNECT
qmerrno.h, 1246
eQCWWAN_ERR_QMI_DEVICE_IN_USE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_DEVICE_NOT_READY
qmerrno.h, 1248
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL
qmerrno.h, 1248
eQCWWAN_ERR_QMI_DISABLED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_ENCODING
qmerrno.h, 1248
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE

qmerrno.h, 1250
eQCWWAN_ERR_QMI_EVENT_REG_FAILED
qmerrno.h, 1250
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL
qmerrno.h, 1249
eQCWWAN_ERR_QMI_FDN_RESTRICT
qmerrno.h, 1249
eQCWWAN_ERR_QMI_FLOW_SUSPENDED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_GENERAL
qmerrno.h, 1248
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_IFACE
qmerrno.h, 1246
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE
qmerrno.h, 1249
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INCORRECT_PIN
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE
qmerrno.h, 1249
eQCWWAN_ERR_QMI_INJECT_TIMEOUT
qmerrno.h, 1249
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INTERNAL
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_ARG
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD
qmerrno.h, 1250
eQCWWAN_ERR_QMI_INVALID_HANDLE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_ID
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_INDEX
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_OPERATION
qmerrno.h, 1249
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_PINID
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_PROFILE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_QMI_CMD
qmerrno.h, 1249
eQCWWAN_ERR_QMI_INVALID_QOS_ID
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_TECH_PREF
qmerrno.h, 1247
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP
qmerrno.h, 1250
eQCWWAN_ERR_QMI_INVALID_TRANSITION
qmerrno.h, 1248
eQCWWAN_ERR_QMI_INVALID_TX_ID
qmerrno.h, 1247
eQCWWAN_ERR_QMI_MALFORMED_MSG
qmerrno.h, 1247
eQCWWAN_ERR_QMI_MAX
qmerrno.h, 1249
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT
qmerrno.h, 1248
eQCWWAN_ERR_QMI_MISSING_ARG
qmerrno.h, 1247
eQCWWAN_ERR_QMI_MSG_BLOCKED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_NETWORK_ABORTED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_NETWORK_NOT_READY
qmerrno.h, 1248
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_NO_EFFECT
qmerrno.h, 1247
eQCWWAN_ERR_QMI_NO_ENTRY
qmerrno.h, 1248
eQCWWAN_ERR_QMI_NO_FREE_PROFILE
qmerrno.h, 1247
eQCWWAN_ERR_QMI_NO_MEMORY
qmerrno.h, 1247
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND

qmerrno.h, 1247
eQCWWAN_ERR_QMI_NO_RADIO
qmerrno.h, 1249
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION
qmerrno.h, 1249
eQCWWAN_ERR_QMI_NO_THRESHOLDS
qmerrno.h, 1247
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_NOT_PROVISIONED
qmerrno.h, 1247
eQCWWAN_ERR_QMI_NOT_SUPPORTED
qmerrno.h, 1249
eQCWWAN_ERR_QMI_OFFSET
qmerrno.h, 1247
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED
qmerrno.h, 1247
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED
qmerrno.h, 1247
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE
qmerrno.h, 1249
eQCWWAN_ERR_QMI_OUT_OF_CALL
qmerrno.h, 1247
eQCWWAN_ERR_QMI_PIN_BLOCKED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_POLICY_MISMATCH
qmerrno.h, 1249
eQCWWAN_ERR_QMI_REQ
qmerrno.h, 1246
eQCWWAN_ERR_QMI_REQ_SCH
qmerrno.h, 1246
eQCWWAN_ERR_QMI_REQ_TO
qmerrno.h, 1246
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_RSP
qmerrno.h, 1246
eQCWWAN_ERR_QMI_RSP_TO
qmerrno.h, 1246
eQCWWAN_ERR_QMI_SEGMENT_ORDER
qmerrno.h, 1249
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG
qmerrno.h, 1249
eQCWWAN_ERR_QMI_SESSION_INACTIVE
qmerrno.h, 1248
eQCWWAN_ERR_QMI_SESSION_INVALID
qmerrno.h, 1248
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP
qmerrno.h, 1248
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND
qmerrno.h, 1249
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED
qmerrno.h, 1248
eQCWWAN_ERR_QMI_SMSC_ADDR
qmerrno.h, 1249
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE
qmerrno.h, 1249
eQCWWAN_ERR_QMI_TPDU_TYPE
qmerrno.h, 1249
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION
qmerrno.h, 1247
eQCWWAN_ERR_QMI_UNKNOWN
qmerrno.h, 1248
eQCWWAN_ERR_QMI_WIDTH
qmerrno.h, 1250
eQCWWAN_ERR_RESET
qmerrno.h, 1247
eQCWWAN_ERR_SWICM_AM_VERS_ERROR
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_END
qmerrno.h, 1250
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_SOCKET_IN_USE
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_START
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_TIMEOUT
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_V4DWN_V6DWN
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_V4DWN_V6UP
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_V4UP_V6DWN
qmerrno.h, 1249
eQCWWAN_ERR_SWICM_V4UP_V6UP
qmerrno.h, 1249
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED

qmerrno.h, 1250
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND
qmerrno.h, 1250
eQCWWAN_ERR_SWIDCS_END
qmerrno.h, 1250
eQCWWAN_ERR_SWIDCS_FILEIO_ERR
qmerrno.h, 1250
eQCWWAN_ERR_SWIDCS_IOCTL_ERR
qmerrno.h, 1250
eQCWWAN_ERR_SWIDCS_START
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_END
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_NOT_SIGNED
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_INVALID_PATH
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_OPENING_DIR
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_OPENING_FILE
qmerrno.h, 1250
eQCWWAN_ERR_SWIIM_START
qmerrno.h, 1250
eQCWWAN_ERR_SWISM_END
qmerrno.h, 1250
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND
qmerrno.h, 1250
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED
qmerrno.h, 1250
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG
qmerrno.h, 1250
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED
qmerrno.h, 1250
eQCWWAN_ERR_SWISMS_START
qmerrno.h, 1250
eQDL_HW_FLOW_DISABLE
lite-fw.h, 1099
eQDL_HW_FLOW_ENABLE
lite-fw.h, 1099
eQDL_HW_FLOW_INIT
lite-fw.h, 1099
eQDL_HW_FLOW_Unknown
lite-fw.h, 1099
eQDL_MODE_INIT
lite-fw.h, 1099
eQDL_MODE_TTYUSB
lite-fw.h, 1099
eQDL_MODE_UART0
lite-fw.h, 1099
eQDL_MODE_Unknown
lite-fw.h, 1099
eQMI_LOC_SESS_STATUS_FAILURE
loc.h, 1111
eQMI_LOC_SESS_STATUS_IN_PROGRESS
loc.h, 1111
eQMI_LOC_SESS_STATUS_SUCCESS
loc.h, 1111
eQMI_LOC_SESS_STATUS_TIMEOUT
loc.h, 1111
eQOS
common.h, 1013
eREQ
common.h, 1014
eRMS
common.h, 1014
eRSP
common.h, 1014
eSAR
common.h, 1014
eSDP_FWDWL_ERR_END
lite-fw.h, 1099
eSDP_FWDWL_ERR_FAIL
lite-fw.h, 1099
eSDP_FWDWL_ERR_FW_UPGRADE
lite-fw.h, 1099
eSDP_FWDWL_ERR_FW_VERSION_FAIL
lite-fw.h, 1099
eSDP_FWDWL_ERR_GENERAL
lite-fw.h, 1098
eSDP_FWDWL_ERR_INVALID_DEV
lite-fw.h, 1099
eSDP_FWDWL_ERR_INVALID_PATH
lite-fw.h, 1099

eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED
lite-fw.h, 1099
eSDP_FWDWL_ERR_PATH_TOO_LONG
lite-fw.h, 1099
eSDP_FWDWL_ERR_PRI_FAIL
lite-fw.h, 1099
eSDP_FWDWL_ERR_SDK
lite-fw.h, 1099
eSDP_FWDWL_ERR_SDP_TIMEOUT
lite-fw.h, 1099
eSDP_FWDWL_ERR_SET_CBK
lite-fw.h, 1099
eSDP_FWDWL_ERR_TIMEOUT
lite-fw.h, 1099
eSDP_FWDWL_SUCCESS
lite-fw.h, 1098
eSMS
common.h, 1013
eSWIAUDIO
common.h, 1014
eSWIDMS
common.h, 1014
eSWILOC
common.h, 1014
eSWIM2MCMD
common.h, 1014
eSWIM2MCMD_AVC2
common.h, 1014
eSWIOMA
common.h, 1014
eSWIOMAEXT
common.h, 1014
eTIMEOUT_10_S
common.h, 1014
eTIMEOUT_20_S
common.h, 1014
eTIMEOUT_2_S
common.h, 1014
eTIMEOUT_300_S
common.h, 1014
eTIMEOUT_30_S
common.h, 1014
eTIMEOUT_5_S
common.h, 1014
eTIMEOUT_60_S
common.h, 1014
eTIMEOUT_8_S
common.h, 1014
eTIMEOUT_DEFAULT
common.h, 1014
eTMD
common.h, 1014
eUIM
common.h, 1014
eVOICE
common.h, 1013
eWDS
common.h, 1013
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_END
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_FAIL
qmerrno.h, 1251
eWDS_ERR_PROFILE_REG_RESULT_LIST_END
qmerrno.h, 1251
EAWTlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
ECIOThresListLen

nas_ECIOThresh, 167
ECTCallState
voice_ECTNum, 960
eLOG_LEVEL
common.h, 1013
EMTlv
NASQmiCbkNasSystemSelPrefInd, 291
eQCWWANError
qmerrno.h, 1246
eQMI_SVC
common.h, 1013
ERI_DATA_MAX_SIZE
dms.h, 1024
ESATlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
ESAWTlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
ESNString
unpack_dms_GetDeviceSerialNumbers_t, 565
ETWSPLMNInfo
eTWSPLMNInfoTlv, 64
eTWSPLMNInfoTlv, 64
ETWSPLMNInfo, 64
TlvPresent, 64
ETWSPLMNTlv
unpack_sms_SetNewSMSCallback_ind_t, 784
ETWSTlv
unpack_sms_SetNewSMSCallback_ind_t, 785
eTimeout
common.h, 1014
eValid
LibPackTFTIDParams, 116
EarMute
pack_audio_SLQSSetAudioProfile_t, 299
pack_swiaudio_SLQSSetM2MAVMute_t, 412
unpack_audio_SLQSGetAudioProfile_t, 549
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 801
unpack_swiaudio_SLQSGetM2MAVMute_t, 803
earfcn
nas_infoInterFreq, 184
nas_LTEInfoIntrafreq, 194
nas_ltePCI, 201
nas_umtsLTENbrCell, 272
earfcn0
nas_lteEARFCN, 188
earfcn1
nas_lteEARFCN, 188
earlyWakeupTime
dms_PSMEarlyWakeupTimeTlv, 56
ecio
cdmaSSInfo, 49
hdrSSInfo, 68
nas_ecioListElement, 166
nas_rxInfo, 249
nas_UMTSInfo, 270
tdscdmaSigInfoExt, 519
ecioDelta
nas_SLQSSignalStrengthsIndReq, 256
ecioInfo
nas_SLQSSignalStrengthsInformation, 258
ecioList
unpack_nas_SLQSGetSignalStrength_t, 704
ecioListLen
unpack_nas_SLQSGetSignalStrength_t, 704
ecioThresholdList
nas_SLQSSignalStrengthsIndReq, 256
ecioThresholdListLen
nas_SLQSSignalStrengthsIndReq, 256
egprsSupp
nas_GSMSysInfo, 176
egprsSuppValid
nas_GSMSysInfo, 176
ehrpdMTUSize
swidms_ehrpdMTUSizeTlv, 517
elevation
loc_satelliteInfo, 135
EmerMode
nas_EmerModeTlv, 168
NASEmergencyModeTlv, 280
emmConnState
nas_LTEInfo, 191
emmState
nas_LTEInfo, 191
emmSubState
nas_LTEInfo, 191
Enable
pack_swiaudio_SLQSSetM2MAudioLPBK_t, 409
enable
pack_pds_SetPDSState_t, 385
pack_qos_SLQSSetQosEventCallback_t, 396
EnableState
unpack_dms_PSMCfgChange_ind_t, 581
enableState
dms_PSMEnableStateTlv, 57
EnableStateInd
dms_PSMEnableStateIndTlv, 56
enabled
pack_rms_SetSMSWake_t, 396
unpack_rms_GetSMSWake_t, 779
unpack_wds_GetMobileIPProfile_t, 888
EnabledStatus
unpack_pds_SetPdsState_Ind_t, 750
EncryptProt
nas_protocolSubtypeElement, 232
EncryptedPIN1
pack_uim_ChangePin_t, 428
pack_uim_SetPinProtection_t, 430
pack_uim_UnblockPin_t, 438
EndPointType
qos_BindDataPortPeripheralEndPointID_t, 491
EndProactiveSession
cat_EndProactiveSessionTlv, 47
EngineState
unpack_pds_SLQSGetGPSStateInfo_t, 755
engineState

unpack_loc_EngineState_Ind_t, 651
eqmiCbkSetStatus
sms.h, 1268
eriData
eriDataparams, 64
eriDataLen
eriDataparams, 64
eriDataparams, 63
eriData, 64
eriDataLen, 64
eriFile
unpack_dms_SLQSGetERIFile_t, 598
error
unpack_wds_GetLastMobileIPError_t, 886
errorClass
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 799
ErrorCode
imsa_IMSRegStatusErrorCodeInfo, 83
ErrorCodeData
imsa_RatHandoverStatusInfo, 85
ErrorCodeLen
imsa_RatHandoverStatusInfo, 85
errorMask
unpack_cat_SetCATEventCallback_t, 552
errorRate
nas_errorRateListElement, 169
errorRateInfo
nas_SLQSSignalStrengthsInformation, 258
errorRateList
unpack_nas_SLQSGetSignalStrength_t, 704
errorRateListLen
unpack_nas_SLQSGetSignalStrength_t, 704
errorState
slotInf, 497
uim_slotInfo, 544
esn
unpack_dms_GetSerialNumbers_t, 577
esnSize
unpack_dms_GetDeviceSerialNumbers_t, 565
EspSpi
unpack_qos_swiQosFilter_t, 771
EtwsMessageInfo
sMSEtwsMessageTlv, 513
event
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 767
unpack_qos_SLQSSetQosStatusCallback_ind_t, 769
event_Index
unpack_cat_SetCatEventCallback_ind_t, 551
EventID
cat_commonEventTlv, 46
EventLength
cat_commonEventTlv, 46
eventMask
pack_cat_SetCATEventCallback_t, 302
pack_uim_SLQSUIMEventRegister_t, 432
unpack_uim_SLQSUIMEventRegister_t, 831
eventRegister
pack_loc_EventRegister_t, 333
eventType
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 819
evrcCapability
voice_prefVoiceSO, 969
executingImage
FMSImageIDEntries, 66
exponent
unpack_qos_pktErrRate_t, 759
extBit
voice_calledPartySubAdd, 942
extDispInfo
voice_extDispRecInfo, 960
extDispInfoLen
voice_extDispRecInfo, 960
ExtErrorCode
PackCreateProfileOut, 484
extPowerState
pack_loc_SetExtPowerState_t, 333
extendedErrorCode
unpack_wds_SLQSDeleteProfile_t, 895

FLOAT
SwiDataTypes.h, 1311
FMSImageElement, 64
buildId, 65
buildIdLength, 65
imageId, 65
imageType, 65
FMSImageIDEntries, 66
executingImage, 66
imageIDElement, 66
imageIDSize, 66
imageType, 67
maxImages, 67
FMSImageIdElement, 65
buildID, 66
buildIDLength, 66
failureCount, 66
imageID, 66
storageIndex, 66
FMSImageList, 67
imageIDEntries, 67
listSize, 67
FMSPrefImageList, 67
listEntries, 68
listSize, 68
FOTAUpdate
pack_swioma_SLQSOMADMSetSettings_t, 424
pack_swioma_SLQSOMADMSetSettingsExt_t, 425
unpack_swioma_SLQSOMADMGetSettings_t, 823
FOTAdownload
pack_swioma_SLQSOMADMSetSettings_t, 424
pack_swioma_SLQSOMADMSetSettingsExt_t, 425

unpack_swioma_SLQSOMADMGetSettings_t, 823
FUMOEnable
pack_swioma_SLQSOMADMSetSettingsExt_t, 425
facility
pack_dms_UIMGetControlKeyStatus_t, 316
pack_dms_UIMSetControlKeyProtection_t, 317
pack_dms_UIMUnblockControlKey_t, 318
facilityCk
pack_dms_UIMSetControlKeyProtection_t, 317
pack_dms_UIMUnblockControlKey_t, 318
facilityState
pack_dms_UIMSetControlKeyProtection_t, 317
unpack_dms_UIMGetControlKeyStatus_t, 612
FailErrCode
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, 638
failureCount
FMSImageIdElement, 66
family
pack_wds_GetDefaultProfileNum_t, 459
pack_wds_SetDefaultProfileNum_t, 465
feature
uim_depersonalizationInformation, 530
uim_personalizationStatus, 536
fileID
uim_fileAttributes, 534
uim_fileInfo, 535
fileIndex
pack_uim_ReadTransparent_t, 429
pack_uim_SLQSUIMGetFileAttributes_t, 433
fileSize
uim_fileAttributes, 534
fileType
uim_fileAttributes, 534
fill_pack_ctx
common.h, 1015
fill_sdu_hdr
common.h, 1015
filterId
LibPackTFTIDParams, 117
fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 420
unpack_swiloc_SwiLocGetAutoStart_t, 818
fix_rate_reported
unpack_swiloc_SwiLocGetAutoStart_t, 818
fix_type
pack_swiloc_SwiLocSetAutoStart_t, 420
unpack_swiloc_SwiLocGetAutoStart_t, 818
fix_type_reported
unpack_swiloc_SwiLocGetAutoStart_t, 818
fixAccuracy
pack_pds_StartPDSTrackingSessionExt_t, 393
fixCount
pack_pds_StartPDSTrackingSessionExt_t, 393
fixInterval
pack_pds_StartPDSTrackingSessionExt_t, 393
fixTimeout
pack_pds_StartPDSTrackingSessionExt_t, 393
flags
sensorData_t, 494
flowLabel
LibPackTFTIDParams, 117
fms.h, 1074
GetValidFwPriCombinations, 1075
pack_fms_GetImagesPreference, 1075
pack_fms_GetStoredImages, 1076
pack_fms_SetImagesPreference, 1076
unpack_fms_GetImagesPreference, 1076
unpack_fms_GetStoredImages, 1076
unpack_fms_SetImagesPreference, 1076
Forbidden
nas_QmiNas3GppNetworkInfo, 234
forceFlag
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 388
ForceRev0
unpack_nas_GetCDMANetworkParameters_t, 674
format
sMSTransferRouteMTMessageInfo, 516
ForwardMac
nas_protocolSubtypeElement, 232
fqdnAddr
wds_PCSCFFQDNAddress, 990
fqdnLen
wds_PCSCFFQDNAddress, 990
freeSlots
sms_maxStorageSizeResp, 502
freq
nas_PhyCaAggPcellInfo, 223
nas_PhyCaAggScellIndType, 224
nas_PhyCaAggScellInfo, 227
NASPhyCaAggPcellInfo, 285
NASPhyCaAggScellIndType, 287
NASPhyCaAggScellInfo, 288
freqsLen
nas_LTEInfoInterfreq, 192
nas_LTEInfoNeighboringGSM, 195
nas_LTEInfoNeighboringWCDMA, 196
fromServiceId
sms_BroadcastConfig, 498
function
pack_swiloc_SwiLocSetAutoStart_t, 420
unpack_swiloc_SwiLocGetAutoStart_t, 818
function_reported
unpack_swiloc_SwiLocGetAutoStart_t, 818
FwAutoCheck
unpack_swioma_SLQSOMADMGetSettings_t, 823
FwAutoSDM
unpack_swiavms_SLQSAVMSGetSettings_t, 807
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
FwAvailability
unpack_swioma_SLQSOMADMStartSession_t, 823
FwPromptUpdate

unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
FwPromptdownload
unpack_swiavms_SLQSAVMSGetSettings_t, 807
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
fwdloadsize
unpack_omaDmFotaTlv_t, 741
fwloadComplete
unpack_omaDmFotaTlv_t, 741
fwvers
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603

gDlBitRate
LibPackQosClassID, 115
GPRSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 905
GPSLPM
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
GPSSel
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
GSMRSSIThreshListLen
nas_GSMRSSIThresh, 173
GSMSSInfo
unpack_nas_SLQSNasGetSigInfo_t, 722
gUlBitRate
LibPackQosClassID, 115
GWAOPTlv
NASQmiCbkNasSystemSelPrefInd, 291
GWAcqOrderPref
nas_GWAcqOrderPrefTlv, 177
NASGWAcqOrderPrefTlv, 282
GWAddressV4
unpack_wds_SLQSGetRuntimeSettings_t, 906
gcdumpString
crashInformation, 51
gcdumpStrlen
crashInformation, 51
Generator
pack_audio_SLQSGetAudioProfile_t, 295
pack_audio_SLQSGetAudioVolTLBConfig_t, 296
pack_audio_SLQSSetAudioProfile_t, 299
pack_audio_SLQSSetAudioVolTLBConfig_t, 300
pack_swiaudio_SLQSGetM2MAudioVolume_t, 407
pack_swiaudio_SLQSSetM2MAudioVolume_t, 411
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 801
geoSysIdx
nas_AddCDMASysInfo, 143
nas_AddSysInfo, 143
geranArfcn
nas_geranInstInfo, 171
geranBsicBcc
nas_geranInstInfo, 171
geranBsicNcc
nas_geranInstInfo, 171
geranInst
nas_UMTSInfo, 270
GeranInstInfo
nas_UMTSInfo, 270
geranRssi
nas_geranInstInfo, 171
get_version
common.h, 1015
getAllCallInfo
voice_arrCallInfo, 935
GetAllCallRmtPtyName
voice_arrRemotePartyName, 937
GetCustomFeatureV2
unpack_dms_GetCustFeaturesV2_t, 559
GetValidFwPriCombinations
fms.h, 1075
glo_almanac_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
glo_ephemeris_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
glo_health_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
glo_visible_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
globalCellId
nas_LTEInfoIntrafreq, 194
glog
common.h, 1016
gloglvl
common.h, 1016
gnssSvId
loc_satelliteInfo, 135
gnssSvUsedList
loc_svUsedforFix, 138
gnssSvUsedList_len
loc_svUsedforFix, 138
Gpp2TimeZone
unpack_nas_SLQSGetServingSystem_t, 702
GppNetworkDSTAdjustment
unpack_nas_SLQSGetServingSystem_t, 702
GppTimeZone
unpack_nas_SLQSGetServingSystem_t, 702
gps_almanac_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
gps_ephemeris_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
gps_health_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
gps_visible_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
GpsEnable
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
gpsTimeOfWeekMs
loc_gpsTime, 127
gpsWeek
loc_gpsTime, 127
grntDownlinkBitrate

LibPackUMTSQoS, 119
wds_UMTSMinQoS, 996
grntUplinkBitrate
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
gsmAmrStat
voice_curAMRConfig, 957
GsmCellInfo
nas_lteGsmCellInfo, 189
gsmUmtsDl
pack_nas_SLQSNasSwiIndicationRegister_t, 363
gsmUmtsUl
pack_nas_SLQSNasSwiIndicationRegister_t, 363
guaranteedRate
unpack_qos_dataRate_t, 757
gwAddressV6
wds_IPV6GWAddressInfo, 989
gwV6PrefixLen
wds_IPV6GWAddressInfo, 989
gyroData
pack_loc_SLQSLOCInjectSensorData_t, 341
gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 341
gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 342

HASPI
unpack_wds_GetMobileIPProfile_t, 888
HAState
unpack_wds_GetMobileIPProfile_t, 888
HDOP
loc_precisionDilution, 133
HDRECIOThreshListLen
nas_HDRECIOThresh, 178
HDRIOThreshListLen
nas_HDRIOThresh, 178
HDRRSSIThreshListLen
nas_HDRRSSIThresh, 179
HDRSINRThresListLen
nas_HDRSINRThresh, 180
HDRSINRThreshListLen
nas_HDRSINRThreshold, 181
HDRSSInfo
unpack_nas_SLQSNasGetSigInfo_t, 722
HWVersion
nas_DeviceConfigDetail, 165
HardwareControlledMode
unpack_dms_GetPower_t, 575
has_LpmFlag
unpack_dms_SLQSDmsSwiGetPCInfo_t, 589
has_PdpStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
has_PersistentLpm
unpack_dms_SLQSDmsSwiGetPCInfo_t, 589
has_PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 749
has_PowerOffMode
unpack_dms_SLQSDmsSwiGetPCInfo_t, 589
has_RatHandoverStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
has_RegStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
has_ServiceStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
has_SessionStatus
unpack_pds_SetEventReport_Ind_t, 749
has_Wdisable
unpack_dms_SLQSDmsSwiGetPCInfo_t, 589
has_accelTemp
pack_loc_SLQSLOCInjectSensorData_t, 342
has_acceleroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 342
has_accleroData
pack_loc_SLQSLOCInjectSensorData_t, 342
has_altitudeSrcInfo
pack_loc_SLQSLOCInjectPosition_t, 339
has_altitudeWrtEllipsoid
pack_loc_SLQSLOCInjectPosition_t, 339
has_altitudeWrtMeanSeaLevel
pack_loc_SLQSLOCInjectPosition_t, 339
has_confidence
pack_loc_SLQSLOCSetCradleMountConfig_t, 343
has_dLatitude
unpack_pds_SetEventReport_Ind_t, 749
has_dLongitude
unpack_pds_SetEventReport_Ind_t, 749
has_gyroData
pack_loc_SLQSLOCInjectSensorData_t, 342
has_gyroTemp
pack_loc_SLQSLOCInjectSensorData_t, 342
has_gyroTimeSrc
pack_loc_SLQSLOCInjectSensorData_t, 342
has_horConfidence
pack_loc_SLQSLOCInjectPosition_t, 339
has_horReliability
pack_loc_SLQSLOCInjectPosition_t, 339
has_horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 339
has_latitude
pack_loc_SLQSLOCInjectPosition_t, 339
has_longitude
pack_loc_SLQSLOCInjectPosition_t, 339
has_opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 342
has_posSrc
unpack_pds_SetEventReport_Ind_t, 749
has_positionSrc
pack_loc_SLQSLOCInjectPosition_t, 339
has_rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 339
has_rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 339
has_serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 335
has_timestampAge
pack_loc_SLQSLOCInjectPosition_t, 339
has_timestampUtc

pack_loc_SLQSLOCInjectPosition_t, 339
has_vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 339
has_vertRelicability
pack_loc_SLQSLOCInjectPosition_t, 339
has_vertUnc
pack_loc_SLQSLOCInjectPosition_t, 340
hdrActiveProt
nas_HDRSysInfo, 183
hdrActiveProtValid
nas_HDRSysInfo, 183
hdrHybrid
nas_detailSvcInfo, 164
HdrPersonality
unpack_nas_SLQSGetServingSystem_t, 702
hdrPersonality
nas_HDRSysInfo, 183
NASServingSystemInfo, 293
hdrPersonalityValid
nas_HDRSysInfo, 183
hdrSSInfo, 68
ecio, 68
io, 68
rssi, 69
sinr, 69
hdrSrvStatus
nas_detailSvcInfo, 164
healthStatus
loc_satelliteInfo, 135
helper_get_error_code
common.h, 1015
helper_get_error_reason
common.h, 1015
helper_get_req_str
common.h, 1015
helper_get_resp_ctx
common.h, 1015
helper_get_xid
common.h, 1015
helper_set_log_func
common.h, 1015
helper_set_log_lvl
common.h, 1015
homeOrigVoiceSO
voice_prefVoiceSO, 969
homePageVoiceSO
voice_prefVoiceSO, 969
horConfidence
pack_loc_SLQSLOCInjectPosition_t, 340
horReliability
pack_loc_SLQSLOCInjectPosition_t, 340
horUncCircular
pack_loc_SLQSLOCInjectPosition_t, 340
HorizontalUncertainty
unpack_pds_SLQSGetGPSStateInfo_t, 755
hotSwap
uim_hotSwapStatus, 535
hotSwapLength
uim_hotSwapStatus, 535
hour
nas_timeInfo, 267
nas_UniversalTime, 273
hrpdMTUSize
swidms_hrpdMTUSizeTlv, 517
hsCallStatus
nas_WCDMASysInfo, 279
hsCallStatusValid
nas_WCDMASysInfo, 279
hsInd
nas_WCDMASysInfo, 279
hsIndValid
nas_WCDMASysInfo, 279
hwType
wds_DHCPv4HWConfig, 983
wdsDhcpv4HwConfig, 997
hwVer
unpack_dms_GetHardwareRevision_t, 570

IDTlv
unpack_ims_SLQSUserCfgCallBack_ind_t, 632
iFaceTab
swiaudio_PCMparams, 516
iFaceTabLen
swiaudio_PCMparams, 516
iLTEbandValue
nas_PhyCaAggPcellInfo, 223
nas_PhyCaAggScellInfo, 227
NASPhyCaAggPcellInfo, 285
NASPhyCaAggScellInfo, 288
IMCNflag
unpack_wds_SLQSGetRuntimeSettings_t, 906
IMEIString
unpack_dms_GetDeviceSerialNumbers_t, 565
IMG_MASK_CLEAR
lite-fw.h, 1097
IMG_MASK_GENERIC
lite-fw.h, 1097
IMG_MASK_MDM
lite-fw.h, 1097
IMG_MASK_PRI
lite-fw.h, 1097
IMSInfo
sMSOnIMSTlv, 515
IMSRegistration
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 640
IMSRegistrationError
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 640
IMSTlv
unpack_sms_SetNewSMSCallback_ind_t, 785
INT32
SwiDataTypes.h, 1311
INT8
SwiDataTypes.h, 1311
IOThresListLen
nas_IOThresh, 185

IPAddress
nas_DataStatusDetail, 162
IPAddressV6
ipv6AddressInfo, 90
wds_IPV6AddressInfo, 988
IPFamSupport
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
IPFamilyPreference
pack_wds_SLQSSetIPFamilyPreference_t, 477
unpack_wds_SLQSGetRuntimeSettings_t, 906
IPSECSPI
LibPackTFTIDParams, 117
IPV6AddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 906
IPV6GWAddrInfo
unpack_wds_SLQSGetRuntimeSettings_t, 906
IPV6PrefixLen
ipv6AddressInfo, 90
wds_IPV6AddressInfo, 988
IPv4
unpack_wds_SLQSGetRuntimeSettings_t, 906
IPv4Addr
loc_IPv4Config, 129
wds_IPv4AdTlv, 988
IPv4AddrTlv
unpack_wds_DHCPv4ClientLease_ind_t, 880
IPv4DstAddr
unpack_qos_swiQosFilter_t, 771
IPv4Port
loc_IPv4Config, 129
IPv4SrcAddr
unpack_qos_swiQosFilter_t, 771
IPv4Tos
unpack_qos_swiQosFilter_t, 771
IPv6Addr
loc_IPv6Config, 130
IPv6DstAddr
unpack_qos_swiQosFilter_t, 771
IPv6Label
unpack_qos_swiQosFilter_t, 771
IPv6Port
loc_IPv6Config, 130
IPv6SrcAddr
unpack_qos_swiQosFilter_t, 771
IPv6TrafCls
unpack_qos_swiQosFilter_t, 771
Id
unpack_qos_swiQosFilter_t, 771
id
loc_BdsSV, 122
loc_SV, 137
nas_CSGID, 159
pack_dms_UIMChangePIN_t, 315
pack_dms_UIMSetPINProtection_t, 318
pack_dms_UIMUnblockPIN_t, 319
pack_dms_UIMVerifyPIN_t, 319
unpack_qos_QosFlowInfoState_t, 761
unpack_qos_SLQSSetQosStatusCallback_ind_t, 769
id_length
DMScustSettingInfo, 60
IdleState
nas_protocolSubtypeElement, 232
IfaceID
qos_BindDataPortPeripheralEndPointID_t, 491
image_info_t, 69
buildID, 69
buildIDLen, 69
imageType, 69
uniqueID, 69
imageID
FMSImageIdElement, 66
imageIDElement
FMSImageIDEntries, 66
imageIDEntries
FMSImageList, 67
imageIDSize
FMSImageIDEntries, 66
imageId
FMSImageElement, 65
imageList
unpack_fms_GetStoredImages_t, 618
ImageListSize
unpack_fms_GetImagesPreference_t, 617
imageListSize
pack_fms_SetImagesPreference_t, 321
imageType
FMSImageElement, 65
FMSImageIDEntries, 67
image_info_t, 69
ImageTypes
unpack_fms_SetImagesPreference_t, 619
ImageTypesSize
unpack_fms_SetImagesPreference_t, 619
imagelistSize
unpack_fms_GetStoredImages_t, 618
imei_no
unpack_dms_GetSerialNumbers_t, 577
imeiSize
unpack_dms_GetDeviceSerialNumbers_t, 565
imeiSvnSize
unpack_dms_GetDeviceSerialNumbers_t, 565
ImeiSvnString
unpack_dms_GetDeviceSerialNumbers_t, 565
imeisv_svn
unpack_dms_GetSerialNumbers_t, 577
imgType
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
ims.h, 1077
MAX_NAME_LEN, 1079
pack_ims_SLQSGetIMSSMSConfig, 1079
pack_ims_SLQSGetIMSUserConfig, 1079
pack_ims_SLQSGetIMSVoIPConfig, 1080
pack_ims_SLQSGetRegMgrConfig, 1080
pack_ims_SLQSGetSIPConfig, 1080

pack_ims_SLQSImsConfigIndicationRegister, 1081
pack_ims_SLQSSetIMSSMSConfig, 1081
pack_ims_SLQSSetIMSUserConfig, 1082
pack_ims_SLQSSetIMSVoIPConfig, 1082
pack_ims_SLQSSetRegMgrConfig, 1083
pack_ims_SLQSSetSIPConfig, 1083
unpack_ims_SLQSGetIMSSMSConfig, 1084
unpack_ims_SLQSGetIMSUserConfig, 1084
unpack_ims_SLQSGetIMSVoIPConfig, 1084
unpack_ims_SLQSGetRegMgrConfig, 1085
unpack_ims_SLQSGetSIPConfig, 1085
unpack_ims_SLQSImsConfigIndicationRegister, 1085
unpack_ims_SLQSImsConfigIndicationRegister_t, 1079
unpack_ims_SLQSRegMgrCfgCallBack_ind, 1086
unpack_ims_SLQSSIPCfgCallBack_ind, 1088
unpack_ims_SLQSSMSCfgCallBack_ind, 1088
unpack_ims_SLQSSetIMSSMSConfig, 1086
unpack_ims_SLQSSetIMSUserConfig, 1086
unpack_ims_SLQSSetIMSVoIPConfig, 1087
unpack_ims_SLQSSetRegMgrConfig, 1087
unpack_ims_SLQSSetSIPConfig, 1087
unpack_ims_SLQSUserCfgCallBack_ind, 1088
unpack_ims_SLQSVoIPCfgCallBack_ind, 1089
ims_AMRModeInfo, 69
amrMode, 70
TlvPresent, 70
ims_AMROctAlgnInfo, 70
amrOctAlgn, 70
TlvPresent, 70
ims_AMRWBModeInfo, 71
amrWBMode, 71
TlvPresent, 71
ims_AMRWBOctAlgnInfo, 71
amrWBOctAlgn, 72
TlvPresent, 72
ims_CSCFPortNameInfo, 72
cscfPortName, 72
TlvPresent, 72
ims_EnabAMRWBInfo, 72
amrWBEnable, 72
TlvPresent, 72
ims_EnabSCRAMRInfo, 73
scrAmrEnable, 73
TlvPresent, 73
ims_EnabSCRAMRWBInfo, 73
scrAmrWBEnable, 73
TlvPresent, 73
ims_IMSDomainInfo, 74
imsDomainName, 74
TlvPresent, 74
ims_IMSTestModeInfo, 74
imsTestMode, 74
TlvPresent, 74
ims_MinSessExpInfo, 74
minSessExp, 75
TlvPresent, 75
ims_PCSCFPortInfo, 75
priCSCFPort, 75
TlvPresent, 75
ims_PhCtxtURIInfo, 75
PhCtxtURI, 76
TlvPresent, 76
ims_RTPRTCPInactTmrDurInfo, 77
InactTmr, 77
TlvPresent, 77
ims_RngBkTmrInfo, 76
RingBkTmr, 76
TlvPresent, 76
ims_RngTmrInfo, 76
RingTmr, 77
TlvPresent, 77
ims_SIPPortInfo, 78
SIPLocalPort, 79
TlvPresent, 79
ims_SIPRegnTmrInfo, 79
TlvPresent, 79
tmrSIPRegn, 79
ims_SMSFmtInfo, 79
smsFormat, 80
TlvPresent, 80
ims_SMSoIPNwInfo, 80
smsoIPNW, 80
TlvPresent, 80
ims_SessDurInfo, 77
sessExp, 78
TlvPresent, 78
ims_SigCompEnInfo, 78
SigCompEn, 78
TlvPresent, 78
ims_SubscrTmrInfo, 80
subscrTmr, 81
TlvPresent, 81
ims_TmrT1Info, 81
TlvPresent, 81
tmrT1, 81
ims_TmrT2Info, 81
TlvPresent, 82
tmrT2, 82
ims_TmrTfInfo, 82
TlvPresent, 82
tmrTf, 82
imsDomainName
ims_IMSDomainInfo, 74
ImsFailErrCode
imsa_IMSFailErrCodeTlv, 83
ImsRegErrCode
unpack_imsa_SLQSGetIMSARegStatus_t, 635
imsRegState
nas_CommInfo, 158
ImsRegStatus
imsa_NewIMSRegStatusInfo, 84
unpack_imsa_SLQSGetIMSARegStatus_t, 635
ImsRegistered

imsa_IMSRegStatusInfo, 84
imsTestMode
ims_IMSTestModeInfo, 74
imsa.h, 1089
MAX_ERROR_CODE_LEN, 1090
pack_imsa_SLQSGetIMSARegStatus, 1090
pack_imsa_SLQSGetIMSAServiceStatus, 1091
pack_imsa_SLQSRegisterIMSAIndication, 1091
unpack_imsa_SLQSGetIMSARegStatus, 1092
unpack_imsa_SLQSGetIMSAServiceStatus, 1092
unpack_imsa_SLQSImsaPdpStatusCallBack_ind, 1093
unpack_imsa_SLQSImsaRatStatusCallBack_ind, 1093
unpack_imsa_SLQSImsaRegStatusCallBack_ind, 1093
unpack_imsa_SLQSImsaSvcStatusCallBack_ind, 1094
unpack_imsa_SLQSRegisterIMSAIndication, 1094
unpack_imsa_SLQSRegisterIMSAIndication_t, 1090
imsa_IMSFailErrCodeTlv, 82
ImsFailErrCode, 83
TlvPresent, 83
imsa_IMSRegStatusErrorCodeInfo, 83
ErrorCode, 83
TlvPresent, 83
imsa_IMSRegStatusInfo, 83
ImsRegistered, 84
TlvPresent, 84
imsa_NewIMSRegStatusInfo, 84
ImsRegStatus, 84
TlvPresent, 84
imsa_RatHandoverStatusInfo, 84
ErrorCodeData, 85
ErrorCodeLen, 85
RatHandoverStatus, 85
SourceRAT, 85
TargetRAT, 85
TlvPresent, 85
imsa_SmsRatInfo, 85
SmsRatVal, 86
TlvPresent, 86
imsa_SmsSvcStatusInfo, 86
SmsSvcStatus, 86
TlvPresent, 86
imsa_UtRatInfo, 86
TlvPresent, 87
UtRatVal, 87
imsa_UtSvcStatusInfo, 87
TlvPresent, 87
UtSvcStatus, 87
imsa_VoipRatInfo, 88
TlvPresent, 88
VoipRatVal, 88
imsa_VoipSvcStatusInfo, 88
TlvPresent, 88
VoipSvcStatus, 88
imsa_VtRatInfo, 89
TlvPresent, 89
VtRatVal, 89
imsa_VtSvcStatusInfo, 89
TlvPresent, 90
VtSvcStatus, 90
imsi
unpack_dms_GetIMSI_t, 571
imsi_11_12
nas_CDMASysInfoExt, 153
imsiM1112
nas_minBasedIMSI, 210
imsiMS1
nas_minBasedIMSI, 210
imsiMS2
nas_minBasedIMSI, 210
imsiT1112
nas_trueIMSI, 268
imsiTS1
nas_trueIMSI, 268
imsiTS2
nas_trueIMSI, 268
imsiTaddrNum
nas_trueIMSI, 268
InUse
nas_QmiNas3GppNetworkInfo, 234
InactTmr
ims_RTPRTCPInactTmrDurInfo, 77
includes_pcs_digit
nas_QmisNasPcsDigit, 236
index
pack_wds_GetMobileIPProfile_t, 461
pack_wds_SetDefaultProfileNum_t, 465
pack_wds_SetMobileIPProfile_t, 468
unpack_qos_swiQosFilter_t, 771
unpack_qos_swiQosFlow_t, 775
unpack_wds_GetDefaultProfileNum_t, 885
index1xPri
uim_cardStatus, 528
index1xSec
uim_cardStatus, 528
indexGwPri
uim_cardStatus, 528
indexGwSec
uim_cardStatus, 528
Info
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 727
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 729
InfoInterfreq
nas_LTEInfoInterfreq, 192
injectEnable
loc_accelAcceptReady, 121
loc_accelTempAcceptReady, 122
loc_gyroAcceptReady, 128
loc_gyroTempAcceptReady, 128
injectSensorDataStatus

unpack_loc_InjectSensorDataCallback_Ind_t, 657
injectTimeSyncStatus
unpack_loc_InjectTimeSyncDataCallback_Ind_t, 658
insNmrCellInfo
nas_GERANInfo, 170
instancesSize
unpack_nas_GetRFInfo_t, 680
interval
pack_pds_SetPDSDefaults_t, 385
pack_pds_SetXTRAAutomaticDownload_t, 387
pack_wds_SLQSSetWdsEventCallback_t, 478
Io
unpack_nas_SLQSGetSignalStrength_t, 704
io
hdrSSInfo, 68
nas_SLQSSignalStrengthsInformation, 258
ioDelta
nas_SLQSSignalStrengthsIndReq, 257
Iono_valid
unpack_pds_SLQSGetGPSStateInfo_t, 755
ipAddress
pack_wds_SetDefaultProfile_t, 465
ipFamily
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 909
ipVersion
LibPackTFTIDParams, 117
ipaddr
unpack_wds_GetDefaultProfile_t, 884
ipaddrv6
unpack_wds_GetDefaultProfile_t, 884
ipv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
ipv4GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
ipv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
ipv6AddressInfo, 90
IPAddressV6, 90
IPV6PrefixLen, 90
ipv6GWAddress
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
is856SysId
nas_HDRSysInfo, 183
is856SysIdValid
nas_HDRSysInfo, 183
is_DataRate_Available
unpack_qos_swiQosFlow_t, 775
is_EspSpi_Available
unpack_qos_swiQosFilter_t, 771
is_IPv4DstAddr_Available
unpack_qos_swiQosFilter_t, 771
is_IPv4SrcAddr_Available
unpack_qos_swiQosFilter_t, 771
is_IPv4Tos_Available
unpack_qos_swiQosFilter_t, 771
is_IPv6DstAddr_Available
unpack_qos_swiQosFilter_t, 771
is_IPv6Label_Available
unpack_qos_swiQosFilter_t, 771
is_IPv6SrcAddr_Available
unpack_qos_swiQosFilter_t, 771
is_IPv6TrafCls_Available
unpack_qos_swiQosFilter_t, 771
is_Id_Available
unpack_qos_swiQosFilter_t, 771
is_Jitter_Available
unpack_qos_swiQosFlow_t, 775
is_Latency_Available
unpack_qos_swiQosFlow_t, 775
is_LteBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 597
is_LteQci_Available
unpack_qos_swiQosFlow_t, 775
is_MaxAllowedPktSz_Available
unpack_qos_swiQosFlow_t, 775
is_MinPolicedPktSz_Available
unpack_qos_swiQosFlow_t, 775
is_NxtHdrProto_Available
unpack_qos_swiQosFilter_t, 772
is_PktErrRate_Available
unpack_qos_swiQosFlow_t, 775
is_Precedence_Available
unpack_qos_swiQosFilter_t, 772
is_ProfileId3GPP2_Available
unpack_qos_swiQosFlow_t, 775
is_RxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 760
is_TCPDstPort_Available
unpack_qos_swiQosFilter_t, 772
is_TCPSrcPort_Available
unpack_qos_swiQosFilter_t, 772
is_TdsBandCapability_Available
unpack_dms_SLQSGetBandCapability_t, 597
is_TokenBucket_Available
unpack_qos_swiQosFlow_t, 775
is_TrafficClass_Available
unpack_qos_swiQosFlow_t, 775
is_TranDstPort_Available
unpack_qos_swiQosFilter_t, 772
is_TranSrcPort_Available
unpack_qos_swiQosFilter_t, 772
is_TxQFlowGranted_Available
unpack_qos_QosFlowInfo_t, 760
is_UDPDstPort_Available
unpack_qos_swiQosFilter_t, 772
is_UDPSrcPort_Available
unpack_qos_swiQosFilter_t, 772
is_val_3GPP2Pri_Available
unpack_qos_swiQosFlow_t, 775
is_val_3GPPImCn_Available

unpack_qos_swiQosFlow_t, 775
is_val_3GPPResResidualBER_Available
unpack_qos_swiQosFlow_t, 775
is_val_3GPPSigInd_Available
unpack_qos_swiQosFlow_t, 776
is_val_3GPPTraHdlPri_Available
unpack_qos_swiQosFlow_t, 776
isEmpty
voice_getAllCallInformation, 961
isInTraffic
nas_txInfo, 268
isModByCC
voice_SUPSInfo, 974
isNewFlow
unpack_qos_QosFlowInfoState_t, 761
isPrefDataPath
nas_GSMSrvStatusInfo, 174
nas_SrvStatusInfo, 260
isRadioTuned
nas_rxInfo, 249
nas_RxSigInfo, 250
isSysForbidden
nas_detailSvcInfo, 164
nas_sysInfoCommon, 262
isSysForbiddenValid
nas_sysInfoCommon, 262
isSysPrlMatch
nas_CDMASysInfo, 152
nas_HDRSysInfo, 183
isSysPrlMatchValid
nas_CDMASysInfo, 152
nas_HDRSysInfo, 183
IsVoiceEnabled
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
Item
pack_audio_SLQSGetAudioPathConfig_t, 295
pack_audio_SLQSGetAudioVolTLBConfig_t, 296
pack_audio_SLQSSetAudioVolTLBConfig_t, 300

Jitter
unpack_qos_swiQosFlow_t, 776

KeyExchange
nas_protocolSubtypeElement, 232

LITE_VOICE_SUPS_SRV_CLASS_DATA
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_FAX
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_NONE
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_PADACCESS
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_SMS
voice.h, 1360
LITE_VOICE_SUPS_SRV_CLASS_VOICE
voice.h, 1360
LITEQMI_QMI_CBK_PARAM_NOCHANGE
sms.h, 1269
LITEQMI_QMI_CBK_PARAM_RESET
sms.h, 1269
LITEQMI_QMI_CBK_PARAM_SET
sms.h, 1269
LBPTlv
NASQmiCbkNasSystemSelPrefInd, 291
LMCERTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
LOCEVENTMASKGNSSSVINFO
loc.h, 1109
LOCEVENTMASKNMEA
loc.h, 1110
LOCEVENTMASKWIFIREQ
loc.h, 1111
LPCSTR
SwiDataTypes.h, 1311
LTEAttachProfile
unpack_wds_SLQSGet3GPPConfigItem_t, 897
LTEAttachProfileList
unpack_wds_SLQSGet3GPPConfigItem_t, 897
LTEAttachProfileListLen
pack_wds_SLQSSet3GPPConfigItem_t, 476
unpack_wds_SLQSGet3GPPConfigItem_t, 897
LTEBandPref
nas_LTEBandPrefTlv, 187
NASLTEBandPreferenceTlv, 282
LTECphyCaInfo
unpack_nas_SlqsGetLTECphyCAInfo_t, 692
LTERSRPThreshListLen
nas_LTERSRPThresh, 201
LTERSRQThreshListLen
nas_LTERSRQThresh, 202
LTERSSIThreshListLen
nas_LTERSSIThresh, 203
LTESNRThresListLen
nas_LTESNRThresh, 205
LTESNRThreshListLen
nas_LTESNRThreshold, 206
LTESSInfo
unpack_nas_SLQSNasGetSigInfo_t, 722
LTEWCDMACellInfo
nas_LTEInfoNeighboringWCDMA, 196
Lac
unpack_nas_SLQSGetServingSystem_t, 702
lac
nas_GERANInfo, 170
nas_GSMSysInfo, 176
nas_LTESysInfo, 208

nas_UMTSInfo, 270
nas_WCDMASysInfo, 279
lac1
nas_OperatorPLMNData, 221
lac2
nas_OperatorPLMNData, 221
lacValid
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_WCDMASysInfo, 279
language
sms_CDMABroadcastConfig, 498
lastCallDataBearerTech
unpack_wds_SLQSGetDUNCallInfo_t, 903
lastCallDataBearerTechnology
unpack_wds_SLQSGetDataBearerTechnology_t, 900
lastCallRXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 903
lastCallTXOKBytesCnt
unpack_wds_SLQSGetDUNCallInfo_t, 903
LastErrCode
nas_DataStatusDetail, 162
Latency
unpack_qos_swiQosFlow_t, 776
Latitude
unpack_pds_SLQSGetGPSStateInfo_t, 755
latitude
pack_loc_SLQSLOCInjectPosition_t, 340
leapSeconds
nas_qaQmi3Gpp2TimeZone, 233
leaseState
wds_DHCPLeaseStateTlv, 981
len
loc_BdsSVInfo, 123
loc_SVInfo, 137
unpack_nas_GetSignalStrengths_t, 684
length
sMSCAddressInfo, 511
sMSEtwsMessageInfo, 512
sMSTransferRouteMTMessageInfo, 516
uim_readTransparentInfo, 537
Level
pack_swiaudio_SLQSSetM2MAudioVolume_t, 411
unpack_swiaudio_SLQSGetM2MAudioVolume_t, 802
LibPackGPRSRequestedQoS, 90
delayClass, 91
meanThroughputClass, 91
peakThroughputClass, 91
precedenceClass, 91
reliabilityClass, 91
LibPackQosClassID, 114
gDlBitRate, 115
gUlBitRate, 115
maxDlBitRate, 115
maxUlBitRate, 115
QCI, 115
LibPackTFTIDParams, 115
destPortRangeEnd, 116
destPortRangeStart, 116
eValid, 116
filterId, 117
flowLabel, 117
IPSECSPI, 117
ipVersion, 117
nextHeader, 117
pSourceIP, 117
sourceIPMask, 117
srcPortRangeEnd, 117
srcPortRangeStart, 117
tosMask, 117
LibPackUMTSQoS, 117
deliveryErrSDU, 119
grntDownlinkBitrate, 119
grntUplinkBitrate, 119
maxDownlinkBitrate, 119
maxSDUSize, 119
maxUplinkBitrate, 119
qosDeliveryOrder, 119
resBerRatio, 119
sduErrorRatio, 119
trafficClass, 119
trafficPriority, 119
transferDelay, 119
LibPackUMTSReqQoSSigInd, 120
SigInd, 120
UMTSReqQoS, 120
LibPackprofile_3GPP, 103
pAPNClass, 108
pAPNDisabledFlag, 108
pAPNName, 108
pAPNnameSize, 108
pAddrAllocPref, 107
pAuthenticationPref, 108
pGPRSMinimumQoS, 108
pGPRSRequestedQos, 108
pIPv4AddrPref, 108
pIPv6AddPref, 108
pImCnFlag, 108
pPDNInactivTimeout, 108
pPDPtype, 108
pPassword, 108
pPasswordSize, 108
pPcscfAddrUsingDhcp, 108
pPcscfAddrUsingPCO, 108
pPdpAccessConFlag, 108
pPdpContext, 108
pPdpDataCompType, 108
pPdpHdrCompType, 108
pPriDNSIPv4AddPref, 108
pPriDNSIPv6addpref, 108
pPrimaryID, 108
pProfilename, 108
pProfilenameSize, 108
pQosClassID, 108

pSecDNSIPv4AddPref, 108
pSecDNSIPv6addpref, 108
pSecondaryFlag, 109
pSupportEmergencyCalls, 109
pTFTID1Params, 109
pTFTID2Params, 109
pUMTSMinQoS, 109
pUMTSMinQosSigInd, 109
pUMTSReqQoS, 109
pUMTSReqQoSSigInd, 109
pUsername, 109
pUsernameSize, 109
LibPackprofile_3GPP2, 109
pAPNClass3GPP2, 113
pAPNEnabled3GPP2, 113
pAllowLinger, 113
pApnString, 113
pApnStringSize, 113
pAppPriority, 113
pAppType, 113
pAuthPassword, 113
pAuthPassword_tSize, 113
pAuthProtocol, 113
pAuthRetryCount, 113
pAuthTimeout, 113
pDataMode, 113
pDataRate, 113
pIpcpAckTimeout, 113
pIpcpCreqRetryCount, 114
pIsPcscfAddressNedded, 114
pLcpAckTimeout, 114
pLcpCreqRetryCount, 114
pNegoDnsSrvrPref, 114
pPDNInactivTimeout3GPP2, 114
pPdnType, 114
pPppSessCloseTimer1x, 114
pPppSessCloseTimerDO, 114
pPriV6DnsAddress, 114
pPrimaryV4DnsAddress, 114
pRATType, 114
pSecV6DnsAddress, 114
pSecondaryV4DnsAddress, 114
pUserId, 114
pUserIdSize, 114
libSDP_BuildImagesPreferenceRequest
lite-fw.h, 1097
libSDP_CalculateImageMask
lite-fw.h, 1097
libSDP_CheckValidFirmwareInfo
lite-fw.h, 1097
libSDP_DownloadFW
lite-fw.h, 1097
libSDP_ExtractFirmwareParametersByPath
lite-fw.h, 1097
libSDP_FirmwareInfo
lite-fw.h, 1097
libSDP_GetModelFamily
lite-fw.h, 1097
libSDP_GetVersion
lite-fw.h, 1097
libSDP_getFileType
lite-fw.h, 1097
LibpackProfile3GPP, 91
pAPNClass, 96
pAPNDisabledFlag, 96
pAPNName, 96
pAPNnameSize, 96
pAddrAllocPref, 96
pAuthenticationPref, 96
pGPRSMinimumQoS, 96
pGPRSRequestedQos, 96
pIPv4AddrPref, 97
pIPv6AddPref, 97
pImCnFlag, 97
pPDNInactivTimeout, 97
pPDPtype, 97
pPassword, 97
pPasswordSize, 97
pPcscfAddrUsingDhcp, 97
pPcscfAddrUsingPCO, 97
pPdpAccessConFlag, 97
pPdpContext, 97
pPdpDataCompType, 97
pPdpHdrCompType, 97
pPriDNSIPv4AddPref, 97
pPriDNSIPv6addpref, 97
pPrimaryID, 97
pProfilename, 97
pProfilenameSize, 97
pQosClassID, 97
pSecDNSIPv4AddPref, 97
pSecDNSIPv6addpref, 97
pSecondaryFlag, 97
pSupportEmergencyCalls, 97
pTFTID1Params, 97
pTFTID2Params, 97
pUMTSMinQoS, 97
pUMTSMinQosSigInd, 97
pUMTSReqQoS, 97
pUMTSReqQoSSigInd, 98
pUsername, 98
pUsernameSize, 98
LibpackProfile3GPP2, 98
pAPNClass3GPP2, 102
pAPNEnabled3GPP2, 102
pAllowLinger, 102
pApnString, 102
pApnStringSize, 102
pAppPriority, 102
pAppType, 102
pAuthPassword, 102
pAuthPasswordSize, 102
pAuthProtocol, 102
pAuthRetryCount, 102
pAuthTimeout, 102
pDataMode, 102

pDataRate, 102
pIpcpAckTimeout, 102
pIpcpCreqRetryCount, 102
pIsPcscfAddressNedded, 102
pLcpAckTimeout, 102
pLcpCreqRetryCount, 102
pNegoDnsSrvrPref, 103
pPDNInactivTimeout3GPP2, 103
pPdnType, 103
pPppSessCloseTimer1x, 103
pPppSessCloseTimerDO, 103
pPriV6DnsAddress, 103
pPrimaryV4DnsAddress, 103
pRATType, 103
pSecV6DnsAddress, 103
pSecondaryV4DnsAddress, 103
pUserId, 103
pUserIdSize, 103
libsdp_SetReadBlockSize
lite-fw.h, 1097
libsdp_set_log_func
lite-fw.h, 1097
libsdplogger
lite-fw.h, 1097
lineValue
pack_voice_SLQSVoiceALSSelectLine_t, 441
linkage
altSrcInfo_t, 35
list_type
DMScustSettingList, 61
DMSgetCustomInput, 62
pack_dms_GetCustFeaturesV2_t, 303
listEntries
FMSPrefImageList, 68
listSize
FMSImageList, 67
FMSPrefImageList, 68
lite-fw.h
eFW_TYPE_CWE, 1098
eFW_TYPE_CWE_NVU, 1098
eFW_TYPE_INVALID, 1098
eFW_TYPE_MBN, 1098
eFW_TYPE_MBN_GOBI, 1098
eFW_TYPE_NVU, 1098
eFW_TYPE_SPK, 1098
eModel_9X15, 1099
eModel_9X30, 1099
eModel_9x06, 1099
eModel_9x07, 1099
eModel_Unknown, 1099
eModel_WP9X15, 1099
eQDL_HW_FLOW_DISABLE, 1099
eQDL_HW_FLOW_ENABLE, 1099
eQDL_HW_FLOW_INIT, 1099
eQDL_HW_FLOW_Unknown, 1099
eQDL_MODE_INIT, 1099
eQDL_MODE_TTYUSB, 1099
eQDL_MODE_UART0, 1099
eQDL_MODE_Unknown, 1099
eSDP_FWDWL_ERR_END, 1099
eSDP_FWDWL_ERR_FAIL, 1099
eSDP_FWDWL_ERR_FW_UPGRADE, 1099
eSDP_FWDWL_ERR_FW_VERSION_FAIL, 1099
eSDP_FWDWL_ERR_GENERAL, 1098
eSDP_FWDWL_ERR_INVALID_DEV, 1099
eSDP_FWDWL_ERR_INVALID_PATH, 1099
eSDP_FWDWL_ERR_PATH_NOT_SPECIFIED, 1099
eSDP_FWDWL_ERR_PATH_TOO_LONG, 1099
eSDP_FWDWL_ERR_PRI_FAIL, 1099
eSDP_FWDWL_ERR_SDK, 1099
eSDP_FWDWL_ERR_SDP_TIMEOUT, 1099
eSDP_FWDWL_ERR_SET_CBK, 1099
eSDP_FWDWL_ERR_TIMEOUT, 1099
eSDP_FWDWL_SUCCESS, 1098
lite-fw.h, 1094
IMG_MASK_CLEAR, 1097
IMG_MASK_GENERIC, 1097
IMG_MASK_MDM, 1097
IMG_MASK_PRI, 1097
libSDP_BuildImagesPreferenceRequest, 1097
libSDP_CalculateImageMask, 1097
libSDP_CheckValidFirmwareInfo, 1097
libSDP_DownloadFW, 1097
libSDP_ExtractFirmwareParametersByPath, 1097
libSDP_FirmwareInfo, 1097
libSDP_GetModelFamily, 1097
libSDP_GetVersion, 1097
libSDP_getFileType, 1097
libsdp_SetReadBlockSize, 1097
libsdp_set_log_func, 1097
libsdplogger, 1097
litefw_BuildImagesPreferenceRequest, 1100
litefw_CalculateImageMask, 1100
litefw_CheckValidFirmwareInfo, 1100
litefw_DownloadFW, 1100
litefw_ExtractFirmwareParametersByPath, 1101
litefw_FirmwareInfo, 1097
litefw_Fw_Type, 1098
litefw_GetModelFamily, 1102
litefw_GetQTLDownloadMode, 1102
litefw_GetQTLHWFlowControl, 1102
litefw_GetVersion, 1103
litefw_Models, 1099
litefw_QDL_FLOW_CONTROLs, 1099
litefw_QDL_MODEs, 1099
litefw_SetQTLDownloadMode, 1103
litefw_SetQTLHWFlowControl, 1103
litefw_SetReadBlockSize, 1104
litefw_fwdwl_error_codes, 1098
litefw_getFileType, 1101
litefw_logsenable, 1103
litefw_set_log_func, 1103
litefw_switch_9x07_to_downloadmode, 1104
litefw_switch_to_BootHoldMode, 1104
litefwlogger, 1098

liteQmiDataBearerMasks
wds.h, 1394
liteServiceClassInformation
voice.h, 1360
litefw_BuildImagesPreferenceRequest
lite-fw.h, 1100
litefw_CalculateImageMask
lite-fw.h, 1100
litefw_CheckValidFirmwareInfo
lite-fw.h, 1100
litefw_DownloadFW
lite-fw.h, 1100
litefw_ExtractFirmwareParametersByPath
lite-fw.h, 1101
litefw_FirmwareInfo
lite-fw.h, 1097
litefw_Fw_Type
lite-fw.h, 1098
litefw_GetModelFamily
lite-fw.h, 1102
litefw_GetQTLDownloadMode
lite-fw.h, 1102
litefw_GetQTLHWFlowControl
lite-fw.h, 1102
litefw_GetVersion
lite-fw.h, 1103
litefw_Models
lite-fw.h, 1099
litefw_QDL_FLOW_CONTROLs
lite-fw.h, 1099
litefw_QDL_MODEs
lite-fw.h, 1099
litefw_SetQTLDownloadMode
lite-fw.h, 1103
litefw_SetQTLHWFlowControl
lite-fw.h, 1103
litefw_SetReadBlockSize
lite-fw.h, 1104
litefw_fwdwl_error_codes
lite-fw.h, 1098
litefw_getFileType
lite-fw.h, 1101
litefw_logsenable
lite-fw.h, 1103
litefw_set_log_func
lite-fw.h, 1103
litefw_switch_9x07_to_downloadmode
lite-fw.h, 1104
litefw_switch_to_BootHoldMode
lite-fw.h, 1104
litefwlogger
lite-fw.h, 1098
liteqmi_GetVersion
common.h, 1015
liteqmi_helper_decode7bitAsciiEncString
common.h, 1016
liteqmi_log
common.h, 1016
loc.h
eQMI_LOC_SESS_STATUS_FAILURE, 1111
eQMI_LOC_SESS_STATUS_IN_PROGRESS, 1111
eQMI_LOC_SESS_STATUS_SUCCESS, 1111
eQMI_LOC_SESS_STATUS_TIMEOUT, 1111
loc.h, 1105
LOCEVENTMASKNMEA, 1110
LOCEVENTMASKWIFIREQ, 1111
MAX_TEMP_DATA_LEN, 1111
pack_loc_DeleteAssistData, 1111
pack_loc_EventRegister, 1112
pack_loc_SLQSLOCGetBestAvailPos, 1113
pack_loc_SLQSLOCGetServer, 1113
pack_loc_SLQSLOCInjectPosition, 1114
pack_loc_SLQSLOCInjectSensorData, 1114
pack_loc_SLQSLOCInjectUTCTime, 1114
pack_loc_SLQSLOCSetCradleMountConfig, 1115
pack_loc_SLQSLOCSetServer, 1115
pack_loc_SetExtPowerState, 1112
pack_loc_SetOperationMode, 1112
pack_loc_Start, 1116
pack_loc_Stop, 1116
unpack_loc_BestAvailPos_Ind, 1116
unpack_loc_CradleMountCallback_Ind, 1117
unpack_loc_DeleteAssistData, 1117
unpack_loc_DeleteAssistData_Ind, 1118
unpack_loc_EngineState_Ind, 1118
unpack_loc_EventNMEA_Ind, 1118
unpack_loc_EventRegister, 1119
unpack_loc_EventTimeSyncCallback_Ind, 1119
unpack_loc_GetServer_Ind, 1119
unpack_loc_GnssSvInfo_Ind, 1120
unpack_loc_InjectPositionCallback_Ind, 1120
unpack_loc_InjectSensorDataCallback_Ind, 1120
unpack_loc_InjectTimeSyncDataCallback_Ind, 1121
unpack_loc_InjectUTCTimeCallback_Ind, 1121
unpack_loc_PositionRpt_Ind, 1121
unpack_loc_SLQSLOCGetBestAvailPos, 1124
unpack_loc_SLQSLOCGetServer, 1124
unpack_loc_SLQSLOCGetServer_t, 1111
unpack_loc_SLQSLOCInjectPosition, 1124
unpack_loc_SLQSLOCInjectPosition_t, 1111
unpack_loc_SLQSLOCInjectSensorData, 1125
unpack_loc_SLQSLOCInjectSensorData_t, 1111
unpack_loc_SLQSLOCInjectUTCTime, 1125
unpack_loc_SLQSLOCInjectUTCTime_t, 1111
unpack_loc_SLQSLOCSetCradleMountConfig, 1125
unpack_loc_SLQSLOCSetCradleMountConfig_t, 1111
unpack_loc_SLQSLOCSetServer, 1126
unpack_loc_SensorStreamingCallback_Ind, 1122
unpack_loc_SetExtPowerConfig_Ind, 1122
unpack_loc_SetExtPowerState, 1122
unpack_loc_SetOperationMode, 1123
unpack_loc_SetOperationMode_Ind, 1123

unpack_loc_SetServer_Ind, 1123
unpack_loc_Start, 1126
unpack_loc_Stop, 1126
loc_BdsSV, 122
id, 122
mask, 122
loc_BdsSVInfo, 122
len, 123
pSV, 123
loc_CellDb, 123
mask, 123
loc_ClkInfo, 123
mask, 124
loc_GnssData, 125
mask, 126
loc_IPv4Config, 129
IPv4Addr, 129
IPv4Port, 129
loc_IPv4Info, 129
address, 130
port, 130
TlvPresent, 130
loc_IPv6Config, 130
IPv6Addr, 130
IPv6Port, 130
loc_IPv6Info, 130
address, 131
port, 131
TlvPresent, 131
loc_LocApplicationInfo, 131
appNameLength, 132
appProviderLength, 132
appVersionLength, 132
appVersionValid, 132
pAppName, 132
pAppProvider, 132
pAppVersion, 132
loc_SV, 136
id, 137
mask, 137
system, 137
loc_SVInfo, 137
len, 137
pSV, 137
loc_URLAddrInfo, 138
urlAddr, 139
loc_accelAcceptReady, 120
batchPerSec, 121
injectEnable, 121
samplesPerBatch, 121
loc_accelTempAcceptReady, 121
batchPerSec, 122
injectEnable, 122
samplesPerBatch, 122
loc_gpsTime, 126
gpsTimeOfWeekMs, 127
gpsWeek, 127
loc_gyroAcceptReady, 127
batchPerSec, 128
injectEnable, 128
samplesPerBatch, 128
loc_gyroTempAcceptReady, 128
batchPerSec, 128
injectEnable, 128
samplesPerBatch, 128
loc_precisionDilution, 132
HDOP, 133
PDOP, 133
VDOP, 133
loc_satelliteInfo, 133
azimuth, 135
elevation, 135
gnssSvId, 135
healthStatus, 135
snr, 135
svInfoMask, 135
svListLen, 135
svStatus, 135
system, 135
validMask, 135
loc_sensorDataUsage, 135
aidingIndicatorMask, 136
usageMask, 136
loc_svUsedforFix, 137
gnssSvUsedList, 138
gnssSvUsedList_len, 138
loc_urlAddr, 138
address, 138
TlvPresent, 138
localTimeOffset
nas_qaQmi3Gpp2TimeZone, 233
logString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
logger
common.h, 1013
longName
nas_PLMNNetworkNameData, 230
unpack_nas_SLQSGetPLMNName_t, 697
longNameCI
unpack_nas_SLQSGetPLMNName_t, 697
longNameEn
unpack_nas_SLQSGetPLMNName_t, 697
longNameLen
nas_PLMNNetworkNameData, 230
unpack_nas_SLQSGetPLMNName_t, 697
longNameSB
unpack_nas_SLQSGetPLMNName_t, 697
longNameSpareBits
nas_PLMNNetworkNameData, 230
Longitude
unpack_pds_SLQSGetGPSStateInfo_t, 755
longitude
pack_loc_SLQSLOCInjectPosition_t, 340
loopbackMode
pack_wds_SLQSSSetLoopback_t, 479
loopbackMultiplier

pack_wds_SLQSSSetLoopback_t, 479
LpmFlag
unpack_dms_SLQSDmsSwiGetPCInfo_t, 590
LteBandCapability
unpack_dms_SLQSGetBandCapability_t, 597
lteEmmDl
pack_nas_SLQSNasSwiIndicationRegister_t, 363
lteEmmUl
pack_nas_SLQSNasSwiIndicationRegister_t, 363
lteEsmDl
pack_nas_SLQSNasSwiIndicationRegister_t, 363
lteEsmUl
pack_nas_SLQSNasSwiIndicationRegister_t, 363
LteGsmCellInfo
nas_LTEInfoNeighboringGSM, 195
lteM1BandPref
nas_LteM1BandPrefTlv, 197
lteNb1BandPref
nas_LteNb1BandPrefTlv, 199
LteQci
unpack_qos_swiQosFlow_t, 776
lteRsrpDelta
nas_SLQSSignalStrengthsIndReq, 257
lteRsrpinfo
nas_SLQSSignalStrengthsInformation, 258
lteSSInfo, 139
rsrp, 139
rsrq, 140
rssi, 140
snr, 140
lteSnrDelta
nas_SLQSSignalStrengthsIndReq, 257
lteSnrinfo
nas_SLQSSignalStrengthsInformation, 258
ltersrp
unpack_nas_SLQSGetSignalStrength_t, 705
ltesnr
unpack_nas_SLQSGetSignalStrength_t, 705
lvl
unpack_tmd_SLQSTmdMitigationLvlRptCallback_-ind_t, 826

m_FwBuildId
CarrierImage_t, 44
m_FwImageId
CarrierImage_t, 44
m_PriBuildId
CarrierImage_t, 44
m_PriImageId
CarrierImage_t, 45
m_nCarrierId
CarrierImage_t, 44
m_nFolderId
CarrierImage_t, 44
m_nStorage
CarrierImage_t, 44
MACIndex
nas_NetworkStatEVDO, 217
MAX_BUILD_ID_LEN
dms.h, 1024
MAX_ERROR_CODE_LEN
imsa.h, 1090
MAX_ICCID_LENGTH
uim.h, 1339
MAX_MSE_TWS_MSG
sms.h, 1268
MAX_NAME_LEN
ims.h, 1079
MAX_NO_OF_SLOTS
uim.h, 1339
MAX_SENSOR_DATA_LEN
loc.h, 1111
MAX_SLOTS_STATUS
uim.h, 1339
MAX_SMS_LIST_SIZE
sms.h, 1268
MAX_TEMP_DATA_LEN
loc.h, 1111
MAXVOICEUSSDLENGTH
voice.h, 1360
MCC
nas_CDMASysInfo, 152
nas_CDMASysInfoExt, 153
nas_currentPLMN, 160
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_QmiNas3GppNetworkInfo, 234
nas_QmiNas3GppNetworkRAT, 235
nas_QmisNasPcsDigit, 236
nas_WCDMASysInfo, 279
unpack_nas_GetServingNetwork_t, 682
MDMCallDuration
connectionStatus, 49
MDMConnStatus
connectionStatus, 49
wds_ConnStatusTlv, 978
MEID_MAX_SIZE
dms.h, 1024
MEIDString
unpack_dms_GetDeviceSerialNumbers_t, 565
MIN
unpack_dms_GetVoiceNumber_t, 580
MINREQBKLEN
common.h, 1012
MMTlv
unpack_sms_SetNewSMSCallback_ind_t, 785
MNC
nas_CDMASysInfo, 152
nas_currentPLMN, 160
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_QmiNas3GppNetworkInfo, 234
nas_QmiNas3GppNetworkRAT, 235
nas_QmisNasPcsDigit, 236
nas_WCDMASysInfo, 279
unpack_nas_GetServingNetwork_t, 682
MPTlv

NASQmiCbkNasSystemSelPrefInd, 291
MSETlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
MSGID_AND_LEN
common.h, 1012
MSGID_DONT_CARE
common.h, 1012
MTMessageInfo
newMTMessageTlv, 294
MTUSize
pack_swidms_SLQSSwiDmsSetMTU_t, 418
MTUSize3gpp
swidms_mtuSize3gppTlv, 518
manString
pack_dms_SLQSSwiSetHostDevInfo_t, 313
unpack_dms_SLQSSwiGetHostDevInfo_t, 606
manufacturer
unpack_dms_GetManufacturer_t, 571
Mask
pack_wds_SLQSGetDUNCallInfo_t, 472
mask
loc_BdsSV, 122
loc_CellDb, 123
loc_ClkInfo, 124
loc_GnssData, 126
loc_SV, 137
unpack_qos_IPv6TrafCls_t, 758
unpack_qos_Tos_t, 777
max
PackSwiAVMSSettingsPeriodsInfo, 489
max_channel_rx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 899
max_channel_tx_rate
unpack_wds_SLQSGetCurrentChannelRate_t, 899
max_dist
pack_swiloc_SwiLocSetAutoStart_t, 420
unpack_swiloc_SwiLocGetAutoStart_t, 818
max_dist_reported
unpack_swiloc_SwiLocGetAutoStart_t, 818
max_time
pack_swiloc_SwiLocSetAutoStart_t, 420
unpack_swiloc_SwiLocGetAutoStart_t, 818
max_time_reported
unpack_swiloc_SwiLocGetAutoStart_t, 818
MaxAllowedPktSz
unpack_qos_swiQosFlow_t, 776
MaxChanRxRate
dunchannelRate, 63
MaxChanTxRate
dunchannelRate, 63
maxChannelRXRate
unpack_wds_GetConnectionRate_t, 881
maxChannelTXRate
unpack_wds_GetConnectionRate_t, 881
maxDlBitRate
LibPackQosClassID, 115
maxDownlinkBitrate
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
maxImages
FMSImageIDEntries, 67
maxMitigationLevel
tmd_mitigationDevList, 521
MaxRXChannelRate
unpack_dms_GetDeviceCap_t, 560
maxRxChannelRate
unpack_dms_GetDeviceCapabilities_t, 562
maxSDUSize
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
maxStorageSize
sms_maxStorageSizeResp, 502
MaxTXChannelRate
unpack_dms_GetDeviceCap_t, 560
maxTxChannelRate
unpack_dms_GetDeviceCapabilities_t, 562
maxUlBitRate
LibPackQosClassID, 115
maxUplinkBitrate
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
mcc
nas_CSGID, 159
nas_MNRInfo, 211
nas_netSelectionPref, 213
nas_OperatorPLMNData, 221
pack_nas_SLQSGetPLMNName_t, 352
unpack_nas_GetHomeNetwork_t, 678
mccM
nas_minBasedIMSI, 210
mccT
nas_trueIMSI, 268
mdmCallDurationActive
unpack_wds_SLQSGetDUNCallInfo_t, 903
meanThroughputClass
LibPackGPRSRequestedQoS, 91
wds_GPRSQoS, 987
meid
unpack_dms_GetSerialNumbers_t, 577
meidSize
unpack_dms_GetDeviceSerialNumbers_t, 565
meidString
unpack_dms_SLQSSwiGetSerialNoExt_t, 607
message
unpack_sms_SLQSGetSMS_t, 788
message_type
NASOTAMessageTlv, 285
messageClass
sms_routeEntry, 506
messageFailureCode
unpack_sms_SendSMS_t, 783
messageFormat
pack_sms_SaveSMS_t, 398
pack_sms_SendSMS_t, 398

sms_sendAsyncsmsParams, 508
unpack_sms_SLQSGetSMS_t, 788
messageID
unpack_sms_SendSMS_t, 783
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 799
messageIndex
pack_sms_SLQSGetSMS_t, 401
pack_sms_SLQSModifySMSStatus_t, 403
qmiSmsMessageList, 489
sMSMTMessageInfo, 514
messageList
unpack_sms_SLQSGetSMSList_t, 790
messageListSize
unpack_sms_SLQSGetSMSList_t, 790
messageMode
sMSMessageModeInfo, 513
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 800
MessageModeInfo
messageModeTlv, 140
messageModeTlv, 140
MessageModeInfo, 140
TlvPresent, 140
messageSize
pack_sms_SaveSMS_t, 398
pack_sms_SendSMS_t, 398
sms_sendAsyncsmsParams, 508
unpack_sms_SLQSGetSMS_t, 788
messageTag
pack_sms_SLQSModifySMSStatus_t, 403
qmiSmsMessageList, 489
unpack_sms_SLQSGetSMS_t, 788
messageType
sms_routeEntry, 506
MicMute
pack_audio_SLQSSetAudioProfile_t, 299
pack_swiaudio_SLQSSetM2MAVMute_t, 412
unpack_audio_SLQSGetAudioProfile_t, 550
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 801
unpack_swiaudio_SLQSGetM2MAVMute_t, 803
min
PackSwiAVMSSettingsPeriodsInfo, 489
MinPolicedPktSz
unpack_qos_swiQosFlow_t, 776
minSessExp
ims_MinSessExpInfo, 75
minSize
unpack_dms_GetVoiceNumber_t, 580
minute
nas_timeInfo, 267
nas_UniversalTime, 273
mipMode
unpack_wds_GetMobileIP_t, 886
mipStatus
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
mipstatAvail
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
mitigationDevID
pack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 426
pack_tmd_SLQSTmdGetMitigationLvl_t, 427
pack_tmd_SLQSTmdRegNotMitigationLvl_t, 427
mitigationDevIDLen
pack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 426
pack_tmd_SLQSTmdGetMitigationLvl_t, 427
pack_tmd_SLQSTmdRegNotMitigationLvl_t, 428
mitigationDevId
tmd_mitigationDevList, 521
mitigationDevIdLen
tmd_mitigationDevList, 521
MitigationDevList
unpack_tmd_SLQSTmdGetMitigationDevList_t, 824
MitigationDevListLen
unpack_tmd_SLQSTmdGetMitigationDevList_t, 824
mnc
nas_CSGID, 159
nas_MNRInfo, 211
nas_netSelectionPref, 213
nas_OperatorPLMNData, 221
pack_nas_SLQSGetPLMNName_t, 352
unpack_nas_GetHomeNetwork_t, 678
mncPcsDigits
nas_CSGID, 159
mobileCountryCode
sMSEtwsPlmnInfo, 513
mobileIP
pack_wds_SLQSSetWdsEventCallback_t, 478
mobileNetworkCode
sMSEtwsPlmnInfo, 513
mode
pack_dms_SetEventReport_t, 308
pack_dms_SetPower_t, 310
pack_loc_SetOperationMode_t, 334
pack_sms_SLQSGetSmsBroadcastConfig_t, 401
pack_sms_SLQSSetSmsBroadcastActivation_t, 404
pack_sms_SLQSSetSmsBroadcastConfig_t, 405
pack_wds_SetMobileIP_t, 466
uim_refreshevent, 539
voice_callInfo, 948
ModePref
nas_ModePrefTlv, 211
NASModePreferenceTlv, 283
modelString
pack_dms_SLQSSwiSetHostDevInfo_t, 313
unpack_dms_SLQSSwiGetHostDevInfo_t, 606
modelid
unpack_dms_GetModelID_t, 572
modelid_str

unpack_dms_GetFirmwareInfo_t, 567
modemMode
nas_CommInfo, 158
modemindex
pack_fms_SetImagesPreference_t, 321
month
nas_timeInfo, 267
nas_UniversalTime, 273
msgCount
sms_messageWaitingInfoContent, 503
msgDelFailureCause
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 799
msgDelFailureType
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 799
msgProtocol
sms_msgProtocolResp, 503
msgType
sms_messageWaitingInfoContent, 503
msgWaitInfo
sms_getMsgWaitingInfo, 500
unpack_sms_SLQSWmsMessageWaitingCall-Back_ind_t, 800
msgid
pack_qmi_t, 394
unpack_qmi_t, 756
msgtype
common.h, 1014
Mtu
unpack_wds_SLQSGetRuntimeSettings_t, 906
MultDisc
nas_protocolSubtypeElement, 232
multiplier
unpack_qos_pktErrRate_t, 759
MuxID
qos_BindDataPortMuxID_t, 490

NAI
unpack_wds_GetMobileIPProfile_t, 888
NAS_NAM_NAME_LENGTH
nas.h, 1136
NAS_PLMN_LENGTH
nas.h, 1136
NASBandPreferenceTlv, 280
band_pref, 280
TlvPresent, 280
NASEmergencyModeTlv, 280
EmerMode, 280
TlvPresent, 280
NASGWAcqOrderPrefTlv, 281
GWAcqOrderPref, 282
TlvPresent, 282
NASLTEBandPreferenceTlv, 282
LTEBandPref, 282
TlvPresent, 282
NASLteNasReleaseInfoTlv, 282
nas_major, 283
nas_minor, 283
nas_release, 283
TlvPresent, 283
NASModePreferenceTlv, 283
ModePref, 283
TlvPresent, 283
NASNetSelPreferenceTlv, 283
NetSelPref, 284
TlvPresent, 284
NASOTAMessageTlv, 284
data_buf, 284
data_len, 284
message_type, 285
TlvPresent, 285
NASPRLPreferenceTlv, 288
PRLPref, 289
TlvPresent, 289
NASPhyCaAggPcellInfo, 285
dl_bw_value, 285
freq, 285
iLTEbandValue, 285
pci, 285
TlvPresent, 285
NASPhyCaAggScellDlBw, 286
dl_bw_value, 286
TlvPresent, 286
NASPhyCaAggScellIndType, 287
freq, 287
pci, 287
scell_state, 287
TlvPresent, 287
NASPhyCaAggScellIndex, 286
scell_idx, 286
TlvPresent, 286
NASPhyCaAggScellInfo, 287
dl_bw_value, 288
freq, 288
iLTEbandValue, 288
pci, 288
scell_state, 288
TlvPresent, 288
NASQmiCbkNasSwiOTAMessageInd, 289
nasRelInfoTlv, 289
otaMsgTlv, 289
timeTlv, 289
NASQmiCbkNasSystemSelPrefInd, 289
BPTlv, 290
EMTlv, 291
GWAOPTlv, 291
LBPTlv, 291
MPTlv, 291
NSPTlv, 291
PRLPTlv, 291
RPTlv, 291
SDPTlv, 291
NASRoamPreferenceTlv, 291
RoamPref, 291
TlvPresent, 291
NASServDomainPrefTlv, 291

SrvDomainPref, 292
TlvPresent, 292
NASServingSystemInfo, 292
csAttachState, 293
hdrPersonality, 293
psAttachState, 293
radioInterfaceList, 293
radioInterfaceNo, 293
registrationState, 293
selectedNetwork, 293
NASTimeInfoTlv, 293
time, 294
TlvPresent, 294
NMEAData
unpack_loc_EventNMEA_Ind_t, 651
NSPTlv
NASQmiCbkNasSystemSelPrefInd, 291
nUsbNetNum
pack_swidms_SLQSSwiDmsSetUsbNetNum_t, 419
NWQoSStatus
unpack_qos_SLQSQosGetNetworkStatus_t, 762
NWRegStat
unpack_sms_SLQSNWRegInfoCallback_ind_t, 792
naiSize
unpack_wds_GetMobileIPProfile_t, 888
namID
pack_nas_SLQSNasGet3GPP2Subscription_t, 358
voice_airTimer, 929
voice_prefVoiceSO, 969
voice_roamTimer, 973
namName
nas_namName, 212
namNameLen
nas_namName, 212
Name
unpack_nas_GetServingNetwork_t, 682
name
unpack_nas_GetHomeNetwork_t, 678
unpack_wds_GetDefaultProfile_t, 884
nameLen
voice_remotePartyName, 971
namePI
voice_remotePartyName, 971
nameSize
unpack_nas_GetHomeNetwork3GPP2_t, 676
unpack_nas_GetServingNetwork_t, 683
nameString
pack_dms_SLQSSwiSetOSInfo_t, 314
unpack_dms_SLQSSwiGetOSInfo_t, 607
namelength
unpack_omaDmFotaTlv_t, 741
namesize
unpack_wds_GetDefaultProfile_t, 884
nas.h
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6, 1137
eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75, 1137
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED, 1138
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED, 1137
eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED, 1137
eNAS_LTE_CPHY_CA_BW_NRB_LITE_100, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_15, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_25, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_50, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_6, 1138
eNAS_LTE_CPHY_CA_BW_NRB_LITE_75, 1138
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED_LITE, 1138
eNAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED_LITE, 1138
eNAS_LTE_CPHY_SCELL_STATE_DECONFIGURED_LITE, 1138
nas.h, 1127
NAS_PLMN_LENGTH, 1136
pack_nas_GetACCOLC, 1138
pack_nas_GetANAAAAuthenticationStatus, 1138
pack_nas_GetCDMANetworkParameters, 1139
pack_nas_GetHomeNetwork, 1139
pack_nas_GetHomeNetwork3GPP2, 1139
pack_nas_GetNetworkPreference, 1140
pack_nas_GetRFInfo, 1140
pack_nas_GetServingNetwork, 1140
pack_nas_GetServingNetworkCapabilities, 1140
pack_nas_GetSignalStrengths, 1141
pack_nas_InitiateDomainAttach, 1141
pack_nas_PerformNetworkScan, 1142
pack_nas_SLQSConfigSigInfo, 1144
pack_nas_SLQSGetErrorRate, 1144
pack_nas_SLQSGetNetworkTime, 1145
pack_nas_SLQSGetOperatorNameData, 1145
pack_nas_SLQSGetPLMNName, 1145
pack_nas_SLQSGetServingSystem, 1145
pack_nas_SLQSGetSignalStrength, 1146
pack_nas_SLQSGetSysInfo, 1146
pack_nas_SLQSGetSysSelectionPref, 1147
pack_nas_SLQSGetSysSelectionPrefExt, 1147
pack_nas_SLQSInitiateNetworkRegistration, 1147
pack_nas_SLQSNASGeteDRXParams, 1149
pack_nas_SLQSNASSeteDRXParams, 1151

pack_nas_SLQSNASSwiGetChannelLock, 1151
pack_nas_SLQSNASSwiSetChannelLock, 1152
pack_nas_SLQSNasConfigSigInfo2, 1148
pack_nas_SLQSNasGet3GPP2Subscription, 1148
pack_nas_SLQSNasGetCellLocationInfo, 1148
pack_nas_SLQSNasGetHDRColorCode, 1149
pack_nas_SLQSNasGetRFInfo, 1149
pack_nas_SLQSNasGetSigInfo, 1150
pack_nas_SLQSNasGetTxRxInfo, 1150
pack_nas_SLQSNasIndicationRegisterExt, 1150
pack_nas_SLQSNasSwiIndicationRegister, 1152
pack_nas_SLQSNasSwiModemStatus, 1152
pack_nas_SLQSSetBandPreference, 1153
pack_nas_SLQSSetSignalStrengthsCallback, 1153
pack_nas_SLQSSetSysSelectionPref, 1153
pack_nas_SLQSSetSysSelectionPrefExt, 1154
pack_nas_SLQSSwiGetHDRPersonality, 1154
pack_nas_SLQSSwiGetHDRProtSubtype, 1155
pack_nas_SLQSSwiGetHRPDStats, 1155
pack_nas_SLQSSwiGetLteCQI, 1155
pack_nas_SLQSSwiGetLteSccRxInfo, 1156
pack_nas_SLQSSwiNetworkDebug, 1156
pack_nas_SLQSSwiPSDetach, 1156
pack_nas_SetACCOLC, 1142
pack_nas_SetCDMANetworkParameters, 1142
pack_nas_SetLURejectCallback, 1143
pack_nas_SetNetworkPreference, 1143
pack_nas_SetRFInfoCallback, 1143
pack_nas_SlqsGetLTECphyCAInfo, 1144
unpack_nas_GetACCOLC, 1157
unpack_nas_GetANAAAAuthenticationStatus, 1157
unpack_nas_GetCDMANetworkParameters, 1157
unpack_nas_GetHomeNetwork, 1158
unpack_nas_GetHomeNetwork3GPP2, 1158
unpack_nas_GetNetworkPreference, 1158
unpack_nas_GetRFInfo, 1159
unpack_nas_GetServingNetwork, 1159
unpack_nas_GetServingNetworkCapabilities, 1159
unpack_nas_GetSignalStrengths, 1160
unpack_nas_InitiateDomainAttach, 1160
unpack_nas_InitiateDomainAttach_t, 1137
unpack_nas_PerformNetworkScan, 1160
unpack_nas_SLQSConfigSigInfo, 1164
unpack_nas_SLQSConfigSigInfo_t, 1137
unpack_nas_SLQSGetErrorRate, 1165
unpack_nas_SLQSGetNetworkTime, 1165
unpack_nas_SLQSGetOperatorNameData, 1166
unpack_nas_SLQSGetPLMNName, 1166
unpack_nas_SLQSGetServingSystem, 1166
unpack_nas_SLQSGetSignalStrength, 1167
unpack_nas_SLQSGetSysInfo, 1167
unpack_nas_SLQSGetSysSelectionPref, 1167
unpack_nas_SLQSGetSysSelectionPrefExt, 1168
unpack_nas_SLQSInitiateNetworkRegistration, 1168
unpack_nas_SLQSInitiateNetworkRegistration_t, 1137
unpack_nas_SLQSNASGeteDRXParams, 1170
unpack_nas_SLQSNASSeteDRXParams, 1172
unpack_nas_SLQSNASSeteDRXParams_t, 1137
unpack_nas_SLQSNASSwiGetChannelLock, 1173
unpack_nas_SLQSNASSwiSetChannelLock, 1174
unpack_nas_SLQSNASSwiSetChannelLock_t, 1137
unpack_nas_SLQSNasConfigSigInfo2, 1169
unpack_nas_SLQSNasConfigSigInfo2_t, 1137
unpack_nas_SLQSNasGet3GPP2Subscription, 1169
unpack_nas_SLQSNasGetCellLocationInfo, 1169
unpack_nas_SLQSNasGetHDRColorCode, 1170
unpack_nas_SLQSNasGetRFInfo, 1170
unpack_nas_SLQSNasGetSigInfo, 1171
unpack_nas_SLQSNasGetTxRxInfo, 1171
unpack_nas_SLQSNasIndicationRegisterExt, 1171
unpack_nas_SLQSNasIndicationRegisterExt_t, 1137
unpack_nas_SLQSNasNetworkTimeCallBack_ind, 1172
unpack_nas_SLQSNasSigInfoCallback_ind, 1173
unpack_nas_SLQSNasSwiIndicationRegister, 1173
unpack_nas_SLQSNasSwiIndicationRegister_t, 1137
unpack_nas_SLQSNasSwiModemStatus, 1174
unpack_nas_SLQSNasSwiOTAMessageCallback_ind, 1174
unpack_nas_SLQSNasSysInfoCallback_ind, 1175
unpack_nas_SLQSNasTimerCallback_ind, 1175
unpack_nas_SLQSSetBandPreference, 1175
unpack_nas_SLQSSetBandPreference_t, 1137
unpack_nas_SLQSSetSignalStrengthsCallback, 1176
unpack_nas_SLQSSetSignalStrengthsCallback_t, 1137
unpack_nas_SLQSSetSysSelectionPref, 1176
unpack_nas_SLQSSetSysSelectionPref_t, 1137
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind, 1176
unpack_nas_SLQSSetSysSelectionPrefExt, 1177
unpack_nas_SLQSSetSysSelectionPrefExt_t, 1137
unpack_nas_SLQSSwiGetHDRPersonality, 1177
unpack_nas_SLQSSwiGetHDRProtSubtype, 1177
unpack_nas_SLQSSwiGetHRPDStats, 1178
unpack_nas_SLQSSwiGetLteCQI, 1178
unpack_nas_SLQSSwiGetLteSccRxInfo, 1178
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind, 1179
unpack_nas_SLQSSwiNetworkDebug, 1179
unpack_nas_SLQSSwiPSDetach, 1179
unpack_nas_SLQSSwiPSDetach_t, 1137

unpack_nas_SLQSSwiRandIndicatorCallback_Ind, 1180
unpack_nas_SetACCOLC, 1161
unpack_nas_SetACCOLC_t, 1137
unpack_nas_SetCDMANetworkParameters, 1161
unpack_nas_SetCDMANetworkParameters_t, 1137
unpack_nas_SetDataCapabilitiesCallback_ind, 1161
unpack_nas_SetEventReportInd, 1162
unpack_nas_SetLURejectCallback, 1162
unpack_nas_SetLURejectCallback_t, 1137
unpack_nas_SetNasLTECphyCaIndCallback_ind, 1163
unpack_nas_SetNetworkPreference, 1163
unpack_nas_SetRFInfoCallback, 1163
unpack_nas_SetRFInfoCallback_t, 1137
unpack_nas_SetRoamingIndicatorCallback_ind, 1164
unpack_nas_SetServingSystemCallback_ind, 1164
unpack_nas_SlqsGetLTECphyCAInfo, 1165
unpack_valid_nas_GetCDMANetworkParameters, 1180
unpack_valid_nas_SLQSGetServingSystem, 1180
unpack_valid_nas_SLQSGetSignalStrength, 1181
unpack_valid_nas_SLQSNasGetSigInfo, 1182
nas_AcqOrderPrefTlv, 141
acqOrdeLen, 141
pAcqOrder, 141
TlvPresent, 142
nas_ActPilotPNElement, 142
ActSetPilotPN, 142
ActSetPilotPNStrength, 142
nas_AddCDMASysInfo, 142
geoSysIdx, 143
regPrd, 143
nas_AddSysInfo, 143
cellBroadcastCap, 143
geoSysIdx, 143
nas_BandPrefTlv, 143
BandPref, 145
TlvPresent, 145
nas_CDMAChannel, 146
priChA, 147
priChB, 147
secChA, 147
secChB, 147
nas_CDMAECIOThresh, 147
CDMAECIOThreshListLen, 147
pCDMAECIOThreshList, 147
nas_CDMAInfo, 148
baseId, 148
baseLat, 148
baseLong, 148
nid, 148
refpn, 149
sid, 149
nas_CDMARSSIThresh, 149
CDMARSSIThreshListLen, 149
pCDMARSSIThreshList, 149
nas_CDMASysInfo, 149
baseId, 152
baseLat, 152
baseLong, 152
bsInfoValid, 152
bsPRev, 152
bsPRevValid, 152
ccsSupported, 152
ccsSupportedValid, 152
cdmaSysIdValid, 152
isSysPrlMatch, 152
isSysPrlMatchValid, 152
MCC, 152
MNC, 152
networkID, 152
networkIdValid, 152
pRevInUse, 153
pRevInUseValid, 153
packetZone, 153
packetZoneValid, 153
sysInfoCDMA, 153
systemID, 153
nas_CDMASysInfoExt, 153
imsi_11_12, 153
MCC, 153
nas_CSGID, 158
id, 159
mcc, 159
mnc, 159
mncPcsDigits, 159
rat, 159
nas_CallBarringSysInfo, 145
csBarStatus, 145
psBarStatus, 145
nas_CiotAcqOrderPrefTlv, 155
ciotAcqOrderLen, 156
pCiotAcqOrder, 156
TlvPresent, 156
nas_CiotLteOpModePrefTlv, 156
ciotLteOpModePref, 156
TlvPresent, 156
nas_CommInfo, 156
imsRegState, 158
modemMode, 158
psState, 158
systemMode, 158
temperature, 158
nas_DRCParams, 165
DRCCover, 166
DRCValue, 166
nas_DataStatusDetail, 161
IPAddress, 162
LastErrCode, 162
nas_DeviceConfigDetail, 164
Chipset, 165

HWVersion, 165
QLIC, 165
Technology, 165
nas_ECIOThresh, 167
ECIOThresListLen, 167
pECIOThresList, 167
nas_EmerModeTlv, 167
EmerMode, 168
TlvPresent, 168
nas_GERANInfo, 169
arfcn, 170
bsic, 170
cellID, 170
insNmrCellInfo, 170
lac, 170
nmrInst, 170
plmn, 170
rxLev, 170
timingAdvance, 170
nas_GSMRSSIThresh, 172
GSMRSSIThreshListLen, 173
pGSMRSSIThreshList, 173
nas_GSMSrvStatusInfo, 173
isPrefDataPath, 174
srvStatus, 174
trueSrvStatus, 174
nas_GSMSysInfo, 174
cellId, 176
cellIdValid, 176
dtmSupp, 176
dtmSuppValid, 176
egprsSupp, 176
egprsSuppValid, 176
lac, 176
lacValid, 176
MCC, 176
MNC, 176
networkIdValid, 176
regRejectInfoValid, 176
rejCause, 176
rejectSrvDomain, 177
sysInfoGSM, 177
nas_GWAcqOrderPrefTlv, 177
GWAcqOrderPref, 177
TlvPresent, 177
nas_HDRECIOThresh, 177
HDRECIOThreshListLen, 178
pHDRECIOThreshList, 178
nas_HDRIOThresh, 178
HDRIOThreshListLen, 178
pHDRIOThreshList, 178
nas_HDRPersonality_Ind_Data, 178
pCurrentPersonality, 179
pPersonalityListLength, 179
pProtocolSubtypeElement, 179
nas_HDRRSSIThresh, 179
HDRRSSIThreshListLen, 179
pHDRRSSIThreshList, 179
nas_HDRSINRThresh, 180
HDRSINRThresListLen, 180
pHDRSINRThresList, 180
nas_HDRSINRThreshold, 180
HDRSINRThreshListLen, 181
pHDRSINRThreshList, 181
nas_HDRSysInfo, 181
hdrActiveProt, 183
hdrActiveProtValid, 183
hdrPersonality, 183
hdrPersonalityValid, 183
is856SysId, 183
is856SysIdValid, 183
isSysPrlMatch, 183
isSysPrlMatchValid, 183
sysInfoHDR, 183
nas_IOThresh, 185
IOThresListLen, 185
pIOThresList, 185
nas_LTEBandPrefTlv, 186
LTEBandPref, 187
TlvPresent, 188
nas_LTEInfo, 189
band, 191
bandwidth, 191
emmConnState, 191
emmState, 191
emmSubState, 191
RXChan, 191
TXChan, 191
nas_LTEInfoInterfreq, 192
freqsLen, 192
InfoInterfreq, 192
ueInIdle, 192
nas_LTEInfoIntrafreq, 192
CellParams, 194
cellReselPriority, 194
cellsLen, 194
earfcn, 194
globalCellId, 194
plmn, 194
sIntraSearch, 194
sNonIntraSearch, 194
servingCellId, 194
tac, 194
threshServingLow, 194
ueInIdle, 194
nas_LTEInfoNeighboringGSM, 194
freqsLen, 195
LteGsmCellInfo, 195
ueInIdle, 195
nas_LTEInfoNeighboringWCDMA, 195
freqsLen, 196
ueInIdle, 196
nas_LTEOperationMode, 200
pLTEOperationMode, 200
TlvPresent, 200
nas_LTERSRPThresh, 201

LTERSRPThreshListLen, 201
pLTERSRPThreshList, 202
nas_LTERSRQThresh, 202
LTERSRQThreshListLen, 202
pLTERSRQThreshList, 202
nas_LTERSSIThresh, 202
LTERSSIThreshListLen, 203
pLTERSSIThreshList, 203
nas_LTESNRThresh, 205
LTESNRThresListLen, 205
pLTESNRThresList, 205
nas_LTESNRThreshold, 205
LTESNRThreshListLen, 206
pLTESNRThreshList, 206
nas_LTESigRptCfg, 203
avgPeriod, 203
rptRate, 203
nas_LTESigRptConfig, 204
avgPeriod, 204
rptRate, 204
nas_LTESysInfo, 206
cellId, 208
cellIdValid, 208
lac, 208
lacValid, 208
MCC, 208
MNC, 208
networkIdValid, 208
regRejectInfoValid, 208
rejCause, 208
rejectSrvDomain, 208
sysInfoLTE, 208
tac, 208
tacValid, 208
nas_LteM1BandPrefTlv, 196
lteM1BandPref, 197
TlvPresent, 198
nas_LteNb1BandPrefTlv, 198
lteNb1BandPref, 199
TlvPresent, 200
nas_MNRInfo, 210
mcc, 211
mnc, 211
rat, 211
nas_ModePrefTlv, 211
ModePref, 211
TlvPresent, 211
nas_NR5gBandPrefTlv, 219
bits_129_192, 220
bits_193_256, 220
bits_1_64, 220
bits_65_128, 220
TlvPresent, 220
nas_NetSelPrefTlv, 213
NetSelPref, 213
TlvPresent, 213
nas_NetworkStat1x, 213
ActSetCnt, 215
NeighborSetCnt, 215
pActPilotPNElements, 215
pNeighborSetPilotPN, 215
RX_EC_IO, 215
RX_PWR, 215
SO, 215
State, 215
TX_PWR, 215
nas_NetworkStatEVDO, 216
MACIndex, 217
PER, 217
pSectorID, 217
PilotEnergy, 217
RX_PWR, 217
SNR, 217
SectorIDLen, 217
State, 217
nas_OperatorPLMNData, 220
lac1, 221
lac2, 221
mcc, 221
mnc, 221
PLMNRecID, 221
nas_PLMNNetworkName, 228
numInstance, 229
PLMNNetName, 229
nas_PLMNNetworkNameData, 229
codingScheme, 230
countryInitials, 230
longName, 230
longNameLen, 230
longNameSpareBits, 230
shortName, 230
shortNameLen, 230
shortNameSpareBits, 230
nas_PRLPrefTlv, 230
PRLPref, 231
TlvPresent, 231
nas_PhyCaAggPcellInfo, 222
dl_bw_value, 223
freq, 223
iLTEbandValue, 223
pci, 223
TlvPresent, 223
nas_PhyCaAggScellDlBw, 223
dl_bw_value, 223
TlvPresent, 223
nas_PhyCaAggScellIndType, 224
freq, 224
pci, 224
scell_state, 224
TlvPresent, 224
nas_PhyCaAggScellIndex, 223
scell_idx, 224
TlvPresent, 224
nas_PhyCaAggScellInfo, 225
dl_bw_value, 227
freq, 227

iLTEbandValue, 227
pci, 227
scell_state, 227
TlvPresent, 227
nas_PilotSetData, 227
NumPilots, 227
pPilotSetInfo, 227
nas_PilotSetParams, 228
PilotPN, 228
PilotStrength, 228
PilotType, 228
nas_QmiNas3GppNetworkInfo, 233
Desription, 234
Forbidden, 234
InUse, 234
MCC, 234
MNC, 234
Preferred, 234
Roaming, 234
nas_QmiNas3GppNetworkRAT, 235
MCC, 235
MNC, 235
RAT, 235
nas_QmisNasPcsDigit, 235
includes_pcs_digit, 236
MCC, 236
MNC, 236
nas_RFBandInfoElements, 238
activeBandClass, 238
activeChannel, 238
radioInterface, 238
nas_RFInfoTlv, 243
activeBandClass, 244
activeChannel, 244
radioInterface, 244
radioInterfaceSize, 244
TlvPresent, 244
nas_RSRPThresh, 245
pRSRPThresList, 246
RSRPThresListLen, 246
nas_RSRQThresh, 246
pRSRQThresList, 247
RSRQThresListLen, 247
nas_RSSIThresh, 247
pRSSIThresList, 248
RSSIThresListLen, 248
nas_RankIndicatorTlv, 236
count0, 236
count1, 236
TlvPresent, 236
nas_RatDisabledMaskTlv, 236
ratDisabledMask, 237
TlvPresent, 237
nas_RejectReasonTlv, 237
rejectCause, 237
serviceDomain, 238
TlvPresent, 238
nas_RfBandInfoExtFormat, 238
pInstancesSize, 239
pRfBandInfoExtFormatParam, 239
TlvPresent, 239
nas_RfBandInfoExtFormatElements, 239
activeBand, 239
activeChannel, 239
radioInterface, 239
nas_RfBandInfoList, 240
pInstanceSize, 240
pRFBandInfoParam, 240
nas_RfBandwidthInfo, 240
pInstancesSize, 241
pRfBandwidthInfoParam, 241
TlvPresent, 241
nas_RfBandwidthInfoElements, 241
bandwidth, 241
radioInterface, 241
nas_RfDedicatedBandInfo, 242
pInstancesSize, 242
pRfDedicatedBandInfoParam, 242
TlvPresent, 242
nas_RfDedicatedBandInfoElements, 242
dedicatedBand, 243
radioInterface, 243
nas_RoamPrefTlv, 245
RoamPref, 245
TlvPresent, 245
nas_RxSigInfo, 249
isRadioTuned, 250
rsrp, 250
rxChainIndex, 250
rxPower, 250
nas_SLQSSignalStrengthsIndReq, 255
ecioDelta, 256
ecioThresholdList, 256
ecioThresholdListLen, 256
ioDelta, 257
lteRsrpDelta, 257
lteSnrDelta, 257
rsrqDelta, 257
rxSignalStrengthDelta, 257
sinrDelta, 257
sinrThresholdList, 257
sinrThresholdListLen, 257
nas_SLQSSignalStrengthsInformation, 257
ecioInfo, 258
errorRateInfo, 258
io, 258
lteRsrpinfo, 258
lteSnrinfo, 258
rsrqInfo, 258
rxSignalStrengthInfo, 258
sinr, 258
nas_SLQSSignalStrengthsTlv, 258
sSLQSSignalStrengthsInfo, 259
TlvPresent, 259
nas_SccRxInfo, 251
numInstances, 252

rsrq, 252
sigInfo, 252
snr, 252
TlvPresent, 252
nas_SignalStrengthTlv, 255
radioInterface, 255
signalStrength, 255
TlvPresent, 255
nas_SrvDomainPrefTlv, 259
SrvDomainPref, 259
TlvPresent, 259
nas_SrvStatusInfo, 259
isPrefDataPath, 260
srvStatus, 260
nas_TDSCDMAECIOThresh, 263
nas_TDSCDMARSCPThresh, 263
nas_TDSCDMARSSIThresh, 264
nas_TDSCDMASINRCONFThresh, 264
nas_TDSCDMASINRThresh, 265
nas_UMTSInfo, 268
cellID, 270
ecio, 270
geranInst, 270
GeranInstInfo, 270
lac, 270
plmn, 270
psc, 270
rscp, 270
UMTSInstInfo, 270
uarfcn, 270
umtsInst, 270
nas_UMTSinstInfo, 270
umtsEcio, 271
umtsPsc, 271
umtsRscp, 271
umtsUarfcn, 271
nas_UniversalTime, 272
day, 273
dayOfWeek, 273
hour, 273
minute, 273
month, 273
second, 273
year, 273
nas_WCDMAECIOThresh, 274
nas_WCDMAInfoLTENeighborCell, 274
umtsLTENbrCellLen, 275
wcdmaRRCState, 275
nas_WCDMARSSIThresh, 275
nas_WCDMASysInfo, 276
cellId, 278
cellIdValid, 279
hsCallStatus, 279
hsCallStatusValid, 279
hsInd, 279
hsIndValid, 279
lac, 279
lacValid, 279
MCC, 279
MNC, 279
networkIdValid, 279
psc, 279
pscValid, 279
regRejectInfoValid, 279
rejCause, 279
rejectSrvDomain, 279
sysInfoWCDMA, 279
nas_acqOrderPref, 140
acqOrdeLen, 141
pAcqOrder, 141
nas_callBarStatus, 145
csBarStatus, 146
psBarStatus, 146
nas_cellParams, 153
pci, 154
rsrp, 154
rsrq, 154
rssi, 154
srxlev, 154
nas_ciotAcqOrderPref, 154
ciotAcqOrderLen, 155
pCiotAcqOrder, 155
nas_currentPLMN, 159
MCC, 160
MNC, 160
netDescr, 160
netDescrLength, 160
nas_dataSrvCapabilities, 160
dataCapabilities, 161
dataCapabilitiesLen, 161
nas_detailSvcInfo, 163
hdrHybrid, 164
hdrSrvStatus, 164
isSysForbidden, 164
srvCapability, 164
srvStatus, 164
nas_dirNum, 165
dirNum, 165
dirNumLen, 165
nas_ecioListElement, 166
ecio, 166
radioIf, 167
nas_errorRateListElement, 168
errorRate, 169
radioIf, 169
nas_geranInstInfo, 171
geranArfcn, 171
geranBsicBcc, 171
geranBsicNcc, 171
geranRssi, 171
nas_gsmCellInfo, 171
arfcn, 172
band1900, 172
bsicId, 172
cellIdValid, 172
rssi, 172

srxlev, 172
nas_homeSIDNID, 183
numInstances, 183
SidNid, 183
nas_infoInterFreq, 183
cell_resel_priority, 184
cellInterFreqParams, 184
cells_len, 184
earfcn, 184
threshXHigh, 184
threshXLow, 185
nas_lteBandPrefExt, 185
bits_129_192, 186
bits_193_256, 186
bits_1_64, 186
bits_65_128, 186
nas_lteEARFCN, 188
earfcn0, 188
earfcn1, 188
status, 188
nas_lteGsmCellInfo, 188
cellReselPriority, 189
cells_len, 189
GsmCellInfo, 189
nccPermitted, 189
threshGsmHigh, 189
threshGsmLow, 189
nas_ltePCI, 200
earfcn, 201
pci, 201
status, 201
nas_lteRsrpinformation, 201
rsrplevel, 201
nas_lteSnrinformation, 204
snrlevel, 205
nas_lteWcdmaCellInfo, 208
cellReselPriority, 209
cellsLen, 209
threshXhigh, 209
threshXlow, 209
uarfcn, 209
WCDMACellInfo, 209
nas_major
NASLteNasReleaseInfoTlv, 283
nas_minBasedIMSI, 209
imsiM1112, 210
imsiMS1, 210
imsiMS2, 210
mccM, 210
nas_minor
NASLteNasReleaseInfoTlv, 283
nas_namName, 212
namName, 212
namNameLen, 212
nas_netSelectionPref, 212
mcc, 213
mnc, 213
netReg, 213
nas_nmrCellInfo, 217
nmrArfcn, 218
nmrBsic, 218
nmrCellID, 218
nmrLac, 218
nmrPlmn, 218
nmrRxLev, 218
nas_nr5gBandPref, 218
bits_129_192, 219
bits_193_256, 219
bits_1_64, 219
bits_65_128, 219
nas_operatorNameString, 220
PLMNName, 220
nas_operatorPLMNList, 221
numInstance, 222
PLMNData, 222
nas_protocolSubtypeElement, 231
AccessMac, 232
AuthProt, 232
ControlMac, 232
EncryptProt, 232
ForwardMac, 232
IdleState, 232
KeyExchange, 232
MultDisc, 232
PhysicalLayer, 232
ReverseMac, 232
SecProt, 232
VirtStream, 232
nas_qaQmi3Gpp2TimeZone, 233
daylightSavings, 233
leapSeconds, 233
localTimeOffset, 233
nas_release
NASLteNasReleaseInfoTlv, 283
nas_roamIndList, 244
numInstances, 244
radioInterface, 245
roamIndicator, 245
nas_rsrqInformation, 246
radioIf, 246
rsrq, 246
nas_rxInfo, 248
ecio, 249
isRadioTuned, 249
phase, 249
rscp, 249
rsrp, 249
rxPower, 249
nas_rxSignalStrengthListElement, 250
radioIf, 251
rxSignalStrength, 251
nas_servSystem, 253
csAttachState, 254
numRadioInterfaces, 254
psAttachState, 254
radioInterface, 254

regState, 254
selNetwork, 254
nas_serviceProviderName, 252
displayCondition, 252
spn, 252
spnLength, 252
nas_sidNid, 254
nid, 254
sid, 254
nas_sysInfoCommon, 260
isSysForbidden, 262
isSysForbiddenValid, 262
roamStatus, 262
roamStatusValid, 262
srvCapability, 262
srvCapabilityValid, 262
srvDomain, 262
srvDomainValid, 262
nas_timeInfo, 265
day, 267
dayLtSavingAdj, 267
dayOfWeek, 267
hour, 267
minute, 267
month, 267
radioInterface, 267
second, 267
timeZone, 267
TlvPresent, 267
year, 267
nas_trueIMSI, 267
imsiT1112, 268
imsiTS1, 268
imsiTS2, 268
imsiTaddrNum, 268
mccT, 268
nas_txInfo, 268
isInTraffic, 268
txPower, 268
nas_umtsLTENbrCell, 271
cellIsTDD, 272
earfcn, 272
pci, 272
rsrp, 272
rsrq, 272
srxlev, 272
nas_wcdmaCellInfo, 273
cpich_ecno, 274
cpich_rscp, 274
psc, 274
srxlev, 274
nas_wcdmaUARFCN, 279
status, 280
uarfcn, 280
NasGetLTECphyCaInfo, 280
PhyCaAggPcellInfo, 281
PhyCaAggScellDlBw, 281
PhyCaAggScellIndType, 281
PhyCaAggScellIndex, 281
PhyCaAggScellInfo, 281
nasRelInfoTlv
NASQmiCbkNasSwiOTAMessageInd, 289
nccPermitted
nas_lteGsmCellInfo, 189
NeighborSetCnt
nas_NetworkStat1x, 215
netDescr
nas_currentPLMN, 160
netDescrLength
nas_currentPLMN, 160
netInfoLen
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
netReg
nas_netSelectionPref, 213
NetSelPref
nas_NetSelPrefTlv, 213
NASNetSelPreferenceTlv, 284
NetworkID
unpack_nas_SLQSGetServingSystem_t, 702
networkID
nas_CDMASysInfo, 152
networkIdValid
nas_CDMASysInfo, 152
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_WCDMASysInfo, 279
networkInfo
voice_USSDNotificationNetworkInfo, 975
networkInfoLen
unpack_wds_SLQSGetCurrDataSystemStat_t, 898
NetworkType
currNetworkInfo, 52
wds_currNetworkInfo, 979
NewIMSRegistration
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 640
NewImsRegStatus
unpack_imsa_SLQSGetIMSARegStatus_t, 635
NewMMTlv
unpack_sms_SetNewSMSCallback_ind_t, 785
newMTMessageTlv, 294
MTMessageInfo, 294
TlvPresent, 294
newPINLen
uim_unblockUIMPIN, 545
newPINVal
uim_unblockUIMPIN, 545
newPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 453
newPasswdAgain
pack_voice_SLQSVoiceSetCallBarringPassword_t, 453
newPin

pack_dms_UIMUnblockPIN_t, 319
newPwd
voice_newPwdData, 965
newPwdAgain
voice_newPwdData, 965
newValue
pack_dms_UIMChangePIN_t, 315
nextHeader
LibPackTFTIDParams, 117
nid
nas_CDMAInfo, 148
nas_sidNid, 254
unpack_nas_GetHomeNetwork_t, 678
nmrArfcn
nas_nmrCellInfo, 218
nmrBsic
nas_nmrCellInfo, 218
nmrCellID
nas_nmrCellInfo, 218
nmrInst
nas_GERANInfo, 170
nmrLac
nas_nmrCellInfo, 218
nmrPlmn
nas_nmrCellInfo, 218
nmrRxLev
nas_nmrCellInfo, 218
noReplyTimer
voice_callFWExtInfo, 945
voice_callFWInfo, 946
notifType
unpack_voice_SUPSNotificationCallback_ind_t, 876
notification
unpack_omaDmNotificationsTlv_t, 742
notification_Type
unpack_voice_USSDNotificationCallback_ind_t, 876
notificationType
sMSEtwsMessageInfo, 512
notused
unpack_dms_SetCrashAction_t, 583
num_instances
DMScustSettingList, 61
sms_qaQmi3GPP2BroadcastCfgInfo, 504
sms_qaQmi3GPPBroadcastCfgInfo, 505
numApp
slotInf, 497
uim_slotInfo, 544
numCrashes
crashInformation, 51
numEntries
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603
numFeatures
uim_personalizationStatus, 536
numFiles
uim_registerRefresh, 540
NumFlows
unpack_qos_SLQSSetQosEventCallback_ind_t, 766
numInstance
nas_operatorPLMNList, 222
nas_PLMNNetworkName, 229
numInstances
nas_homeSIDNID, 183
nas_roamIndList, 244
nas_SccRxInfo, 252
sms_getMsgWaitingInfo, 500
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t, 800
voice_arrAlertingPattern, 932
voice_arrAlertingType, 933
voice_arrAlphaID, 933
voice_arrCalledPartyNum, 934
voice_arrCallEndReason, 935
voice_arrCallInfo, 935
voice_arrConnectPartyNum, 936
voice_arrDiagInfo, 936
voice_arrRedirPartyNum, 937
voice_arrRemotePartyName, 937
voice_arrRemotePartyNum, 938
voice_arrSvcOption, 938
voice_arrUUSInfo, 939
voice_getCallFWExtInfo, 963
voice_getCallFWInfo, 963
wds_DomainNameList, 986
wds_PCSCFFQDNAddressList, 990
wds_PCSCFIPv4ServerAddressList, 991
numLen
voice_calledPartyInfo, 941
voice_callFWExtInfo, 945
voice_callFWInfo, 947
voice_callingPartyInfo, 950
voice_peerNumberInfo, 967
voice_redirNumInfo, 970
voice_remotePartyNum, 972
numOfFiles
uim_refreshevent, 539
numOfRoutes
sms_setRoutesReq, 510
numOpt
wds_DHCPLeaseOptTlv, 981
wds_DHCPv4OptionList, 984
wdsDhcpv4OptionList, 998
numPI
voice_peerNumberInfo, 967
NumPilots
nas_PilotSetData, 227
numPlan
voice_calledPartyInfo, 941
voice_callFWExtInfo, 946
voice_callingPartyInfo, 950
voice_connectNumInfo, 955
voice_peerNumberInfo, 967
voice_redirNumInfo, 970
numPresInd

voice_connectNumInfo, 955
numQosFlow
unpack_qos_SLQSQosSwiReadDataStats_t, 765
numRadioInterfaces
nas_servSystem, 254
NumRxFilters
unpack_qos_QosFlowInfo_t, 760
numSI
voice_peerNumberInfo, 967
numSlot
uim_cardStatus, 528
NumSupUSBComps
unpack_dms_GetUSBComp_t, 579
NumTxFilters
unpack_qos_QosFlowInfo_t, 761
numType
voice_calledPartyInfo, 941
voice_callFWExtInfo, 946
voice_callingPartyInfo, 950
voice_connectNumInfo, 956
voice_peerNumberInfo, 967
voice_redirNumInfo, 970
number
voice_calledPartyInfo, 941
voice_callFWExtInfo, 945
voice_callFWInfo, 946
voice_callingPartyInfo, 950
voice_ECTNum, 960
voice_peerNumberInfo, 967
voice_redirNumInfo, 970
numberPlan
voice_callFwdTypeAndPlan, 944
numberType
voice_callFwdTypeAndPlan, 944
NxtHdrProto
unpack_qos_swiQosFilter_t, 772

OKtoRefresh
pack_uim_SLQSUIMRefreshOK_t, 436
OMADMEnable
pack_swioma_SLQSOMADMSetSettingsExt_t, 425
OMADMEnabled
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
unpack_swioma_SLQSOMADMGetSettings_t, 823
OMADMLogEnable
pack_swioma_SLQSOMADMSetSettingsExt_t, 425
OTASPStatus
unpack_voice_OTASPStatusCallback_ind_t, 841
oddEvenInd
voice_calledPartySubAdd, 942
OfflineReason
unpack_dms_GetPower_t, 575
offset
uim_readTransparentInfo, 537
oldPINLen
uim_changeUIMPIN, 529
oldPINVal
uim_changeUIMPIN, 529
oldPasswd
pack_voice_SLQSVoiceSetCallBarringPassword_t, 453
oldValue
pack_dms_UIMChangePIN_t, 315
opMode
unpack_dms_SLQSDmsSwiGetPCInfo_t, 590
opaqueId
pack_loc_SLQSLOCInjectSensorData_t, 342
operatingMode
dms_OperatingModeTlv, 53
OperatingModeTlv
unpack_dms_SetEventReport_ind_t, 584
operation
pack_pds_SetPDSDefaults_t, 385
uim_depersonalizationInformation, 530
OperationMode
unpack_dms_GetPower_t, 575
optCode
wds_DHCPOpt, 982
wds_DHCPv4Option, 984
wdsDhcpv4Option, 998
optList
wds_DHCPLeaseOptTlv, 981
optListData
wds_DHCPLeaseOptTlv, 981
optVal
wds_DHCPv4Option, 984
wdsDhcpv4Option, 998
optValLen
wds_DHCPOpt, 982
wds_DHCPv4Option, 984
wdsDhcpv4Option, 998
otaMsgTlv
NASQmiCbkNasSwiOTAMessageInd, 289

p1Status
unpack_dms_UIMGetPINStatus_t, 614
p1UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 614
p1VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 614
p2Status
unpack_dms_UIMGetPINStatus_t, 614
p2UnblockRetriesLeft
unpack_dms_UIMGetPINStatus_t, 614
p2VerifyRetriesLeft
unpack_dms_UIMGetPINStatus_t, 614
p3GPP2TimeInfo
unpack_nas_SLQSGetNetworkTime_t, 693
p3GPPTimeInfo
unpack_nas_SLQSGetNetworkTime_t, 693
p3GppNetworkInfoInstances
unpack_nas_PerformNetworkScan_t, 686
p3GppNetworkInstanceSize
unpack_nas_PerformNetworkScan_t, 686

p3gppRelease
pack_wds_SLQSSet3GPPConfigItem_t, 476
pAAASPI
pack_wds_SetMobileIPProfile_t, 469
PACK_WDS_IPV4
wds.h, 1393
PACK_WDS_IPV6
wds.h, 1393
pAMRStatus
pack_voice_SLQSVoiceGetConfig_t, 449
pAPN
PackSwiAVMSSettingsAPNInfo, 488
pAPNClass
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pAPNClass3GPP2
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAPNDisabledFlag
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pAPNEnabled3GPP2
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAPNInfo
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
pAPNName
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pAPNnameSize
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pAVRXAVCHeadroom
audio_RXAVCList, 39
pAVRXAVCSens
audio_RXAVCList, 39
pAccelAcceptReady
unpack_loc_SensorStreamingCallback_Ind_t, 666
pAccelSamplesAccepted
unpack_loc_InjectSensorDataCallback_Ind_t, 657
pAccelTempAcceptReady
unpack_loc_SensorStreamingCallback_Ind_t, 666
pAccelTempSamplesAccepted
unpack_loc_InjectSensorDataCallback_Ind_t, 657
pAccolc
unpack_nas_GetACCOLC_t, 671
pAccuracy
unpack_pds_GetPDSDefaults_t, 743
pAcqOrder
nas_acqOrderPref, 141
nas_AcqOrderPrefTlv, 141
pAcqOrderPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pActPilotPNElements
nas_NetworkStat1x, 215
pActiveTimer
pack_dms_SLQSSetPowerSaveModeConfig_t, 312
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pAddCDMASysInfo
unpack_nas_SLQSGetSysInfo_t, 707
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pAddGSMSysInfo
unpack_nas_SLQSGetSysInfo_t, 707
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pAddHDRSysInfo
unpack_nas_SLQSGetSysInfo_t, 707
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pAddLTESysInfo
unpack_nas_SLQSGetSysInfo_t, 707
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pAddWCDMASysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pAddrAllocPref
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 107
pAddress
pack_wds_SetMobileIPProfile_t, 469
pAirTimer
pack_voice_SLQSVoiceGetConfig_t, 449
pAirTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 866
pAirTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 455
pAirTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 869
pAlertType
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pAlertingPattern
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pAllowLinger
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAlphaID
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 799
pAlphaIDInfo
unpack_voice_SLQSOriginateUSSD_t, 844
unpack_voice_SLQSVoiceDialCall_t, 846
unpack_voice_SLQSVoiceGetCallBarring_t, 851
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
unpack_voice_SLQSVoiceGetCallInfo_t, 855
unpack_voice_SLQSVoiceGetCallWaiting_t, 857
unpack_voice_SLQSVoiceGetCLIP_t, 858
unpack_voice_SLQSVoiceGetCLIR_t, 859
unpack_voice_SLQSVoiceGetCNAP_t, 861
unpack_voice_SLQSVoiceGetCOLP_t, 862

unpack_voice_SLQSVoiceGetCOLR_t, 864
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 868
unpack_voice_SLQSVoiceSetSUPSService_t, 871
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pAltitudeAssumed
unpack_loc_PositionRpt_Ind_t, 664
pAltitudeWrtEllipsoid
pack_pds_SLQSPDSInjectPositionData_t, 390
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pAltitudeWrtMeanSeaLevel
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pAltitudeWrtSealevel
pack_pds_SLQSPDSInjectPositionData_t, 390
pAmrMode
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pAmrOctetAligned
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pAmrWBMode
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pAmrWBOctetAligned
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pAmrWbEnable
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pApnString
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pApnStringSize
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pApnname
pack_wds_SetDefaultProfile_t, 465
pAppName
loc_LocApplicationInfo, 132
pAppPriority
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAppProvider
loc_LocApplicationInfo, 132
pAppSubType
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 730
pAppType
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAppVersion
loc_LocApplicationInfo, 132
pApplication
pack_nas_SetCDMANetworkParameters_t, 348
pApplicationInfo
pack_loc_Start_t, 345
pArrAlertingPattern
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrAlertingType
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrAlphaID
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrCallEndReason
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrCallInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrCalledPartyNum
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrConnectPartyNum
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrDiagInfo
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrRedirPartyNum
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrRemotePartyName
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrRemotePartyNum
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrSvcOption
unpack_voice_allCallStatusCallback_ind_t, 839
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pArrUUSInfo
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
pAuth
pack_wds_SLQSStartDataSession_t, 480
pAuthPassword
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAuthPassword_tSize
LibPackprofile_3GPP2, 113
pAuthPasswordSize
LibpackProfile3GPP2, 102
pAuthProtocol
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAuthRetryCount
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pAuthStatus
unpack_nas_GetANAAAAuthenticationStatus_t, 672
pAuthTimeout
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113

pAuthenticateResult
unpack_uim_SLQSUIMAuthenticate_t, 830
pAuthenticationPref
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pAutoAnsStatus
unpack_voice_SLQSVoiceSetConfig_t, 869
pAutoAnswer
pack_voice_SLQSVoiceGetConfig_t, 449
pack_voice_SLQSVoiceSetConfig_t, 455
pAutoAnswerStat
unpack_voice_SLQSVoiceGetConfig_t, 866
pAutoReboot
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
pAutoSelection
unpack_uim_SLQSUIMGetConfiguration_t, 832
pAutosdm
pack_swioma_SLQSOMADMSetSettings_t, 424
pBandPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pBdsSVInfo
pack_loc_Delete_Assist_Data_t, 330
pBinaryUpdateSessionInfo
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 811
pBroadcast
pack_nas_SetCDMANetworkParameters_t, 348
pBroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 405
unpack_sms_SLQSGetSmsBroadcastConfig_t, 789
pBurstDTMFLengths
pack_voice_SLQSVoiceBurstDTMF_t, 443
pCCResType
unpack_voice_SLQSVoiceGetCallBarring_t, 851
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
unpack_voice_SLQSVoiceGetCallWaiting_t, 857
unpack_voice_SLQSVoiceGetCLIP_t, 858
unpack_voice_SLQSVoiceGetCLIR_t, 859
unpack_voice_SLQSVoiceGetCNAP_t, 861
unpack_voice_SLQSVoiceGetCOLP_t, 862
unpack_voice_SLQSVoiceGetCOLR_t, 864
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 868
pCCResultType
unpack_voice_SLQSVoiceDialCall_t, 846
unpack_voice_SLQSVoiceSetSUPSService_t, 871
pCCSUPSType
unpack_voice_SLQSVoiceDialCall_t, 846
unpack_voice_SLQSVoiceGetCallBarring_t, 851
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
unpack_voice_SLQSVoiceGetCallWaiting_t, 857
unpack_voice_SLQSVoiceGetCLIP_t, 858
unpack_voice_SLQSVoiceGetCLIR_t, 860
unpack_voice_SLQSVoiceGetCNAP_t, 861
unpack_voice_SLQSVoiceGetCOLP_t, 862
unpack_voice_SLQSVoiceGetCOLR_t, 864
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 868
unpack_voice_SLQSVoiceSetSUPSService_t, 871
pCCSuppsType
unpack_voice_SLQSOriginateUSSD_t, 844
pCDMABroadcastConfig
pack_sms_SLQSSetSmsBroadcastConfig_t, 405
unpack_sms_SLQSGetSmsBroadcastConfig_t, 789
pCDMAChannel
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
pCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 356
pCDMAECIOThreshList
nas_CDMAECIOThresh, 147
pCDMAFrameErrRate
unpack_nas_SLQSGetErrorRate_t, 692
pCDMAInfo
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 356
pCDMARSSIThreshList
nas_CDMARSSIThresh, 149
pCDMASigInfo
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
pCDMASrvStatusInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pCDMASysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pCLIPResp
unpack_voice_SLQSVoiceGetCLIP_t, 858
pCLIPstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCLIRCause
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pCLIRResp
unpack_voice_SLQSVoiceGetCLIR_t, 860
pCLIRType
pack_voice_SLQSVoiceDialCall_t, 445
pCLIRstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874

pCNAPResp
unpack_voice_SLQSVoiceGetCNAP_t, 861
pCNAPstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCOLPResp
unpack_voice_SLQSVoiceGetCOLP_t, 862
pCOLPstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCOLRResp
unpack_voice_SLQSVoiceGetCOLR_t, 864
pCOLRstatus
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
PCSCFAddrPCO
unpack_wds_SLQSGetRuntimeSettings_t, 906
PCSCFFQDNAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 906
pCSCFPortName
pack_ims_SLQSSetRegMgrConfig_t, 327
pCSCFPortNameLen
pack_ims_SLQSSetRegMgrConfig_t, 327
pCSGID
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
PCTlv
unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 627
pCUGIndex
unpack_voice_SUPSNotificationCallback_ind_t, 876
pCUGInfo
pack_voice_SLQSVoiceDialCall_t, 445
PCURTlv
unpack_ims_SLQSSMSCfgCallBack_ind_t, 631
pCallBarPasswd
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCallBarringPasswd
pack_voice_SLQSVoiceSetSUPSService_t, 457
pCallFWNum
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCallFWTimerVal
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCallForwardingNumber
pack_voice_SLQSVoiceSetSUPSService_t, 457
pCallFwdInfo
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pCallFwdTypeAndPlan
pack_voice_SLQSVoiceSetSUPSService_t, 457
pCallID
pack_voice_SLQSVoiceManageCalls_t, 451
pack_voice_SLQSVoiceSendFlash_t, 452
pack_voice_SLQSVoiceStartContDTMF_t, 458
unpack_voice_SLQSVoiceBurstDTMF_t, 845
unpack_voice_SLQSVoiceDialCall_t, 846
unpack_voice_SLQSVoiceGetCallBarring_t, 851
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
unpack_voice_SLQSVoiceGetCallWaiting_t, 857
unpack_voice_SLQSVoiceGetCLIP_t, 858
unpack_voice_SLQSVoiceGetCLIR_t, 859
unpack_voice_SLQSVoiceGetCNAP_t, 861
unpack_voice_SLQSVoiceGetCOLP_t, 862
unpack_voice_SLQSVoiceGetCOLR_t, 864
unpack_voice_SLQSVoiceSendFlash_t, 867
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 868
unpack_voice_SLQSVoiceSetSUPSService_t, 871
unpack_voice_SLQSVoiceStartContDTMF_t, 871
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
voice_burstDTMFInfo, 940
pCallId
pack_voice_SLQSVoiceAnswerCall_t, 442
pack_voice_SLQSVoiceEndCall_t, 445
unpack_voice_SLQSOriginateUSSD_t, 844
unpack_voice_SLQSVoiceAnswerCall_t, 845
unpack_voice_SLQSVoiceEndCall_t, 847
pCallInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pCallPartySubAdd
pack_voice_SLQSVoiceDialCall_t, 445
pCallType
pack_voice_SLQSVoiceDialCall_t, 445
pCallWaitInd
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pCalledPartyInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pCallerIDInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pCallerNameInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pCallingPartyInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pCardResult
unpack_uim_ReadTransparent_t, 828
unpack_uim_SLQSUIMAuthenticate_t, 831
unpack_uim_SLQSUIMGetFileAttributes_t, 833
pCardStatus
unpack_uim_GetCardStatus_t, 827
unpack_uim_SLQSUIMSetStatusChangeCall-Back_ind_t, 835
pCcResultType
unpack_voice_SLQSOriginateUSSD_t, 844
pCellDataMask
pack_pds_ResetPDSData_t, 383
pCellDb
pack_loc_Delete_Assist_Data_t, 330
pChangeDuration
pack_nas_SLQSInitiateNetworkRegistration_t, 352
pChgDuration
pack_nas_SLQSSetSysSelectionPref_t, 371

pack_nas_SLQSSetSysSelectionPrefExt_t, 381
pCiotAcqOrder
nas_ciotAcqOrderPref, 155
nas_CiotAcqOrderPrefTlv, 156
pCiotAcqOrderPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pCiotLteOpMode
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
pCiotLteOpModePref
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pClientPerformOperationFlag
pack_swiavms_SLQSAVMSSendSelection_t, 413
pClkInfo
pack_loc_Delete_Assist_Data_t, 330
pCodecSTGain
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pColorCode
unpack_nas_SLQSNasGetHDRColorCode_t, 720
pConfig
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 811
pConfigAltitudeAssumed
pack_loc_Start_t, 345
pConfigurationMask
pack_uim_SLQSUIMGetConfiguration_t, 433
pConnectNumInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 855
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pConnectionRequest
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
pConnectionRetryTimers
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
pCreateProfileOut
unpack_wds_SLQSCreateProfile_t, 894
pCurAMRConfig
unpack_voice_SLQSVoiceGetConfig_t, 866
pCurPrefVoiceSO
unpack_voice_SLQSVoiceGetConfig_t, 866
pCurProfile
pack_wds_SLQSCreateProfile_t, 470
pCurVoiceDomainPref
unpack_voice_SLQSVoiceGetConfig_t, 866
pCurVoicePrivacyPref
unpack_voice_SLQSVoiceGetConfig_t, 866
pCurrChannelRateInd
pack_wds_SLQSWdsSetEventReport_t, 482
pCurrDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 482
pCurrImgInfo
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603
pCurrPrefDataSysInd
pack_wds_SLQSWdsSetEventReport_t, 482
pCurrTTYMode
unpack_voice_SLQSVoiceGetConfig_t, 866
pCurrentPersonality
nas_HDRPersonality_Ind_Data, 179
unpack_nas_SLQSSwiGetHDRPersonality_t, 730
pCurrentPrsnlty
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 731
pCustSettingInfo
DMSgetCustomFeatureV2, 62
pCustSettingList
DMSgetCustomFeatureV2, 62
pCustomSCP
pack_nas_SetCDMANetworkParameters_t, 348
pCwtMute
pack_swiaudio_SLQSSetM2MAudioProfile_t, 410
pack_swiaudio_SLQSSetM2MAVMute_t, 412
pCycleLen
pack_nas_SLQSNASSeteDRXParams_t, 362
unpack_nas_SLQSNASGeteDRXParams_t, 719
pDDTMInd
pack_nas_SLQSNasIndicationRegisterExt_t, 360
PDOP
loc_precisionDilution, 133
PDPType
unpack_wds_SLQSGetRuntimeSettings_t, 906
pDRCParams
unpack_nas_SLQSSwiGetHRPDStats_t, 731
pDTMFTXGain
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pData
pack_cat_CATSendEnvelopeCommand_t, 301
pack_cat_CATSendTerminalResponse_t, 302
pDataBearer
unpack_wds_GetDataBearerTechnology_t, 882
pDataBearerTechInd
pack_wds_SLQSWdsSetEventReport_t, 482
pDataCallStatusChangeInd
pack_wds_SLQSWdsSetEventReport_t, 482
pDataMode
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pDataRate
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pDataSessionStatus
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
pDataSrc
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pDataStatusDetail
unpack_nas_SLQSSwiNetworkDebug_t, 734
pDataSystemStatusChangeInd

pack_wds_SLQSWdsSetEventReport_t, 482
pDayltSavAdj
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 724
pDefaultPDNEnabled
pack_wds_SLQSSet3GPPConfigItem_t, 476
pDeferTime
pack_swiavms_SLQSAVMSSendSelection_t, 413
pack_swioma_SLQSOMADMSendSelection_t, 422
pDestSMSContent
pack_dms_SLQSSwiSetDyingGaspCfg_t, 313
packgetDyingGaspCfg, 484
pDestSMSNum
pack_dms_SLQSSwiSetDyingGaspCfg_t, 313
packgetDyingGaspCfg, 484
pDetachAction
pack_nas_SLQSSwiPSDetach_t, 382
pDeviceConfigDetail
unpack_nas_SLQSSwiNetworkDebug_t, 734
pDiagInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pDigitBuff
voice_burstDTMFInfo, 940
pDirNum
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
pDispInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pDormancyStatusInd
pack_wds_SLQSWdsSetEventReport_t, 483
pDualStandByPrefInd
pack_nas_SLQSNasIndicationRegisterExt_t, 360
pDuration
unpack_pds_GetXTRAValidity_t, 747
pDurationDueToOOS
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pDurationThreshold
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pECIOThresList
nas_ECIOThresh, 167
pECIOThresh
pack_nas_SLQSConfigSigInfo_t, 351
pECMode
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pECTNum
unpack_voice_SUPSNotificationCallback_ind_t, 876
PER
nas_NetworkStatEVDO, 217
pEVDOPageMonPerChangeInd
pack_wds_SLQSWdsSetEventReport_t, 483
pEarMute
pack_swiaudio_SLQSSetM2MAudioProfile_t, 410
pEarlyWakeupTime
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pEdrxEnable
pack_nas_SLQSNASSeteDRXParams_t, 362
unpack_nas_SLQSNASGeteDRXParams_t, 719
pEhrpdMTUSize
unpack_swidms_SLQSSwiDmsGetMTU_t, 815
pEmerMode
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pEmergencyCategory
pack_voice_SLQSVoiceDialCall_t, 445
pEnableNotification
pack_wds_DHCPv4ClientLeaseChange_t, 459
pEnabled
pack_wds_SetMobileIPProfile_t, 469
pEnabledStatus
unpack_pds_GetPDSState_t, 744
pEncryptData
pack_uim_ReadTransparent_t, 429
pEncryptedData
unpack_uim_ReadTransparent_t, 828
pEncryptedPIN1
pack_uim_VerifyPin_t, 439
unpack_uim_ChangePin_t, 827
unpack_uim_SetPinProtection_t, 829
unpack_uim_UnblockPin_t, 836
unpack_uim_VerifyPin_t, 836
pErrorRateInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pExtDispInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pExtDispRecInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pExtErrCode
UnPackGetProfileSettingOut, 919
pExtErrorCode
unpack_wds_SLQSModifyProfile_t, 907
pFailCause
unpack_voice_SLQSVoiceGetCallBarring_t, 851
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
unpack_voice_SLQSVoiceGetCallWaiting_t, 857
unpack_voice_SLQSVoiceGetCLIP_t, 858
unpack_voice_SLQSVoiceGetCLIR_t, 860
unpack_voice_SLQSVoiceGetCNAP_t, 861
unpack_voice_SLQSVoiceGetCOLP_t, 862
unpack_voice_SLQSVoiceGetCOLR_t, 864
unpack_voice_SLQSVoiceManageCalls_t, 866
unpack_voice_SLQSVoiceSetCallBarringPassword-_t, 868
unpack_voice_SLQSVoiceSetSUPSService_t, 871
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pFailureReason
unpack_wds_SLQSStartDataSession_t, 916

pFileAttributes
unpack_uim_SLQSUIMGetFileAttributes_t, 833
pFixId
unpack_loc_PositionRpt_Ind_t, 664
pFlag
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pFlashPayLd
pack_voice_SLQSVoiceSendFlash_t, 452
pFlashType
pack_voice_SLQSVoiceSendFlash_t, 452
pFollowOnDC
sms_sendAsyncsmsParams, 508
pForceOnDC
sms_sendAsyncsmsParams, 508
pForceRev0
pack_nas_SetCDMANetworkParameters_t, 348
pFwAutoCheck
pack_swioma_SLQSOMADMSetSettings_t, 424
pFwAutoSDM
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
pGERANInfo
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pGPRSMinimumQoS
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pGPRSRequestedQos
LibpackProfile3GPP, 96
LibPackprofile_3GPP, 108
pGPSDataMask
pack_pds_ResetPDSData_t, 384
pGPSWeek
unpack_pds_GetXTRAValidity_t, 747
pGPSWeekOffset
unpack_pds_GetXTRAValidity_t, 747
pGSMBER
unpack_nas_SLQSGetErrorRate_t, 692
pGSMCallBarringSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pGSMCipherDomainSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pGSMRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pGSMRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 356
pGSMRSSIThreshList
nas_GSMRSSIThresh, 173
pGSMSigInfo
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
pGSMSrvStatusInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pGSMSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pGWAcqOrderPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pGenerator
pack_swiaudio_SLQSGetM2MAudioProfile_t, 407
pack_swiaudio_SLQSSetM2MAudioProfile_t, 410
pGetCallFWExtInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
pGetCallFWInfo
unpack_voice_SLQSVoiceGetCallForwarding-Status_t, 852
pGetCustomInput
DMSgetCustomFeatureV2, 62
pGetDyingGaspCfg
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 602
pGetDyingGaspStatistics
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 602
pGetIndicationRegInfo
unpack_sms_SLQSGetIndicationRegister_t, 787
pGetMsgWaitingInfoResp
unpack_sms_SLQSGetMessageWaiting_t, 787
pGetTransLayerInfo
unpack_sms_SLQSGetTransLayerInfo_t, 790
pGetTransNWRegInfo
unpack_sms_SLQSGetTransNWRegInfo_t, 791
pGnssData
pack_loc_Delete_Assist_Data_t, 330
pGpsTime
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pGyroAcceptReady
unpack_loc_SensorStreamingCallback_Ind_t, 666
pGyroSamplesAccepted
unpack_loc_InjectSensorDataCallback_Ind_t, 657
pGyroTempAcceptReady
unpack_loc_SensorStreamingCallback_Ind_t, 666
pGyroTempSamplesAccepted
unpack_loc_InjectSensorDataCallback_Ind_t, 657
pHA2002bis
pack_wds_SetMobileIPParameters_t, 467
pHAAuthenticator
pack_wds_SetMobileIPParameters_t, 467
pHASPI
pack_wds_SetMobileIPProfile_t, 469
pHDRECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRECIOThreshList
nas_HDRECIOThresh, 178
pHDRIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 356

pHDRIOThreshList
nas_HDRIOThresh, 178
pHDRNewUATIAssInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pHDRPackErrRate
unpack_nas_SLQSGetErrorRate_t, 692
pHDRPersInd
unpack_nas_SLQSSwiHDRPersonalityCallback_-Ind_t, 733
pHDRRSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRRSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRRSSIThreshList
nas_HDRRSSIThresh, 179
pHDRSINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRSINRThresList
nas_HDRSINRThresh, 180
pHDRSINRThresh
pack_nas_SLQSConfigSigInfo_t, 351
pack_nas_SLQSNasConfigSigInfo2_t, 356
pHDRSINRThreshList
nas_HDRSINRThreshold, 181
pHDRSessionCloseInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pHDRSigInfo
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
pHDRSrvStatusInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pHDRSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pHTTPStatus
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
pHaltSubscription
unpack_uim_SLQSUIMGetConfiguration_t, 832
pHeading
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHeadingUnc
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHomeSIDNID
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
pHorCirConf
unpack_loc_BestAvailPos_Ind_t, 647
pHorConfidence
unpack_loc_PositionRpt_Ind_t, 664
pHorEllpConf
unpack_loc_BestAvailPos_Ind_t, 647
pHorReliability
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHorUncCircular
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHorUncEllipseOrientAzimuth
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHorUncEllipseSemiMajor
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHorUncEllipseSemiMinor
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pHorizontalAccuracyLvl
pack_loc_Start_t, 345
pHorizontalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 390
pHorizontalUncCircular
pack_pds_SLQSPDSInjectPositionData_t, 390
pHotSwapStatus
unpack_uim_GetCardStatus_t, 827
pHrpdMTUSize
unpack_swidms_SLQSSwiDmsGetMTU_t, 815
pHwConfig
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 479
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 915
PI
voice_calledPartyInfo, 941
voice_callerIDInfo, 943
voice_callFWExtInfo, 946
voice_callingPartyInfo, 950
voice_redirNumInfo, 970
PIFACEId
pack_swiaudio_SLQSSetM2MAudioAVCFG_t, 409
pIMSDomain
pack_ims_SLQSSetIMSUserConfig_t, 324
unpack_ims_SLQSGetIMSUserConfig_t, 620
pIMSDomainLen
pack_ims_SLQSSetIMSUserConfig_t, 324
unpack_ims_SLQSGetIMSUserConfig_t, 620
pIMSTestMode
pack_ims_SLQSSetRegMgrConfig_t, 327
unpack_ims_SLQSGetRegMgrConfig_t, 624
pIOThresList
nas_IOThresh, 185
pIOThresh
pack_nas_SLQSConfigSigInfo_t, 351
pIPv4AddrInfo
unpack_loc_GetServer_Ind_t, 654
pIPv4AddrPref
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pIPv4Config
pack_loc_SLQSLOCSetServer_t, 344
pIPv6AddPref
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pIPv6AddrInfo
unpack_loc_GetServer_Ind_t, 654

pIPv6Config
pack_loc_SLQSLOCSetServer_t, 344
pIgnoreHotSwapSwitch
pack_uim_SLQSUIMPowerUp_t, 434
pImCnFlag
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pImageList
pack_fms_SetImagesPreference_t, 321
unpack_fms_GetImagesPreference_t, 617
pIndicationToken
pack_uim_ChangePin_t, 428
pack_uim_ReadTransparent_t, 430
pack_uim_SetPinProtection_t, 431
pack_uim_SLQSUIMAuthenticate_t, 431
pack_uim_SLQSUIMGetFileAttributes_t, 433
pack_uim_UnblockPin_t, 438
pack_uim_VerifyPin_t, 439
unpack_uim_ChangePin_t, 827
unpack_uim_ReadTransparent_t, 828
unpack_uim_SetPinProtection_t, 829
unpack_uim_SLQSUIMAuthenticate_t, 831
unpack_uim_SLQSUIMGetFileAttributes_t, 833
unpack_uim_UnblockPin_t, 836
unpack_uim_VerifyPin_t, 836
pInfo
pack_voice_AnswerUSSD_t, 440
pack_voice_OriginateUSSD_t, 440
pInstanceSize
nas_RfBandInfoList, 240
pInstancesSize
nas_RfBandInfoExtFormat, 239
nas_RfBandwidthInfo, 241
nas_RfDedicatedBandInfo, 242
pIntermediateReportState
pack_loc_Start_t, 345
pInterval
unpack_pds_GetPDSDefaults_t, 743
unpack_pds_GetXTRAAutomaticDownload_t, 746
pIpcpAckTimeout
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 113
pIpcpCreqRetryCount
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 114
pIsPcscfAddressNedded
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 114
pKeyReferenceID
pack_uim_ChangePin_t, 428
pack_uim_SetPinProtection_t, 431
pack_uim_UnblockPin_t, 438
pack_uim_VerifyPin_t, 439
PLMNData
nas_operatorPLMNList, 222
PLMNName
nas_operatorNameString, 220
PLMNNetName
nas_PLMNNetworkName, 229
PLMNRecID
nas_OperatorPLMNData, 221
pLTEAttachProfile
pack_wds_SLQSSet3GPPConfigItem_t, 476
pLTEAttachProfileList
pack_wds_SLQSSet3GPPConfigItem_t, 476
pLTEBandPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 715
pLTEBandPrefExt
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
pLTECphyCa
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pLTEInfo
unpack_nas_SLQSNasSwiModemStatus_t, 727
pLTEInfoInterfreq
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pLTEInfoIntrafreq
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pLTEInfoNeighboringGSM
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pLTEInfoNeighboringWCDMA
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pLTEOperationMode
nas_LTEOperationMode, 200
unpack_nas_SLQSNasGetRFInfo_t, 721
pLTERSRPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 356
pLTERSRPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTERSRPThreshList
nas_LTERSRPThresh, 202
pLTERSRQDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTERSRQThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTERSRQThreshList
nas_LTERSRQThresh, 202
pLTERSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTERSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTERSSIThreshList
nas_LTERSSIThresh, 203
pLTESNRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTESNRThresList
nas_LTESNRThresh, 205
pLTESNRThresh
pack_nas_SLQSConfigSigInfo_t, 351
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTESNRThreshList
nas_LTESNRThreshold, 206
pLTESigInfo
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725

pLTESigRptCfg
pack_nas_SLQSConfigSigInfo_t, 351
pLTESigRptConfig
pack_nas_SLQSNasConfigSigInfo2_t, 357
pLTESrvStatusInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pLTESysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pLTEVoiceSupportSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pLatitude
pack_pds_SLQSPDSInjectPositionData_t, 390
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pLcpAckTimeout
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 114
pLcpCreqRetryCount
LibpackProfile3GPP2, 102
LibPackprofile_3GPP2, 114
pLeapSeconds
unpack_loc_PositionRpt_Ind_t, 664
pLineCtrlInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pLinktimer
pack_sms_SendSMS_t, 398
sms_sendAsyncsmsParams, 508
pLongitude
pack_pds_SLQSPDSInjectPositionData_t, 390
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pLteEARFCN
pack_nas_SLQSNASSwiSetChannelLock_t, 364
unpack_nas_SLQSNASSwiGetChannelLock_t, 726
pLteM1BandPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pLteNb1BandPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pLtePCI
pack_nas_SLQSNASSwiSetChannelLock_t, 364
unpack_nas_SLQSNASSwiGetChannelLock_t, 726
pMCC
unpack_nas_GetHomeNetwork3GPP2_t, 676
pMICGainSelect
unpack_audio_SLQSGetAudioPathConfig_t, 548
pMIPStatusInd
pack_wds_SLQSWdsSetEventReport_t, 483
pMNAAA
pack_wds_SetMobileIPProfile_t, 469
pMNC
unpack_nas_GetHomeNetwork3GPP2_t, 676
pMNCIncPCSDigStat
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
pMNHA
pack_wds_SetMobileIPProfile_t, 469
pMNRInfo
pack_nas_SLQSInitiateNetworkRegistration_t, 352
pMTUSize3gpp
unpack_swidms_SLQSSwiDmsGetMTU_t, 815
pMagneticDeviation
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 664
pManagedRoamingInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pMaxStorageSizeReq
pack_sms_SLQSSmsGetMaxStorageSize_t, 406
pMaxStorageSizeResp
unpack_sms_SLQSSmsGetMaxStorageSize_t, 795
pMessage
pack_sms_SaveSMS_t, 398
pack_sms_SendSMS_t, 399
sms_sendAsyncsmsParams, 508
pMessageIndex
pack_sms_SLQSDeleteSMS_t, 400
unpack_sms_SaveSMS_t, 783
pMessageMode
pack_sms_SLQSDeleteSMS_t, 400
pack_sms_SLQSGetSMS_t, 401
pack_sms_SLQSGetSMSList_t, 402
pack_sms_SLQSModifySMSStatus_t, 403
sms_maxStorageSizeReq, 502
pMessageProtocol
unpack_sms_SLQSSmsGetMessageProtocol_t, 795
pMessageTag
pack_sms_SLQSDeleteSMS_t, 400
pMicMute
pack_swiaudio_SLQSSetM2MAudioProfile_t, 410
pMinBasedIMSI
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
pMinIntervalTime
pack_loc_Start_t, 345
pMinSessionExpiryTimer
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pMncPcsDigitStatus
pack_nas_SLQSInitiateNetworkRegistration_t, 352
pMncPcsStatus
pack_nas_SLQSGetPLMNName_t, 352
pMode
pack_wds_SetMobileIPParameters_t, 467
pModePref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716

pMuxID
pack_qos_BindDataPort_t, 395
pNAI
pack_wds_SetMobileIPProfile_t, 469
pNAMNameInfo
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
pNID
unpack_nas_GetHomeNetwork3GPP2_t, 676
pNITZInformation
unpack_nas_SLQSGetOperatorNameData_t, 694
pNSEnable
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pNSSAudioCtrl
unpack_voice_VoiceInfoRecCallback_ind_t, 878
pNSSRelease
unpack_voice_VoiceInfoRecCallback_ind_t, 879
PNTlv
unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 627
pNamID
pack_voice_SLQSVoiceGetConfig_t, 449
pName
pack_wds_SetDefaultProfile_t, 465
unpack_nas_GetHomeNetwork3GPP2_t, 676
pNegoDnsSrvrPref
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pNeighborSetPilotPN
nas_NetworkStat1x, 215
pNetSelPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pNetworkMode
pack_pds_SLQSGetAGPSConfig_t, 388
pack_pds_SLQSSetAGPSConfig_t, 391
pNetworkRejectInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pNetworkStat1x
unpack_nas_SLQSSwiNetworkDebug_t, 734
pNetworkStatEVDO
unpack_nas_SLQSSwiNetworkDebug_t, 734
pNetworkTimeInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pNewPwdData
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pNotifStore
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
pNotification
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 812
pNotificationStore
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
pNr5gBandPref
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pNumberOfPhySlot
unpack_uim_SLQSUIMGetSlotsStatus_t, 834
pNw2DescDisp
unpack_nas_GetHomeNetwork3GPP2_t, 676
pNw2DescEnc
unpack_nas_GetHomeNetwork3GPP2_t, 677
pNw2DescLen
unpack_nas_GetHomeNetwork3GPP2_t, 677
pNw2MCC
unpack_nas_GetHomeNetwork3GPP2_t, 677
pNw2MNC
unpack_nas_GetHomeNetwork3GPP2_t, 677
pNw2Name
unpack_nas_GetHomeNetwork3GPP2_t, 677
pOTASPStatus
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pObjectVer
unpack_nas_SLQSSwiNetworkDebug_t, 735
pOffLength
unpack_voice_DTMFEventCallback_ind_t, 840
pOnLength
unpack_voice_DTMFEventCallback_ind_t, 840
pOpaqueIdentifier
unpack_loc_InjectSensorDataCallback_Ind_t, 657
pOperation
unpack_pds_GetPDSDefaults_t, 743
pOperatorNameString
unpack_nas_SLQSGetOperatorNameData_t, 694
pOperatorPLMNList
unpack_nas_SLQSGetOperatorNameData_t, 694
pOptList
wds_DHCPv4OptionList, 984
wdsDhcpv4OptionList, 999
pOptVal
wds_DHCPOpt, 982
pPCMParams
pack_swiaudio_SLQSSetM2MAudioAVCFG_t, 409
pPCSCFPort
unpack_ims_SLQSGetRegMgrConfig_t, 624
pPCSInstance
unpack_nas_PerformNetworkScan_t, 686
pPCSInstanceSize
unpack_nas_PerformNetworkScan_t, 686
pPDNInactivTimeout
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPDNInactivTimeout3GPP2
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pPDPtype
LibpackProfile3GPP, 97

LibPackprofile_3GPP, 108
pPLMNNetworkName
unpack_nas_SLQSGetOperatorNameData_t, 694
pPRLPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 381
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pPWD
PackSwiAVMSSettingsAPNInfo, 488
pPackageID
pack_swiavms_SLQSAVMSSendSelection_t, 413
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 812
pPagingTimeWindow
pack_nas_SLQSNASSeteDRXParams_t, 362
unpack_nas_SLQSNASGeteDRXParams_t, 719
pPass
pack_wds_SLQSStartDataSession_t, 480
pPassword
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pack_wds_SetDefaultProfile_t, 465
pPasswordSize
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPcscfAddrUsingDhcp
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPcscfAddrUsingPCO
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPdnType
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pPdpAccessConFlag
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPdpContext
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPdpDataCompType
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPdpHdrCompType
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPeriodInfo
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
pPeriodicUpdateTimer
pack_dms_SLQSSetPowerSaveModeConfig_t, 312
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pPeripheralEndPointID
pack_qos_BindDataPort_t, 395
pPeroidsInfo
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
pPersonalityListLength
nas_HDRPersonality_Ind_Data, 179
unpack_nas_SLQSSwiGetHDRPersonality_t, 730
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 731
pPersonalizationStatus
unpack_uim_SLQSUIMGetConfiguration_t, 832
pPhoneCtxtURI
pack_ims_SLQSSetIMSSMSConfig_t, 323
unpack_ims_SLQSGetIMSSMSConfig_t, 620
pPhoneCtxtURILen
pack_ims_SLQSSetIMSSMSConfig_t, 323
unpack_ims_SLQSGetIMSSMSConfig_t, 620
pPilotSetData
unpack_nas_SLQSSwiGetHRPDStats_t, 731
pPilotSetInfo
nas_PilotSetData, 227
pPollingTimer
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
pPositionSource
pack_pds_SLQSPDSInjectPositionData_t, 390
pPppSessCloseTimer1x
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pPppSessCloseTimerDO
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pPrecisionDilution
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pPrefVoiceDomain
pack_voice_SLQSVoiceSetConfig_t, 455
pPrefVoicePrivacy
pack_voice_SLQSVoiceGetConfig_t, 449
pPrefVoiceSO
pack_voice_SLQSVoiceGetConfig_t, 449
pack_voice_SLQSVoiceSetConfig_t, 455
pPrefVoiceSOStatus
unpack_voice_SLQSVoiceSetConfig_t, 869
pPreference
unpack_pds_GetXTRANetwork_t, 746
pPriCSCFPort
pack_ims_SLQSSetRegMgrConfig_t, 327
pPriCSCFPortName
unpack_ims_SLQSGetRegMgrConfig_t, 624
pPriCSCFPortNameLen
unpack_ims_SLQSGetRegMgrConfig_t, 624
pPriDNSIPv4AddPref
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108

pPriDNSIPv6addpref
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPriV6DnsAddress
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pPrimaryHA
pack_wds_SetMobileIPProfile_t, 469
pPrimaryID
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pPrimaryV4DnsAddress
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pProfileID
unpack_wds_SLQSCreateProfile_t, 894
pProfileId
pack_wds_SLQSCreateProfile_t, 470
pack_wds_SLQSModifyProfile_t, 475
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 478
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 479
pProfileList
pack_wds_SLQSSet3GPPConfigItem_t, 476
pProfileSettings
unpack_wds_SLQSGetProfileSettings_t, 903
pProfileType
pack_wds_SLQSCreateProfile_t, 470
pack_wds_SLQSModifyProfile_t, 475
pProfilename
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pProfilenameSize
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pProtoSubTypElmnt
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 731
pProtocol
pack_nas_SetCDMANetworkParameters_t, 348
pProtocolSubtypeElement
nas_HDRPersonality_Ind_Data, 179
unpack_nas_SLQSSwiGetHDRPersonality_t, 730
pPsmEnableState
pack_dms_SLQSSetPowerSaveModeConfig_t, 312
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pQosClassID
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pRAT
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 382
pRATINstance
unpack_nas_PerformNetworkScan_t, 686
pRATInstanceSize
unpack_nas_PerformNetworkScan_t, 686
pRATType
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pRFBandInfoParam
nas_RfBandInfoList, 240
PRIString
unpack_dms_GetFirmwareRevision_t, 568
unpack_dms_GetFirmwareRevisions_t, 569
PRLEnable
pack_swioma_SLQSOMADMSetSettingsExt_t, 425
PRLInd
unpack_nas_SLQSGetServingSystem_t, 702
PRLPTlv
NASQmiCbkNasSystemSelPrefInd, 291
PRLPref
nas_PRLPrefTlv, 231
NASPRLPreferenceTlv, 289
pRSRPThresList
nas_RSRPThresh, 246
pRSRPThresh
pack_nas_SLQSConfigSigInfo_t, 351
pRSRQThresList
nas_RSRQThresh, 247
pRSRQThresh
pack_nas_SLQSConfigSigInfo_t, 351
pRSSIThresList
nas_RSSIThresh, 248
pRSSIThresh
pack_nas_SLQSConfigSigInfo_t, 351
pRTPRTCPInactTimer
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pRXAGCList
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pRXAIG
audio_RXAGCList, 38
pRXAVCAGCSwitch
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pRXAVCList
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pRXChain0Info
unpack_nas_SLQSNasGetTxRxInfo_t, 723
pRXChain1Info
unpack_nas_SLQSNasGetTxRxInfo_t, 723
pRXComprSlope
audio_RXAGCList, 39
pRXComprThres
audio_RXAGCList, 39
pRXDroppedCount
unpack_wds_GetPacketStatistics_t, 890
pRXExpSlope
audio_RXAGCList, 39
pRXExpThres
audio_RXAGCList, 39
pRXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 890

pRXOkBytesCount
unpack_wds_GetPacketStatistics_t, 890
pRXPCMIIRFltr
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pRXPacketErrors
unpack_wds_GetPacketStatistics_t, 890
pRXPacketOverflows
unpack_wds_GetPacketStatistics_t, 890
pRXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 890
pRXStaticGain
audio_RXAGCList, 39
pRXTotalBytes
unpack_wds_GetByteTotals_t, 881
pRadioInterface
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 724
pRandomizationWindow
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
pRankIndicatorInd
pack_nas_SLQSNasSwiIndicationRegister_t, 363
pRatDisabledMask
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pReRegPeriod
pack_wds_SetMobileIPParameters_t, 467
pReRegTraffic
pack_wds_SetMobileIPParameters_t, 467
pReadResult
unpack_uim_ReadTransparent_t, 828
pReason
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pReasonMask
unpack_dms_GetOfflineReason_t, 574
pRecurrenceType
pack_loc_Start_t, 345
pRedirNumInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 879
pRefreshEvent
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 834
pRegCallStatInfoEvt
sms_getIndicationReg, 499
sms_setIndicationReg, 509
pRegDTMFEvents
pack_voice_SLQSVoiceIndicationRegister_t, 450
pRegInd
sms_getTransLayerInfo, 500
pRegMgrConfigEvents
pack_ims_SLQSImsConfigIndicationRegister_t, 322
pRegStatus
sms_getTransNWRegInfo, 501
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
pRegTransLayerInfoEvt
sms_getIndicationReg, 499
sms_setIndicationReg, 509
pRegTransNWRegInfoEvt
sms_getIndicationReg, 499
sms_setIndicationReg, 509
pRegVoicePrivacyEvents
pack_voice_SLQSVoiceIndicationRegister_t, 450
pRejectReason
pack_swiavms_SLQSAVMSSendSelection_t, 413
pack_swioma_SLQSOMADMSendSelection_t, 422
pRemainingRetries
unpack_uim_ChangePin_t, 827
unpack_uim_SetPinProtection_t, 829
unpack_uim_SLQSUIMDepersonalization_t, 831
unpack_uim_UnblockPin_t, 836
unpack_uim_VerifyPin_t, 836
pRemotePartyName
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pRemotePartyNum
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pReportChannelRate
pack_wds_SLQSGetDUNCallInfo_t, 472
pReportConnStatus
pack_wds_SLQSGetDUNCallInfo_t, 472
pReportDataBearerTech
pack_wds_SLQSGetDUNCallInfo_t, 472
pReportDormStatus
pack_wds_SLQSGetDUNCallInfo_t, 472
pReqSettings
pack_wds_SLQSGetRuntimeSettings_t, 474
pRequestOptionList
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 479
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 915
pRequestedTag
pack_sms_SLQSGetSMSList_t, 402
pRetryInterval
pack_wds_SetMobileIPParameters_t, 467
pRetryLimit
pack_wds_SetMobileIPParameters_t, 467
pRetryMessage
sms_sendAsyncsmsParams, 508
pRetryMessageId
sms_sendAsyncsmsParams, 508
pRevInUse
nas_CDMASysInfo, 153
pRevInUseValid
nas_CDMASysInfo, 153
pRevTunneling
pack_wds_SetMobileIPProfile_t, 469
pRfBandInfoExtFormat
unpack_nas_SLQSNasGetRFInfo_t, 721
pRfBandInfoExtFormatParam
nas_RfBandInfoExtFormat, 239
pRfBandwidthInfo
unpack_nas_SLQSNasGetRFInfo_t, 721
pRfBandwidthInfoParam

nas_RfBandwidthInfo, 241
pRfDedicatedBandInfo
unpack_nas_SLQSNasGetRFInfo_t, 721
pRfDedicatedBandInfoParam
nas_RfDedicatedBandInfo, 242
pRingBackTimer
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pRingingTimer
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pRoamPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 382
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pRoamTimer
pack_voice_SLQSVoiceGetConfig_t, 449
pRoamTimerCnt
unpack_voice_SLQSVoiceGetConfig_t, 866
pRoamTimerConfig
pack_voice_SLQSVoiceSetConfig_t, 455
pRoamTimerStatus
unpack_voice_SLQSVoiceSetConfig_t, 870
pRoaming
pack_nas_SetCDMANetworkParameters_t, 348
pRscp
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
PSDomain
unpack_nas_GetServingNetwork_t, 683
pSID
unpack_nas_GetHomeNetwork3GPP2_t, 677
pSIODataPort
pack_qos_BindDataPort_t, 395
pSIPConfigEvents
pack_ims_SLQSImsConfigIndicationRegister_t, 322
pSIPLocalPort
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626
PSMCfgChangeInfo
pack_dms_SetIndicationRegister_t, 309
pSMSAttemptedFlag
packgetDyingGaspStatistics, 485
pSMSCAddress
pack_sms_SetSMSCAddress_t, 399
unpack_sms_GetSMSCAddress_t, 782
pSMSCType
pack_sms_SetSMSCAddress_t, 399
unpack_sms_GetSMSCAddress_t, 782
pSMSConfigEvents
pack_ims_SLQSImsConfigIndicationRegister_t, 322
pSMSFormat
pack_ims_SLQSSetIMSSMSConfig_t, 323
unpack_ims_SLQSGetIMSSMSConfig_t, 620
pSMSOverIPNwInd
pack_ims_SLQSSetIMSSMSConfig_t, 323
unpack_ims_SLQSGetIMSSMSConfig_t, 620
PSMStatus
pack_dms_SetIndicationRegister_t, 309
pSPC
pack_nas_SetCDMANetworkParameters_t, 348
pack_wds_SetMobileIPParameters_t, 467
pSV
loc_BdsSVInfo, 123
loc_SVInfo, 137
pSVInfo
pack_loc_Delete_Assist_Data_t, 330
pSarRFState
unpack_sar_SLQSGetRfSarState_t, 781
pSatelliteInfo
unpack_loc_GnssSvInfo_Ind_t, 655
pScanResult
unpack_nas_PerformNetworkScan_t, 686
pSccRxInfo
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 733
pScrAmrEnable
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pScrAmrWbEnable
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pSecDNSIPv4AddPref
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pSecDNSIPv6addpref
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 108
pSecV6DnsAddress
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pSecondaryFlag
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pSecondaryHA
pack_wds_SetMobileIPProfile_t, 469
pSecondaryV4DnsAddress
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pSectorID
nas_NetworkStatEVDO, 217
pSendSmsParams
pack_sms_SLQSSendAsyncSMS_t, 403
pSensorDataUsage
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pServerAddress
pack_pds_SLQSSetAGPSConfig_t, 391
unpack_pds_SLQSGetAGPSConfig_t, 751
pServerPort
pack_pds_SLQSSetAGPSConfig_t, 391
unpack_pds_SLQSGetAGPSConfig_t, 751
pServerURL
pack_pds_SLQSSetAGPSConfig_t, 391
unpack_pds_SLQSGetAGPSConfig_t, 751

pServerURLLength
pack_pds_SLQSSetAGPSConfig_t, 391
unpack_pds_SLQSGetAGPSConfig_t, 751
pServiceClass
pack_voice_SLQSVoiceSetSUPSService_t, 457
pServiceOption
sms_sendAsyncsmsParams, 508
pServingSystemInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pSessionExpiryTimer
pack_ims_SLQSSetIMSVoIPConfig_t, 326
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
pSessionType
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
pSetIndicationRegReq
pack_sms_SLQSSetIndicationRegister_t, 404
pSetRoutesReq
pack_sms_SLQSSmsSetRoutes_t, 406
pSettingResp
unpack_ims_SLQSGetIMSSMSConfig_t, 620
unpack_ims_SLQSGetIMSUserConfig_t, 620
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
unpack_ims_SLQSGetRegMgrConfig_t, 624
unpack_ims_SLQSGetSIPConfig_t, 626
unpack_ims_SLQSSetIMSSMSConfig_t, 627
unpack_ims_SLQSSetIMSUserConfig_t, 628
unpack_ims_SLQSSetIMSVoIPConfig_t, 628
unpack_ims_SLQSSetRegMgrConfig_t, 629
unpack_ims_SLQSSetSIPConfig_t, 629
pSigCompEnabled
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626
pSigIndReq
pack_nas_SLQSSetSignalStrengthsCallback_t, 366
pSignalInfo
unpack_voice_VoiceInfoRecCallback_ind_t, 879
pSignalStrengthInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pSmsOnIms
sms_sendAsyncsmsParams, 508
pSmsStorage
unpack_sms_SLQSSwiGetSMSStorage_t, 796
pSourceIP
LibPackTFTIDParams, 117
pSpeedHorizontal
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pSpeedUnc
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pSpeedVertical
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pSpeedVerticalUnc
unpack_loc_BestAvailPos_Ind_t, 647
pSrvDomainPref
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 382
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSGetSysSelectionPrefExt_t, 716
pSrvOpt
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pSrvRegRestriction
pack_nas_SLQSSetSysSelectionPref_t, 371
pack_nas_SLQSSetSysSelectionPrefExt_t, 382
pSrvcProviderName
unpack_nas_SLQSGetOperatorNameData_t, 694
pStage0Val
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pStage1Val
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pStage2Val
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pStage3Val
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pStage4Val
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pStageCnt
audio_RXPCMIIRFltr, 41
audio_TXPCMIIRFltr, 43
pStatMask
pack_wds_GetPacketStatistics_t, 461
pSubscribeTimer
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626
pSubscriptionInfoInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pSupportEmergencyCalls
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pSuppressSysInfoInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pSuppsNotifEvents
pack_voice_SLQSVoiceIndicationRegister_t, 450
pSvUsedforFix
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pSvcClass
pack_voice_SLQSVoiceGetCallBarring_t, 446
pack_voice_SLQSVoiceGetCallForwardingStatus-_t, 447
pack_voice_SLQSVoiceGetCallWaiting_t, 448
unpack_voice_SLQSVoiceGetCallBarring_t, 851
unpack_voice_SLQSVoiceGetCallWaiting_t, 857
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pSvcType
pack_voice_SLQSVoiceDialCall_t, 445
pSysInfoInd

pack_nas_SLQSNasIndicationRegisterExt_t, 361
pSysInfoNoChange
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pSystemSelectionInd
pack_nas_SLQSNasIndicationRegisterExt_t, 361
pTDSCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMAECIOThreshList
nas_TDSCDMAECIOThresh, 263
pTDSCDMARSCPDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMARSCPThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMARSCPThreshList
nas_TDSCDMARSCPThresh, 263
pTDSCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMARSSIThreshList
nas_TDSCDMARSSIThresh, 264
pTDSCDMASINRDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMASINRThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pTDSCDMASINRThreshList
nas_TDSCDMASINRThresh, 265
pTDSCDMASigInfoExt
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
pTFTID1Params
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pTFTID2Params
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pTTYConfigStatus
unpack_voice_SLQSVoiceSetConfig_t, 870
pTTYMode
pack_voice_SLQSVoiceGetConfig_t, 449
pack_voice_SLQSVoiceSetConfig_t, 455
pTXAGCList
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pTXAIG
audio_TXAGCList, 41
pTXAVCSwitch
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pTXComprSlope
audio_TXAGCList, 42
pTXComprThres
audio_TXAGCList, 42
pTXDroppedCount
unpack_wds_GetPacketStatistics_t, 890
pTXExpSlope
audio_TXAGCList, 42
pTXExpThres
audio_TXAGCList, 42
pTXGain
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pTXInfo
unpack_nas_SLQSNasGetTxRxInfo_t, 723
pTXOKBytesLastCall
unpack_wds_GetPacketStatistics_t, 890
pTXOkBytesCount
unpack_wds_GetPacketStatistics_t, 890
pTXPCMIIRFltr
pack_audio_SLQSSetAudioPathConfig_t, 298
unpack_audio_SLQSGetAudioPathConfig_t, 548
pTXPacketErrors
unpack_wds_GetPacketStatistics_t, 890
pTXPacketOverflows
unpack_wds_GetPacketStatistics_t, 890
pTXPacketSuccesses
unpack_wds_GetPacketStatistics_t, 890
pTXStaticGain
audio_TXAGCList, 42
pTXTotalBytes
unpack_wds_GetByteTotals_t, 881
pTdscdmaBandPref
pack_nas_SLQSSetSysSelectionPref_t, 372
pack_nas_SLQSSetSysSelectionPrefExt_t, 382
pTech
pack_wds_SLQSStartDataSession_t, 480
pTechnologyMask
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pTempReport
pack_dms_SwiSetEventReport_t, 315
pTimeSrc
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pTimeStamp
pack_pds_SLQSPDSInjectPositionData_t, 390
packgetDyingGaspStatistics, 485
pTimeType
pack_pds_SLQSPDSInjectPositionData_t, 390
pTimeUnc
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pTimeZone
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 724
pTimeout
unpack_pds_GetPDSDefaults_t, 743
pTimer
pack_nas_SLQSNasSwiIndicationRegister_t, 363
pTimerSIPReg
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626
pTimerT1
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626

pTimerT2
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626
pTimerTf
pack_ims_SLQSSetSIPConfig_t, 328
unpack_ims_SLQSGetSIPConfig_t, 626
pTimerVal
pack_voice_SLQSVoiceSetSUPSService_t, 457
pTimestampUtc
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pTrackingStatus
unpack_pds_GetPDSState_t, 744
pTransLayerInfo
sms_getTransLayerInfo, 500
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 797
pTransferStatInd
pack_wds_SLQSGetDUNCallInfo_t, 472
pack_wds_SLQSWdsSetEventReport_t, 483
pTransferStatusReport
sms_setRoutesReq, 510
pTrueIMSI
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
pUATI
unpack_nas_SLQSSwiGetHRPDStats_t, 731
pUMTSCellID
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
pUMTSInfo
unpack_nas_SLQSNasGetCellLocationInfo_t, 719
pUMTSMinQoS
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pUMTSMinQosSigInd
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pUMTSReqQoS
LibpackProfile3GPP, 97
LibPackprofile_3GPP, 109
pUMTSReqQoSSigInd
LibpackProfile3GPP, 98
LibPackprofile_3GPP, 109
pURL
unpack_loc_GetServer_Ind_t, 654
pURLAddr
pack_loc_SLQSLOCSetServer_t, 344
pUSSDInfo
unpack_voice_SLQSOriginateUSSD_t, 844
pUSSInfo
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
pUUSInFo
pack_voice_SLQSVoiceDialCall_t, 445
pUUSInfo
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pUimSlotsStatus
unpack_uim_SLQSUIMGetSlotsStatus_t, 834
pUname
PackSwiAVMSSettingsAPNInfo, 488
pUsbMTUSize
unpack_swidms_SLQSSwiDmsGetMTU_t, 815
pUser
pack_wds_SLQSStartDataSession_t, 480
pUserConfigEvents
pack_ims_SLQSImsConfigIndicationRegister_t, 322
pUserData
sms_sendAsyncsmsParams, 508
pUserId
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pUserIdSize
LibpackProfile3GPP2, 103
LibPackprofile_3GPP2, 114
pUsername
LibpackProfile3GPP, 98
LibPackprofile_3GPP, 109
pack_wds_SetDefaultProfile_t, 465
pUsernameSize
LibpackProfile3GPP, 98
LibPackprofile_3GPP, 109
pVerboseFailReasonType
unpack_wds_SLQSStartDataSession_t, 916
pVerboseFailureReason
unpack_wds_SLQSStartDataSession_t, 916
pVertConfidence
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_PositionRpt_Ind_t, 665
pVertReliability
unpack_loc_BestAvailPos_Ind_t, 648
unpack_loc_PositionRpt_Ind_t, 665
pVertUnc
unpack_loc_BestAvailPos_Ind_t, 648
unpack_loc_PositionRpt_Ind_t, 665
pVerticalConfidence
pack_pds_SLQSPDSInjectPositionData_t, 390
pVerticalUnc
pack_pds_SLQSPDSInjectPositionData_t, 390
pVoIPConfigEvents
pack_ims_SLQSImsConfigIndicationRegister_t, 322
pVoiceDomainPref
pack_voice_SLQSVoiceGetConfig_t, 449
pVoiceDomainPrefStatus
unpack_voice_SLQSVoiceSetConfig_t, 870
pVoicePrivacy
unpack_voice_SLQSVoiceGetAllCallInfo_t, 849
unpack_voice_SLQSVoiceGetCallInfo_t, 855
pVoltReport
pack_dms_SwiSetEventReport_t, 315
pVolume
pack_swiaudio_SLQSSetM2MAudioProfile_t, 410
pWAMSParaChanged
unpack_swiavms_SLQSAVMSEventReportInd_t, 805

pWCDMABER
unpack_nas_SLQSGetErrorRate_t, 692
pWCDMACallBarringSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pWCDMACipherDomainSysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pWCDMAECIODelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pWCDMAECIOThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pWCDMAECIOThreshList
nas_WCDMAECIOThresh, 274
pWCDMAInfoLTENeighborCell
unpack_nas_SLQSNasGetCellLocationInfo_t, 719
pWCDMARSSIDelta
pack_nas_SLQSNasConfigSigInfo2_t, 357
pWCDMARSSIThresh
pack_nas_SLQSNasConfigSigInfo2_t, 357
pWCDMARSSIThreshList
nas_WCDMARSSIThresh, 276
pWCDMASigInfo
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
pWCDMASrvStatusInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pWCDMASysInfo
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSSysInfoCallback_ind_t, 738
pWcdmaUARFCN
pack_nas_SLQSNASSwiSetChannelLock_t, 364
unpack_nas_SLQSNASSwiGetChannelLock_t, 726
pWifiState
pack_pds_SLQSSetPositionMethodState_t, 392
pXid
unpack_loc_BestAvailPos_Ind_t, 648
pXtraDataState
pack_pds_SLQSSetPositionMethodState_t, 392
pXtraTimeState
pack_pds_SLQSSetPositionMethodState_t, 392
pack_audio_SLQSGetAudioPathConfig
audio.h, 1002
pack_audio_SLQSGetAudioPathConfig_t, 294
Item, 295
Profile, 295
pack_audio_SLQSGetAudioProfile
audio.h, 1002
pack_audio_SLQSGetAudioProfile_t, 295
Generator, 295
pack_audio_SLQSGetAudioVolTLBConfig
audio.h, 1003
pack_audio_SLQSGetAudioVolTLBConfig_t, 296
Generator, 296
Item, 296
Profile, 296
Volume, 296
pack_audio_SLQSSetAudioPathConfig
audio.h, 1003
pack_audio_SLQSSetAudioPathConfig_t, 296
pCodecSTGain, 298
pECMode, 298
pNSEnable, 298
pRXAGCList, 298
pRXAVCList, 298
pTXAGCList, 298
pTXAVCSwitch, 298
pTXGain, 298
Profile, 298
pack_audio_SLQSSetAudioProfile
audio.h, 1004
pack_audio_SLQSSetAudioProfile_t, 298
EarMute, 299
Generator, 299
MicMute, 299
Profile, 299
Volume, 299
pack_audio_SLQSSetAudioVolTLBConfig
audio.h, 1004
pack_audio_SLQSSetAudioVolTLBConfig_t, 300
Generator, 300
Item, 300
Profile, 300
VolValue, 300
Volume, 300
pack_cat_CATSendEnvelopeCommand
cat.h, 1008
pack_cat_CATSendEnvelopeCommand_t, 300
cmdID, 301
dataLen, 301
pData, 301
pack_cat_CATSendTerminalResponse
cat.h, 1008
pack_cat_CATSendTerminalResponse_t, 301
dataLen, 302
pData, 302
refID, 302
pack_cat_SetCATEventCallback
cat.h, 1008
pack_cat_SetCATEventCallback_t, 302
eventMask, 302
pack_dms_ActivateAutomatic
dms.h, 1025
pack_dms_ActivateAutomatic_t, 303
actCode, 303
pack_dms_GetActivationState
dms.h, 1025
pack_dms_GetBandCapability
dms.h, 1025
pack_dms_GetCrashAction
dms.h, 1026
pack_dms_GetCustFeature
dms.h, 1026
pack_dms_GetCustFeaturesV2
dms.h, 1026

pack_dms_GetCustFeaturesV2_t, 303
cust_id, 303
list_type, 303
Tlvresult, 303
pack_dms_GetDeviceCap
dms.h, 1027
pack_dms_GetDeviceCapabilities
dms.h, 1027
pack_dms_GetDeviceHardwareRev
dms.h, 1028
pack_dms_GetDeviceMfr
dms.h, 1028
pack_dms_GetDeviceSerialNumbers
dms.h, 1028
pack_dms_GetFSN
dms.h, 1030
pack_dms_GetFirmwareInfo
dms.h, 1029
pack_dms_GetFirmwareRevision
dms.h, 1029
pack_dms_GetFirmwareRevisions
dms.h, 1029
pack_dms_GetHardwareRevision
dms.h, 1030
pack_dms_GetIMSI
dms.h, 1030
pack_dms_GetManufacturer
dms.h, 1031
pack_dms_GetModelID
dms.h, 1031
pack_dms_GetNetworkTime
dms.h, 1031
pack_dms_GetOfflineReason
dms.h, 1032
pack_dms_GetPRLVersion
dms.h, 1032
pack_dms_GetPower
dms.h, 1032
pack_dms_GetSerialNumbers
dms.h, 1033
pack_dms_GetUSBComp
dms.h, 1033
pack_dms_GetVoiceNumber
dms.h, 1033
pack_dms_ResetToFactoryDefaults
dms.h, 1034
pack_dms_ResetToFactoryDefaults_t, 304
spc, 304
pack_dms_SLQSDmsSwiGetPCInfo
dms.h, 1038
pack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1038
pack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1038
pack_dms_SLQSDmsSwiIndicationRegister_t, 311
resetInfoInd, 311
pack_dms_SLQSGetBandCapability
dms.h, 1039
pack_dms_SLQSGetERIFile
dms.h, 1039
pack_dms_SLQSGetPowerSaveModeConfig
dms.h, 1039
pack_dms_SLQSSetPowerSaveModeConfig
dms.h, 1040
pack_dms_SLQSSetPowerSaveModeConfig_t, 311
pActiveTimer, 312
pPeriodicUpdateTimer, 312
pPsmEnableState, 312
pack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1040
pack_dms_SLQSSwiGetCrashInfo
dms.h, 1040
pack_dms_SLQSSwiGetCrashInfo_t, 312
clear, 312
pack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1041
pack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1041
pack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1041
pack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1042
pack_dms_SLQSSwiGetHostDevInfo
dms.h, 1042
pack_dms_SLQSSwiGetOSInfo
dms.h, 1042
pack_dms_SLQSSwiGetSerialNoExt
dms.h, 1043
pack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1043
pack_dms_SLQSSwiSetDyingGaspCfg_t, 312
pDestSMSContent, 313
pDestSMSNum, 313
pack_dms_SLQSSwiSetHostDevInfo
dms.h, 1043
pack_dms_SLQSSwiSetHostDevInfo_t, 313
manString, 313
modelString, 313
plasmaIDString, 314
swVerString, 314
pack_dms_SLQSSwiSetOSInfo
dms.h, 1044
pack_dms_SLQSSwiSetOSInfo_t, 314
nameString, 314
versionString, 314
pack_dms_SLQSUIMGetState
dms.h, 1044
pack_dms_SetActivationStatusCallback
dms.h, 1034
pack_dms_SetActivationStatusCallback_t, 304
activationState, 304
pack_dms_SetCrashAction
dms.h, 1035
pack_dms_SetCrashAction_t, 304
crashAction, 305
pack_dms_SetCustFeature

dms.h, 1035
pack_dms_SetCustFeature_t, 305
DHCPRelayEnabled, 307
DisableIMSI, 307
GPSLPM, 307
GPSSel, 307
GpsEnable, 307
IPFamSupport, 307
IsVoiceEnabled, 307
RMAutoConnect, 307
SMSSupport, 307
pack_dms_SetCustFeaturesV2
dms.h, 1035
pack_dms_SetCustFeaturesV2_t, 307
cust_id, 308
cust_value, 308
Tlvresult, 308
value_length, 308
pack_dms_SetEventReport
dms.h, 1036
pack_dms_SetEventReport_t, 308
mode, 308
pack_dms_SetFirmwarePreference
dms.h, 1036
pack_dms_SetIndicationRegister
dms.h, 1036
pack_dms_SetIndicationRegister_t, 308
PSMCfgChangeInfo, 309
PSMStatus, 309
RptIMSCapability, 309
pack_dms_SetPower
dms.h, 1037
pack_dms_SetPower_t, 309
mode, 310
Tlvresult, 310
pack_dms_SetUSBComp
dms.h, 1037
pack_dms_SetUSBComp_t, 310
Tlvresult, 310
USBComp, 310
pack_dms_SwiSetEventReport
dms.h, 1045
pack_dms_SwiSetEventReport_t, 314
pTempReport, 315
pVoltReport, 315
pack_dms_UIMChangePIN
dms.h, 1045
pack_dms_UIMChangePIN_t, 315
id, 315
newValue, 315
oldValue, 315
pack_dms_UIMGetControlKeyStatus
dms.h, 1045
pack_dms_UIMGetControlKeyStatus_t, 315
facility, 316
pack_dms_UIMGetICCID
dms.h, 1046
pack_dms_UIMGetICCID_t, 316
ParamPresenceMask, 316
Tlvresult, 316
pack_dms_UIMGetPINStatus
dms.h, 1046
pack_dms_UIMSetControlKeyProtection
dms.h, 1046
pack_dms_UIMSetControlKeyProtection_t, 316
facility, 317
facilityCk, 317
facilityState, 317
pack_dms_UIMSetPINProtection
dms.h, 1047
pack_dms_UIMSetPINProtection_t, 317
bEnable, 318
id, 318
value, 318
pack_dms_UIMUnblockControlKey
dms.h, 1047
pack_dms_UIMUnblockControlKey_t, 318
facility, 318
facilityCk, 318
pack_dms_UIMUnblockPIN
dms.h, 1048
pack_dms_UIMUnblockPIN_t, 318
id, 319
newPin, 319
pukValue, 319
pack_dms_UIMVerifyPIN
dms.h, 1048
pack_dms_UIMVerifyPIN_t, 319
id, 319
value, 319
pack_dms_ValidateSPC
dms.h, 1048
pack_fms_GetImagesPreference
fms.h, 1075
pack_fms_GetImagesPreference_t, 320
Tlvresult, 320
pack_fms_GetStoredImages
fms.h, 1076
pack_fms_GetStoredImages_t, 320
Tlvresult, 320
pack_fms_SetImagesPreference
fms.h, 1076
pack_fms_SetImagesPreference_t, 320
bForceDownload, 321
imageListSize, 321
modemindex, 321
pImageList, 321
Tlvresult, 321
pack_ims_SLQSGetIMSSMSConfig
ims.h, 1079
pack_ims_SLQSGetIMSUserConfig
ims.h, 1079
pack_ims_SLQSGetIMSVoIPConfig
ims.h, 1080
pack_ims_SLQSGetRegMgrConfig
ims.h, 1080

pack_ims_SLQSGetSIPConfig
ims.h, 1080
pack_ims_SLQSImsConfigIndicationRegister
ims.h, 1081
pack_ims_SLQSImsConfigIndicationRegister_t, 321
pRegMgrConfigEvents, 322
pSIPConfigEvents, 322
pSMSConfigEvents, 322
pUserConfigEvents, 322
pVoIPConfigEvents, 322
pack_ims_SLQSSetIMSSMSConfig
ims.h, 1081
pack_ims_SLQSSetIMSSMSConfig_t, 323
pack_ims_SLQSSetIMSUserConfig
ims.h, 1082
pack_ims_SLQSSetIMSUserConfig_t, 323
pIMSDomain, 324
pack_ims_SLQSSetIMSVoIPConfig
ims.h, 1082
pack_ims_SLQSSetIMSVoIPConfig_t, 324
pAmrMode, 326
pRingingTimer, 326
pack_ims_SLQSSetRegMgrConfig
ims.h, 1083
pack_ims_SLQSSetRegMgrConfig_t, 326
pCSCFPortName, 327
pIMSTestMode, 327
pPriCSCFPort, 327
pack_ims_SLQSSetSIPConfig
ims.h, 1083
pack_ims_SLQSSetSIPConfig_t, 327
pSIPLocalPort, 328
pSigCompEnabled, 328
pSubscribeTimer, 328
pTimerSIPReg, 328
pTimerT1, 328
pTimerT2, 328
pTimerTf, 328
pack_imsa_SLQSGetIMSARegStatus
imsa.h, 1090
pack_imsa_SLQSGetIMSAServiceStatus
imsa.h, 1091
pack_imsa_SLQSRegisterIMSAIndication
imsa.h, 1091
pack_imsa_SLQSRegisterIMSAIndication_t, 328
PdpStatusConfig, 329
RatHandoverStatusConfig, 329
RegStatusConfig, 329
ServiceStatusConfig, 329
pack_loc_Delete_Assist_Data_t, 329
pBdsSVInfo, 330
pCellDb, 330
pClkInfo, 330
pGnssData, 330
pSVInfo, 330
Tlvresult, 330
pack_loc_DeleteAssistData
loc.h, 1111
pack_loc_EventRegister
loc.h, 1112
pack_loc_EventRegister_t, 330
eventRegister, 333
Tlvresult, 333
pack_loc_SLQSLOCGetBestAvailPos
loc.h, 1113
pack_loc_SLQSLOCGetBestAvailPos_t, 334
Tlvresult, 334
xid, 334
pack_loc_SLQSLOCGetServer
loc.h, 1113
pack_loc_SLQSLOCGetServer_t, 335
has_serverAddrTypeMask, 335
serverAddrTypeMask, 335
serverType, 335
pack_loc_SLQSLOCInjectPosition
loc.h, 1114
pack_loc_SLQSLOCInjectPosition_t, 335
altitudeSrcInfo, 339
altitudeWrtEllipsoid, 339
altitudeWrtMeanSeaLevel, 339
has_altitudeSrcInfo, 339
has_altitudeWrtEllipsoid, 339
has_horConfidence, 339
has_horReliability, 339
has_horUncCircular, 339
has_latitude, 339
has_longitude, 339
has_positionSrc, 339
has_rawHorConfidence, 339
has_rawHorUncCircular, 339
has_timestampAge, 339
has_timestampUtc, 339
has_vertConfidence, 339
has_vertRelicability, 339
has_vertUnc, 340
horConfidence, 340
horReliability, 340
horUncCircular, 340
latitude, 340
longitude, 340
positionSrc, 340
rawHorConfidence, 340
rawHorUncCircular, 340
timestampAge, 340
timestampUtc, 340
vertConfidence, 340
vertReliability, 340
vertUnc, 340
pack_loc_SLQSLOCInjectSensorData
loc.h, 1114
pack_loc_SLQSLOCInjectSensorData_t, 340
accelTemp, 341
acceleroData, 341
acceleroTimeSrc, 341
gyroData, 341
gyroTemp, 341

gyroTimeSrc, 342
has_accelTemp, 342
has_acceleroTimeSrc, 342
has_accleroData, 342
has_gyroData, 342
has_gyroTemp, 342
has_gyroTimeSrc, 342
has_opaqueId, 342
opaqueId, 342
pack_loc_SLQSLOCInjectUTCTime
loc.h, 1114
pack_loc_SLQSLOCInjectUTCTime_t, 342
timeMsec, 342
timeUncMsec, 342
pack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1115
pack_loc_SLQSLOCSetCradleMountConfig_t, 342
confidence, 343
has_confidence, 343
state, 343
pack_loc_SLQSLOCSetServer
loc.h, 1115
pack_loc_SLQSLOCSetServer_t, 343
pIPv4Config, 344
pIPv6Config, 344
pURLAddr, 344
serverType, 344
pack_loc_SetExtPowerState
loc.h, 1112
pack_loc_SetExtPowerState_t, 333
extPowerState, 333
Tlvresult, 333
pack_loc_SetOperationMode
loc.h, 1112
pack_loc_SetOperationMode_t, 333
mode, 334
Tlvresult, 334
pack_loc_Start
loc.h, 1116
pack_loc_Start_t, 344
pApplicationInfo, 345
pConfigAltitudeAssumed, 345
pHorizontalAccuracyLvl, 345
pIntermediateReportState, 345
pMinIntervalTime, 345
pRecurrenceType, 345
SessionId, 346
Tlvresult, 346
pack_loc_Stop
loc.h, 1116
pack_loc_Stop_t, 346
SessionId, 346
Tlvresult, 346
pack_nas_GetACCOLC
nas.h, 1138
pack_nas_GetANAAAAuthenticationStatus
nas.h, 1138
pack_nas_GetCDMANetworkParameters
nas.h, 1139
pack_nas_GetHomeNetwork
nas.h, 1139
pack_nas_GetHomeNetwork3GPP2
nas.h, 1139
pack_nas_GetNetworkPreference
nas.h, 1140
pack_nas_GetRFInfo
nas.h, 1140
pack_nas_GetServingNetwork
nas.h, 1140
pack_nas_GetServingNetworkCapabilities
nas.h, 1140
pack_nas_GetSignalStrengths
nas.h, 1141
pack_nas_InitiateDomainAttach
nas.h, 1141
pack_nas_InitiateDomainAttach_t, 346
action, 346
pack_nas_PerformNetworkScan
nas.h, 1142
pack_nas_SLQSConfigSigInfo
nas.h, 1144
pack_nas_SLQSConfigSigInfo_t, 349
pECIOThresh, 351
pIOThresh, 351
pLTESNRThresh, 351
pLTESigRptCfg, 351
pRSRPThresh, 351
pRSRQThresh, 351
pRSSIThresh, 351
pack_nas_SLQSGetErrorRate
nas.h, 1144
pack_nas_SLQSGetNetworkTime
nas.h, 1145
pack_nas_SLQSGetOperatorNameData
nas.h, 1145
pack_nas_SLQSGetPLMNName
nas.h, 1145
pack_nas_SLQSGetPLMNName_t, 351
mcc, 352
mnc, 352
pMncPcsStatus, 352
pack_nas_SLQSGetServingSystem
nas.h, 1145
pack_nas_SLQSGetSignalStrength
nas.h, 1146
pack_nas_SLQSGetSysInfo
nas.h, 1146
pack_nas_SLQSGetSysSelectionPref
nas.h, 1147
pack_nas_SLQSGetSysSelectionPrefExt
nas.h, 1147
pack_nas_SLQSInitiateNetworkRegistration
nas.h, 1147
pack_nas_SLQSInitiateNetworkRegistration_t, 352
pChangeDuration, 352
pMNRInfo, 352

pMncPcsDigitStatus, 352
regAction, 353
pack_nas_SLQSNASGeteDRXParams
nas.h, 1149
pack_nas_SLQSNASSeteDRXParams
nas.h, 1151
pack_nas_SLQSNASSeteDRXParams_t, 361
pCycleLen, 362
pEdrxEnable, 362
pack_nas_SLQSNASSwiGetChannelLock
nas.h, 1151
pack_nas_SLQSNASSwiSetChannelLock
nas.h, 1152
pack_nas_SLQSNASSwiSetChannelLock_t, 363
pack_nas_SLQSNasConfigSigInfo2
nas.h, 1148
pack_nas_SLQSNasConfigSigInfo2_t, 353
pHDRIODelta, 356
pHDRIOThresh, 356
pLTESigRptConfig, 357
pack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1148
pack_nas_SLQSNasGet3GPP2Subscription_t, 357
namID, 358
pack_nas_SLQSNasGetCellLocationInfo
nas.h, 1148
pack_nas_SLQSNasGetHDRColorCode
nas.h, 1149
pack_nas_SLQSNasGetRFInfo
nas.h, 1149
pack_nas_SLQSNasGetSigInfo
nas.h, 1150
pack_nas_SLQSNasGetTxRxInfo
nas.h, 1150
pack_nas_SLQSNasGetTxRxInfo_t, 358
radio_if, 358
pack_nas_SLQSNasIndicationRegisterExt
nas.h, 1150
pack_nas_SLQSNasIndicationRegisterExt_t, 358
pDDTMInd, 360
pDualStandByPrefInd, 360
pErrorRateInd, 361
pHDRSessionCloseInd, 361
pLTECphyCa, 361
pManagedRoamingInd, 361
pNetworkRejectInd, 361
pNetworkTimeInd, 361
pServingSystemInd, 361
pSignalStrengthInd, 361
pSubscriptionInfoInd, 361
pSuppressSysInfoInd, 361
pSysInfoInd, 361
pSystemSelectionInd, 361
pack_nas_SLQSNasSwiIndicationRegister
nas.h, 1152
pack_nas_SLQSNasSwiIndicationRegister_t, 362
gsmUmtsDl, 363
gsmUmtsUl, 363
lteEmmDl, 363
lteEmmUl, 363
lteEsmDl, 363
lteEsmUl, 363
pRankIndicatorInd, 363
pTimer, 363
pack_nas_SLQSNasSwiModemStatus
nas.h, 1152
pack_nas_SLQSSetBandPreference
nas.h, 1153
pack_nas_SLQSSetBandPreference_t, 364
bandPref, 365
pack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1153
pack_nas_SLQSSetSignalStrengthsCallback_t, 366
bEnable, 366
pSigIndReq, 366
pack_nas_SLQSSetSysSelectionPref
nas.h, 1153
pack_nas_SLQSSetSysSelectionPref_t, 366
pAcqOrderPref, 371
pBandPref, 371
pCSGID, 371
pChgDuration, 371
pEmerMode, 371
pGWAcqOrderPref, 371
pLTEBandPref, 371
pModePref, 371
pNetSelPref, 371
pPRLPref, 371
pRAT, 371
pRoamPref, 371
pSrvDomainPref, 371
pSrvRegRestriction, 371
pTdscdmaBandPref, 372
pack_nas_SLQSSetSysSelectionPrefExt
nas.h, 1154
pack_nas_SLQSSetSysSelectionPrefExt_t, 372
pAcqOrderPref, 381
pBandPref, 381
pCSGID, 381
pChgDuration, 381
pCiotAcqOrderPref, 381
pCiotLteOpMode, 381
pCiotLteOpModePref, 381
pEmerMode, 381
pGWAcqOrderPref, 381
pLTEBandPref, 381
pLteM1BandPref, 381
pLteNb1BandPref, 381
pModePref, 381
pNetSelPref, 381
pNr5gBandPref, 381
pPRLPref, 381
pRAT, 382
pRoamPref, 382
pSrvDomainPref, 382
pSrvRegRestriction, 382

pTdscdmaBandPref, 382
pack_nas_SLQSSwiGetHDRPersonality
nas.h, 1154
pack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1155
pack_nas_SLQSSwiGetHRPDStats
nas.h, 1155
pack_nas_SLQSSwiGetLteCQI
nas.h, 1155
pack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1156
pack_nas_SLQSSwiNetworkDebug
nas.h, 1156
pack_nas_SLQSSwiPSDetach
nas.h, 1156
pack_nas_SLQSSwiPSDetach_t, 382
pDetachAction, 382
pack_nas_SetACCOLC
nas.h, 1142
pack_nas_SetACCOLC_t, 347
accolc, 347
spc, 347
pack_nas_SetCDMANetworkParameters
nas.h, 1142
pack_nas_SetCDMANetworkParameters_t, 347
pApplication, 348
pBroadcast, 348
pCustomSCP, 348
pForceRev0, 348
pProtocol, 348
pRoaming, 348
pSPC, 348
pack_nas_SetLURejectCallback
nas.h, 1143
pack_nas_SetNetworkPreference
nas.h, 1143
pack_nas_SetNetworkPreference_t, 349
Duration, 349
TechnologyPref, 349
Tlvresult, 349
pack_nas_SetRFInfoCallback
nas.h, 1143
pack_nas_SlqsGetLTECphyCAInfo
nas.h, 1144
pack_pds_ForceXTRADownload
pds.h, 1185
pack_pds_GetPDSDefaults
pds.h, 1186
pack_pds_GetPDSState
pds.h, 1186
pack_pds_GetPortAutomaticTracking
pds.h, 1186
pack_pds_GetServiceAutomaticTracking
pds.h, 1187
pack_pds_GetXTRAAutomaticDownload
pds.h, 1187
pack_pds_GetXTRANetwork
pds.h, 1187
pack_pds_GetXTRAValidity
pds.h, 1188
pack_pds_PDSInjectTimeReference
pds.h, 1188
pack_pds_PDSInjectTimeReference_t, 382
systemDiscontinuities, 383
systemTime, 383
pack_pds_ResetPDSData
pds.h, 1188
pack_pds_ResetPDSData_t, 383
pCellDataMask, 383
pGPSDataMask, 384
pack_pds_SLQSGetAGPSConfig
pds.h, 1191
pack_pds_SLQSGetAGPSConfig_t, 387
pNetworkMode, 388
pack_pds_SLQSGetGPSStateInfo
pds.h, 1192
pack_pds_SLQSPDSDeterminePosition
pds.h, 1192
pack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1192
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t, 388
forceFlag, 388
timeBase, 388
timeMsec, 388
timeUncMsec, 389
pack_pds_SLQSPDSInjectPositionData
pds.h, 1193
pack_pds_SLQSPDSInjectPositionData_t, 389
pAltitudeWrtEllipsoid, 390
pAltitudeWrtSealevel, 390
pHorizontalConfidence, 390
pHorizontalUncCircular, 390
pLatitude, 390
pLongitude, 390
pPositionSource, 390
pTimeStamp, 390
pTimeType, 390
pVerticalConfidence, 390
pVerticalUnc, 390
pack_pds_SLQSSetAGPSConfig
pds.h, 1193
pack_pds_SLQSSetAGPSConfig_t, 391
pNetworkMode, 391
pServerAddress, 391
pServerPort, 391
pServerURL, 391
pack_pds_SLQSSetPositionMethodState
pds.h, 1194
pack_pds_SLQSSetPositionMethodState_t, 391
pWifiState, 392
pXtraDataState, 392
pXtraTimeState, 392
pack_pds_SetEventReportCallback
pds.h, 1189
pack_pds_SetEventReportCallback_t, 384

posDataNmea, 384
rptPosData, 384
pack_pds_SetPDSDefaults
pds.h, 1189
pack_pds_SetPDSDefaults_t, 384
accuracy, 385
interval, 385
operation, 385
timeout, 385
pack_pds_SetPDSState
pds.h, 1190
pack_pds_SetPDSState_t, 385
enable, 385
pack_pds_SetPortAutomaticTracking
pds.h, 1190
pack_pds_SetPortAutomaticTracking_t, 385
bAuto, 386
pack_pds_SetServiceAutomaticTracking
pds.h, 1190
pack_pds_SetServiceAutomaticTracking_t, 386
bAuto, 386
pack_pds_SetXTRAAutomaticDownload
pds.h, 1191
pack_pds_SetXTRAAutomaticDownload_t, 386
bEnabled, 387
interval, 387
pack_pds_SetXTRANetwork
pds.h, 1191
pack_pds_SetXTRANetwork_t, 387
preference, 387
pack_pds_StartPDSTrackingSessionExt
pds.h, 1194
pack_pds_StartPDSTrackingSessionExt_t, 392
fixAccuracy, 393
fixCount, 393
fixInterval, 393
fixTimeout, 393
sessionControl, 393
sessionOperation, 393
sessionServerOption, 393
sessionType, 393
pack_pds_StopPDSTrackingSession
pds.h, 1194
pack_qmi_t, 393
msgid, 394
svc, 394
timeout, 394
xid, 394
pack_qos_BindDataPort
qos.h, 1253
pack_qos_BindDataPort_t, 394
pMuxID, 395
pPeripheralEndPointID, 395
pSIODataPort, 395
pack_qos_SLQSQosGetNetworkStatus
qos.h, 1253
pack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 1254
pack_qos_SLQSQosSwiReadApnExtraParams_t, 395
apnId, 395
pack_qos_SLQSQosSwiReadDataStats
qos.h, 1255
pack_qos_SLQSQosSwiReadDataStats_t, 395
apnId, 395
pack_qos_SLQSSetQosEventCallback
qos.h, 1255
pack_qos_SLQSSetQosEventCallback_t, 395
enable, 396
pack_rms_GetSMSWake
rms.h, 1261
pack_rms_SetSMSWake
rms.h, 1261
pack_rms_SetSMSWake_t, 396
enabled, 396
wake_mask, 396
pack_sar_SLQSGetRfSarState
sar.h, 1263
pack_sar_SLQSSetRfSarState
sar.h, 1263
pack_sar_SLQSSetRfSarState_t, 396
RfSarState, 397
pack_sms_GetSMSCAddress
sms.h, 1269
pack_sms_SLQSDeleteSMS
sms.h, 1270
pack_sms_SLQSDeleteSMS_t, 399
pMessageIndex, 400
pMessageMode, 400
pMessageTag, 400
storageType, 400
pack_sms_SLQSGetIndicationRegister
sms.h, 1271
pack_sms_SLQSGetMessageWaiting
sms.h, 1271
pack_sms_SLQSGetSMS
sms.h, 1271
pack_sms_SLQSGetSMS_t, 400
messageIndex, 401
pMessageMode, 401
storageType, 401
pack_sms_SLQSGetSMSList
sms.h, 1272
pack_sms_SLQSGetSMSList_t, 401
pMessageMode, 402
pRequestedTag, 402
storageType, 402
pack_sms_SLQSGetSmsBroadcastConfig
sms.h, 1272
pack_sms_SLQSGetSmsBroadcastConfig_t, 401
mode, 401
pack_sms_SLQSGetTransLayerInfo
sms.h, 1273
pack_sms_SLQSGetTransNWRegInfo
sms.h, 1273
pack_sms_SLQSModifySMSStatus
sms.h, 1273

pack_sms_SLQSModifySMSStatus_t, 402
messageIndex, 403
messageTag, 403
pMessageMode, 403
storageType, 403
pack_sms_SLQSSendAsyncSMS
sms.h, 1274
pack_sms_SLQSSendAsyncSMS_t, 403
pSendSmsParams, 403
pack_sms_SLQSSetIndicationRegister
sms.h, 1274
pack_sms_SLQSSetIndicationRegister_t, 403
pSetIndicationRegReq, 404
pack_sms_SLQSSetSmsBroadcastActivation
sms.h, 1274
pack_sms_SLQSSetSmsBroadcastActivation_t, 404
broadcastActivate, 404
mode, 404
pack_sms_SLQSSetSmsBroadcastConfig
sms.h, 1275
pack_sms_SLQSSetSmsBroadcastConfig_t, 404
mode, 405
pBroadcastConfig, 405
pCDMABroadcastConfig, 405
pack_sms_SLQSSetSmsStorage
sms.h, 1275
pack_sms_SLQSSetSmsStorage_t, 405
smsStorage, 405
pack_sms_SLQSSmsGetMaxStorageSize
sms.h, 1275
pack_sms_SLQSSmsGetMaxStorageSize_t, 406
pMaxStorageSizeReq, 406
pack_sms_SLQSSmsGetMessageProtocol
sms.h, 1276
pack_sms_SLQSSmsSetRoutes
sms.h, 1276
pack_sms_SLQSSmsSetRoutes_t, 406
pSetRoutesReq, 406
pack_sms_SLQSSwiGetSMSStorage
sms.h, 1277
pack_sms_SaveSMS
sms.h, 1269
pack_sms_SaveSMS_t, 397
messageFormat, 398
messageSize, 398
pMessage, 398
storageType, 398
pack_sms_SendSMS
sms.h, 1269
pack_sms_SendSMS_t, 398
messageFormat, 398
messageSize, 398
pLinktimer, 398
pMessage, 399
pack_sms_SetNewSMSCallback
sms.h, 1270
pack_sms_SetNewSMSCallback_t, 399
status, 399
pack_sms_SetSMSCAddress
sms.h, 1270
pack_sms_SetSMSCAddress_t, 399
pSMSCAddress, 399
pSMSCType, 399
pack_swiaudio_SLQSGetM2MAVMute
swiaudio.h, 1289
pack_swiaudio_SLQSGetM2MAVMute_t, 407
Profile, 408
pack_swiaudio_SLQSGetM2MAudioProfile
swiaudio.h, 1289
pack_swiaudio_SLQSGetM2MAudioProfile_t, 406
pGenerator, 407
pack_swiaudio_SLQSGetM2MAudioVolume
swiaudio.h, 1289
pack_swiaudio_SLQSGetM2MAudioVolume_t, 407
Generator, 407
Profile, 407
pack_swiaudio_SLQSGetM2MSpkrGain
swiaudio.h, 1290
pack_swiaudio_SLQSGetM2MSpkrGain_t, 408
Profile, 408
pack_swiaudio_SLQSSetM2MAVMute
swiaudio.h, 1292
pack_swiaudio_SLQSSetM2MAVMute_t, 411
EarMute, 412
MicMute, 412
pCwtMute, 412
Profile, 412
pack_swiaudio_SLQSSetM2MAudioAVCFG
swiaudio.h, 1290
pack_swiaudio_SLQSSetM2MAudioAVCFG_t, 408
Device, 409
Profile, 409
pack_swiaudio_SLQSSetM2MAudioLPBK
swiaudio.h, 1290
pack_swiaudio_SLQSSetM2MAudioLPBK_t, 409
Enable, 409
pack_swiaudio_SLQSSetM2MAudioNVDef
swiaudio.h, 1291
pack_swiaudio_SLQSSetM2MAudioProfile
swiaudio.h, 1291
pack_swiaudio_SLQSSetM2MAudioProfile_t, 409
pCwtMute, 410
pEarMute, 410
pGenerator, 410
pMicMute, 410
pVolume, 410
Profile, 410
pack_swiaudio_SLQSSetM2MAudioVolume
swiaudio.h, 1292
pack_swiaudio_SLQSSetM2MAudioVolume_t, 411
Generator, 411
Level, 411
Profile, 411
pack_swiaudio_SLQSSetM2MSpkrGain
swiaudio.h, 1292
pack_swiaudio_SLQSSetM2MSpkrGain_t, 412

Profile, 412
Value, 412
pack_swiavms_SLQSAVMSGetSettings
swiavms.h, 1299
pack_swiavms_SLQSAVMSGetSettings_v2
swiavms.h, 1299
pack_swiavms_SLQSAVMSSendSelection
swiavms.h, 1300
pack_swiavms_SLQSAVMSSendSelection_t, 413
pDeferTime, 413
pPackageID, 413
pRejectReason, 413
selection, 413
pack_swiavms_SLQSAVMSSessionGetInfo
swiavms.h, 1300
pack_swiavms_SLQSAVMSSetSettings
swiavms.h, 1301
pack_swiavms_SLQSAVMSSetSettings_t, 413
AutoConnect, 415
AutoReboot, 415
pAPNInfo, 415
pConnectionRetryTimers, 415
pNotifStore, 415
pPeriodInfo, 415
pPollingTimer, 415
PromptFwDownload, 415
PromptFwUpdate, 415
pack_swiavms_SLQSAVMSSetSettings_v2
swiavms.h, 1302
pack_swiavms_SLQSAVMSSetSettings_v2_t, 415
AutoConnect, 416
pAutoReboot, 417
pNotifStore, 417
pPeriodInfo, 417
pPollingTimer, 417
PromptFwDownload, 417
PromptFwUpdate, 417
pack_swiavms_SLQSAVMSSetSettingsNoAutoReboot-Field
swiavms.h, 1303
pack_swiavms_SLQSAVMSStartSession
swiavms.h, 1303
pack_swiavms_SLQSAVMSStartSession_t, 417
sessionType, 417
pack_swiavms_SLQSAVMSStopSession
swiavms.h, 1304
pack_swiavms_SLQSAVMSStopSession_t, 417
sessionType, 418
pack_swiavms_SLQSAvmsSetEventReport
swiavms.h, 1301
pack_swidms_SLQSSwiDmsGetMTU
swidms.h, 1312
pack_swidms_SLQSSwiDmsGetUsbNetNum
swidms.h, 1312
pack_swidms_SLQSSwiDmsSetMTU
swidms.h, 1312
pack_swidms_SLQSSwiDmsSetMTU_t, 418
MTUSize, 418
pack_swidms_SLQSSwiDmsSetUsbNetNum
swidms.h, 1313
pack_swidms_SLQSSwiDmsSetUsbNetNum_t, 418
nUsbNetNum, 419
pack_swiloc_SwiLocGetAutoStart
swiloc.h, 1315
pack_swiloc_SwiLocSetAutoStart
swiloc.h, 1315
pack_swiloc_SwiLocSetAutoStart_t, 419
fix_rate, 420
fix_type, 420
function, 420
max_dist, 420
max_time, 420
set_fix_rate, 420
set_fix_type, 420
set_function, 420
set_max_dist, 420
set_max_time, 420
pack_swioma_SLQSOMADMAlertCallback
swioma.h, 1317
pack_swioma_SLQSOMADMCancelSession
swioma.h, 1318
pack_swioma_SLQSOMADMCancelSession_t, 420
sessionType, 421
pack_swioma_SLQSOMADMCancelSessionExt
swiomaext.h, 1326
pack_swioma_SLQSOMADMCancelSessionExt_t, 421
sessionType, 421
pack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 1319
pack_swioma_SLQSOMADMGetSessionInfo_t, 421
SessionType, 422
pack_swioma_SLQSOMADMGetSettings
swioma.h, 1319
pack_swioma_SLQSOMADMSendSelection
swioma.h, 1320
pack_swioma_SLQSOMADMSendSelection_t, 422
pDeferTime, 422
pRejectReason, 422
selection, 422
pack_swioma_SLQSOMADMSendSelectionExt
swiomaext.h, 1327
pack_swioma_SLQSOMADMSendSelectionExt_t, 422
selection, 423
pack_swioma_SLQSOMADMSetSettings
swioma.h, 1320
pack_swioma_SLQSOMADMSetSettings_t, 423
FOTAdownload, 424
pAutosdm, 424
pFwAutoCheck, 424
pack_swioma_SLQSOMADMSetSettingsExt
swiomaext.h, 1327
pack_swioma_SLQSOMADMSetSettingsExt_t, 424
pack_swioma_SLQSOMADMStartSession
swioma.h, 1321
pack_swioma_SLQSOMADMStartSession_t, 425
sessionType, 426

pack_swioma_SLQSOMADMStartSessionExt
swiomaext.h, 1328
pack_swioma_SLQSOMADMStartSessionExt_t, 426
sessionType, 426
pack_tmd_SLQSTmdDeRegNotMitigationLvl
tmd.h, 1333
pack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 426
mitigationDevID, 426
mitigationDevIDLen, 426
pack_tmd_SLQSTmdGetMitigationDevList
tmd.h, 1333
pack_tmd_SLQSTmdGetMitigationLvl
tmd.h, 1334
pack_tmd_SLQSTmdGetMitigationLvl_t, 427
mitigationDevID, 427
mitigationDevIDLen, 427
pack_tmd_SLQSTmdRegNotMitigationLvl
tmd.h, 1334
pack_tmd_SLQSTmdRegNotMitigationLvl_t, 427
mitigationDevID, 427
mitigationDevIDLen, 428
pack_uim_ChangePin
uim.h, 1340
pack_uim_ChangePin_t, 428
changePIN, 428
EncryptedPIN1, 428
pIndicationToken, 428
pKeyReferenceID, 428
sessionInfo, 429
Tlvresult, 429
pack_uim_GetCardStatus
uim.h, 1340
pack_uim_ReadTransparent
uim.h, 1340
pack_uim_ReadTransparent_t, 429
fileIndex, 429
pEncryptData, 429
pIndicationToken, 430
readTransparent, 430
sessionInfo, 430
Tlvresult, 430
pack_uim_SLQSUIMAuthenticate
uim.h, 1341
pack_uim_SLQSUIMAuthenticate_t, 431
authData, 431
pIndicationToken, 431
sessionInfo, 431
pack_uim_SLQSUIMDepersonalization
uim.h, 1341
pack_uim_SLQSUIMDepersonalization_t, 431
depersonilisationInfo, 432
pack_uim_SLQSUIMEventRegister
uim.h, 1342
pack_uim_SLQSUIMEventRegister_t, 432
eventMask, 432
pack_uim_SLQSUIMGetConfiguration
uim.h, 1342
pack_uim_SLQSUIMGetConfiguration_t, 432
pConfigurationMask, 433
pack_uim_SLQSUIMGetFileAttributes
uim.h, 1342
pack_uim_SLQSUIMGetFileAttributes_t, 433
fileIndex, 433
pIndicationToken, 433
sessionInfo, 433
pack_uim_SLQSUIMGetSlotsStatus
uim.h, 1343
pack_uim_SLQSUIMPowerDown
uim.h, 1343
pack_uim_SLQSUIMPowerDown_t, 434
slot, 434
pack_uim_SLQSUIMPowerUp
uim.h, 1344
pack_uim_SLQSUIMPowerUp_t, 434
pIgnoreHotSwapSwitch, 434
slot, 434
pack_uim_SLQSUIMRefreshComplete
uim.h, 1344
pack_uim_SLQSUIMRefreshComplete_t, 435
refreshComplete, 435
sessionInfo, 435
pack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 1344
pack_uim_SLQSUIMRefreshGetLastEvent_t, 435
sessionInfo, 435
pack_uim_SLQSUIMRefreshOK
uim.h, 1345
pack_uim_SLQSUIMRefreshOK_t, 436
OKtoRefresh, 436
sessionInfo, 436
pack_uim_SLQSUIMRefreshRegister
uim.h, 1345
pack_uim_SLQSUIMRefreshRegister_t, 436
regRefresh, 436
sessionInfo, 437
pack_uim_SLQSUIMReset
uim.h, 1345
pack_uim_SLQSUIMSwitchSlot
uim.h, 1346
pack_uim_SLQSUIMSwitchSlot_t, 437
bLogicalSlot, 437
ulPhysicalSlot, 437
pack_uim_SetPinProtection
uim.h, 1341
pack_uim_SetPinProtection_t, 430
EncryptedPIN1, 430
pIndicationToken, 431
pKeyReferenceID, 431
pinProtection, 431
sessionInfo, 431
Tlvresult, 431
pack_uim_UnblockPin
uim.h, 1346
pack_uim_UnblockPin_t, 437
EncryptedPIN1, 438
pIndicationToken, 438

pKeyReferenceID, 438
pinProtection, 438
sessionInfo, 438
Tlvresult, 438
pack_uim_VerifyPin
uim.h, 1346
pack_uim_VerifyPin_t, 438
pEncryptedPIN1, 439
pIndicationToken, 439
pKeyReferenceID, 439
sessionInfo, 439
Tlvresult, 439
verifyPIN, 439
pack_voice_AnswerUSSD
voice.h, 1361
pack_voice_AnswerUSSD_t, 439
pInfo, 440
pack_voice_CancelUSSD
voice.h, 1361
pack_voice_OriginateUSSD
voice.h, 1361
pack_voice_OriginateUSSD_t, 440
pInfo, 440
pack_voice_SLQSOriginateUSSD
voice.h, 1362
pack_voice_SLQSOriginateUSSD_t, 440
ussDCS, 441
ussData, 441
ussLen, 441
pack_voice_SLQSVoiceALSSelectLine
voice.h, 1362
pack_voice_SLQSVoiceALSSelectLine_t, 441
lineValue, 441
pack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 1362
pack_voice_SLQSVoiceALSSetLineSwitching_t, 441
switchOption, 442
pack_voice_SLQSVoiceAnswerCall
voice.h, 1363
pack_voice_SLQSVoiceAnswerCall_t, 442
pCallId, 442
pack_voice_SLQSVoiceBindSubscription
voice.h, 1363
pack_voice_SLQSVoiceBindSubscription_t, 442
subsType, 443
pack_voice_SLQSVoiceBurstDTMF
voice.h, 1363
pack_voice_SLQSVoiceBurstDTMF_t, 443
pack_voice_SLQSVoiceDialCall
voice.h, 1364
pack_voice_SLQSVoiceDialCall_t, 443
callNumber, 445
pCLIRType, 445
pCUGInfo, 445
pCallPartySubAdd, 445
pCallType, 445
pEmergencyCategory, 445
pSvcType, 445
pUUSInFo, 445
pack_voice_SLQSVoiceEndCall
voice.h, 1364
pack_voice_SLQSVoiceEndCall_t, 445
pCallId, 445
pack_voice_SLQSVoiceGetAllCallInfo
voice.h, 1365
pack_voice_SLQSVoiceGetCLIP
voice.h, 1366
pack_voice_SLQSVoiceGetCLIR
voice.h, 1367
pack_voice_SLQSVoiceGetCNAP
voice.h, 1367
pack_voice_SLQSVoiceGetCOLP
voice.h, 1367
pack_voice_SLQSVoiceGetCOLR
voice.h, 1368
pack_voice_SLQSVoiceGetCallBarring
voice.h, 1365
pack_voice_SLQSVoiceGetCallBarring_t, 445
pSvcClass, 446
reason, 446
pack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 1365
pack_voice_SLQSVoiceGetCallForwardingStatus_t, 446
pSvcClass, 447
Reason, 447
pack_voice_SLQSVoiceGetCallInfo
voice.h, 1366
pack_voice_SLQSVoiceGetCallInfo_t, 447
callID, 447
pack_voice_SLQSVoiceGetCallWaiting
voice.h, 1366
pack_voice_SLQSVoiceGetCallWaiting_t, 447
pSvcClass, 448
pack_voice_SLQSVoiceGetConfig
voice.h, 1368
pack_voice_SLQSVoiceGetConfig_t, 448
pAMRStatus, 449
pAirTimer, 449
pAutoAnswer, 449
pNamID, 449
pPrefVoicePrivacy, 449
pPrefVoiceSO, 449
pRoamTimer, 449
pTTYMode, 449
pVoiceDomainPref, 449
pack_voice_SLQSVoiceIndicationRegister
voice.h, 1368
pack_voice_SLQSVoiceIndicationRegister_t, 450
pRegDTMFEvents, 450
pRegVoicePrivacyEvents, 450
pSuppsNotifEvents, 450
pack_voice_SLQSVoiceManageCalls
voice.h, 1369
pack_voice_SLQSVoiceManageCalls_t, 450
pCallID, 451
SUPSType, 451

pack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 1369
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 451
pack_voice_SLQSVoiceSendFlash
voice.h, 1370
pack_voice_SLQSVoiceSendFlash_t, 452
pCallID, 452
pFlashPayLd, 452
pFlashType, 452
pack_voice_SLQSVoiceSetCallBarringPassword
voice.h, 1370
pack_voice_SLQSVoiceSetCallBarringPassword_t, 452
newPasswd, 453
newPasswdAgain, 453
oldPasswd, 453
Reason, 453
pack_voice_SLQSVoiceSetConfig
voice.h, 1370
pack_voice_SLQSVoiceSetConfig_t, 453
pAirTimerConfig, 455
pAutoAnswer, 455
pPrefVoiceDomain, 455
pPrefVoiceSO, 455
pRoamTimerConfig, 455
pTTYMode, 455
pack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 1371
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 455
privacyPref, 455
pack_voice_SLQSVoiceSetSUPSService
voice.h, 1371
pack_voice_SLQSVoiceSetSUPSService_t, 455
pCallBarringPasswd, 457
pCallForwardingNumber, 457
pServiceClass, 457
pTimerVal, 457
reason, 457
voiceSvc, 457
pack_voice_SLQSVoiceStartContDTMF
voice.h, 1371
pack_voice_SLQSVoiceStartContDTMF_t, 457
DTMFdigit, 458
pCallID, 458
pack_voice_SLQSVoiceStopContDTMF
voice.h, 1372
pack_voice_SLQSVoiceStopContDTMF_t, 458
callID, 458
pack_wds_DHCPv4ClientLeaseChange
wds.h, 1394
pack_wds_DHCPv4ClientLeaseChange_t, 458
pEnableNotification, 459
pack_wds_GetAutoconnect
wds.h, 1394
pack_wds_GetByteTotals
wds.h, 1395
pack_wds_GetConnectionRate
wds.h, 1395
pack_wds_GetDataBearerTechnology
wds.h, 1395
pack_wds_GetDefaultProfile
wds.h, 1396
pack_wds_GetDefaultProfile_t, 459
profiletype, 459
pack_wds_GetDefaultProfileNum
wds.h, 1396
pack_wds_GetDefaultProfileNum_t, 459
family, 459
type, 459
pack_wds_GetDormancyState
wds.h, 1397
pack_wds_GetDormancyState_t, 460
pack_wds_GetLastMobileIPError
wds.h, 1397
pack_wds_GetLastMobileIPError_t, 460
pack_wds_GetMobileIP
wds.h, 1397
pack_wds_GetMobileIP_t, 460
pack_wds_GetMobileIPProfile
wds.h, 1398
pack_wds_GetMobileIPProfile_t, 460
index, 461
pack_wds_GetPacketStatistics
wds.h, 1398
pack_wds_GetPacketStatistics_t, 461
pStatMask, 461
pack_wds_GetPacketStatus
wds.h, 1399
pack_wds_GetPacketStatus_t, 461
statmask, 462
pack_wds_GetSessionDuration
wds.h, 1399
pack_wds_GetSessionDuration_t, 462
pack_wds_GetSessionState
wds.h, 1400
pack_wds_RMSetTransferStatistics
wds.h, 1400
pack_wds_RMSetTransferStatistics_t, 462
RmTrasnferStaticsReq, 462
pack_wds_SLQSCreateProfile
wds.h, 1403
pack_wds_SLQSCreateProfile_t, 469
pCurProfile, 470
pProfileId, 470
pProfileType, 470
pack_wds_SLQSDeleteProfile
wds.h, 1404
pack_wds_SLQSDeleteProfile_t, 470
profileIndex, 470
profileType, 470
pack_wds_SLQSGet3GPPConfigItem
wds.h, 1404
pack_wds_SLQSGetCurrDataSystemStat
wds.h, 1405
pack_wds_SLQSGetCurrDataSystemStat_t, 470
pack_wds_SLQSGetCurrentChannelRate
wds.h, 1405

pack_wds_SLQSGetDUNCallInfo
wds.h, 1406
pack_wds_SLQSGetDUNCallInfo_t, 471
Mask, 472
pReportChannelRate, 472
pReportConnStatus, 472
pReportDataBearerTech, 472
pReportDormStatus, 472
pTransferStatInd, 472
pack_wds_SLQSGetDataBearerTechnology
wds.h, 1405
pack_wds_SLQSGetDataBearerTechnology_t, 471
pack_wds_SLQSGetProfileSettings
wds.h, 1406
pack_wds_SLQSGetProfileSettings_t, 472
ProfileId, 473
ProfileType, 473
pack_wds_SLQSGetRuntimeSettings
wds.h, 1407
pack_wds_SLQSGetRuntimeSettings_t, 473
pReqSettings, 474
pack_wds_SLQSModifyProfile
wds.h, 1407
pack_wds_SLQSModifyProfile_t, 474
curProfile, 475
pProfileId, 475
pProfileType, 475
pack_wds_SLQSResetPacketStatics
wds.h, 1408
pack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 1409
pack_wds_SLQSSGetDHCPv4ClientConfig_t, 478
pProfileId, 478
pack_wds_SLQSSGetLoopback
wds.h, 1410
pack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 1410
pack_wds_SLQSSSetDHCPv4ClientConfig_t, 478
pHwConfig, 479
pProfileId, 479
pack_wds_SLQSSSetLoopback
wds.h, 1410
pack_wds_SLQSSSetLoopback_t, 479
loopbackMode, 479
loopbackMultiplier, 479
pack_wds_SLQSSet3GPPConfigItem
wds.h, 1408
pack_wds_SLQSSet3GPPConfigItem_t, 475
p3gppRelease, 476
pProfileList, 476
pack_wds_SLQSSetIPFamilyPreference
wds.h, 1408
pack_wds_SLQSSetIPFamilyPreference_t, 477
IPFamilyPreference, 477
pack_wds_SLQSSetWdsEventCallback
wds.h, 1409
pack_wds_SLQSSetWdsEventCallback_t, 477
currentDataBearer, 478
dataBearer, 478
dataBearerTechExt, 478
dataSystemStatus, 478
dormancyStatus, 478
interval, 478
mobileIP, 478
transferStats, 478
pack_wds_SLQSStartDataSession
wds.h, 1411
pack_wds_SLQSStartDataSession_t, 479
pAuth, 480
pPass, 480
pTech, 480
pUser, 480
pprofileid3gpp, 480
pprofileid3gpp2, 480
pack_wds_SLQSStopDataSession
wds.h, 1411
pack_wds_SLQSStopDataSession_t, 481
psid, 481
pack_wds_SLQSWdsGoActive
wds.h, 1412
pack_wds_SLQSWdsGoDormant
wds.h, 1412
pack_wds_SLQSWdsSetEventReport
wds.h, 1412
pack_wds_SLQSWdsSetEventReport_t, 481
pCurrChannelRateInd, 482
pCurrDataBearerTechInd, 482
pCurrPrefDataSysInd, 482
pDataBearerTechInd, 482
pDataCallStatusChangeInd, 482
pDataSystemStatusChangeInd, 482
pDormancyStatusInd, 483
pMIPStatusInd, 483
pTransferStatInd, 483
pack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 1413
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 483
contextId, 483
contextType, 483
pack_wds_SetAutoconnect
wds.h, 1400
pack_wds_SetAutoconnect_t, 463
acroamsetting, 463
acsetting, 463
pack_wds_SetDefaultProfile
wds.h, 1401
pack_wds_SetDefaultProfile_t, 463
authentication, 464
ipAddress, 465
pApnname, 465
pName, 465
pPassword, 465
pUsername, 465
pdpType, 465
primaryDNS, 465
profileType, 465

secondaryDNS, 465
pack_wds_SetDefaultProfileNum
wds.h, 1401
pack_wds_SetDefaultProfileNum_t, 465
family, 465
index, 465
type, 465
pack_wds_SetMobileIP
wds.h, 1402
pack_wds_SetMobileIP_t, 466
mode, 466
pack_wds_SetMobileIPParameters
wds.h, 1402
pack_wds_SetMobileIPParameters_t, 466
pHA2002bis, 467
pHAAuthenticator, 467
pMode, 467
pReRegPeriod, 467
pReRegTraffic, 467
pRetryInterval, 467
pRetryLimit, 467
pSPC, 467
pack_wds_SetMobileIPProfile
wds.h, 1402
pack_wds_SetMobileIPProfile_t, 467
index, 468
pAAASPI, 469
pAddress, 469
pEnabled, 469
pHASPI, 469
pMNAAA, 469
pMNHA, 469
pNAI, 469
pPrimaryHA, 469
pRevTunneling, 469
pSecondaryHA, 469
spc, 469
pack_wds_SetMuxID
wds.h, 1403
PackCreateProfileOut, 483
ExtErrorCode, 484
ProfileIndex, 484
ProfileType, 484
PackSwiAVMSSettingsAPNInfo, 487
bAPNLength, 487
bPWDLength, 487
bUnameLength, 487
pAPN, 488
pPWD, 488
pUname, 488
PackSwiAVMSSettingsConnectionRetryTimers, 488
Timers, 488
PackSwiAVMSSettingsPeriodsInfo, 488
max, 489
min, 489
PackSwiAvmsSetSettingsAPNInfo, 485
bAPNLength, 485
bPWDLength, 485
bUnameLength, 486
szAPN, 486
szPWD, 486
szUname, 486
PackSwiAvmsSetSettingsConnectionRetryTimers, 486
Timers, 486
PackSwiAvmsSetSettingsPeriodInfo, 487
ulMax, 487
ulMin, 487
package_name
unpack_omaDmFotaTlv_t, 741
packageid_str
unpack_dms_GetFirmwareInfo_t, 567
packetZone
nas_CDMASysInfo, 153
packetZoneValid
nas_CDMASysInfo, 153
packgetDyingGaspCfg, 484
pDestSMSContent, 484
pDestSMSNum, 484
packgetDyingGaspStatistics, 484
pSMSAttemptedFlag, 485
pTimeStamp, 485
ParamPresenceMask
pack_dms_UIMGetICCID_t, 316
unpack_audio_SLQSGetAudioPathConfig_t, 548
unpack_audio_SLQSGetAudioProfile_t, 550
unpack_audio_SLQSGetAudioVolTLBConfig_t, 550
unpack_audio_SLQSSetAudioVolTLBConfig_t, 551
unpack_cat_SetCatEventCallback_ind_t, 551
unpack_cat_SetCATEventCallback_t, 552
unpack_dms_GetActivationState_t, 553
unpack_dms_GetBandCapability_t, 555
unpack_dms_GetCrashAction_t, 555
unpack_dms_GetCustFeature_t, 558
unpack_dms_GetCustFeaturesV2_t, 559
unpack_dms_GetDeviceCap_t, 560
unpack_dms_GetDeviceCapabilities_t, 562
unpack_dms_GetDeviceHardwareRev_t, 562
unpack_dms_GetDeviceMfr_t, 563
unpack_dms_GetDeviceSerialNumbers_t, 565
unpack_dms_GetFirmwareInfo_t, 567
unpack_dms_GetFirmwareRevision_t, 568
unpack_dms_GetFirmwareRevisions_t, 569
unpack_dms_GetFSN_t, 570
unpack_dms_GetHardwareRevision_t, 570
unpack_dms_GetIMSI_t, 571
unpack_dms_GetManufacturer_t, 571
unpack_dms_GetModelID_t, 572
unpack_dms_GetNetworkTime_t, 573
unpack_dms_GetOfflineReason_t, 574
unpack_dms_GetPower_t, 575
unpack_dms_GetPRLVersion_t, 576
unpack_dms_GetSerialNumbers_t, 577
unpack_dms_GetUSBComp_t, 579
unpack_dms_GetVoiceNumber_t, 580

unpack_dms_PSMCfgChange_ind_t, 581
unpack_dms_ResetToFactoryDefaults_t, 582
unpack_dms_SetActivationStatusCallback_t, 582
unpack_dms_SetCrashAction_t, 583
unpack_dms_SetCustFeature_t, 583
unpack_dms_SetCustFeaturesV2_t, 584
unpack_dms_SetEventReport_ind_t, 585
unpack_dms_SetEventReport_t, 585
unpack_dms_SetFirmwarePreference_t, 586
unpack_dms_SetIndicationRegister_t, 586
unpack_dms_SetPower_t, 587
unpack_dms_SetUSBComp_t, 587
unpack_dms_SLQSDmsSwiGetPCInfo_t, 590
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 591
unpack_dms_SLQSDmsSwiGetResetInfo_t, 592
unpack_dms_SLQSDmsSwiIndicationRegister_t, 593
unpack_dms_SLQSGetBandCapability_t, 597
unpack_dms_SLQSGetERIFile_t, 598
unpack_dms_SLQSGetPowerSaveModeConfig_t, 599
unpack_dms_SLQSSetPowerSaveModeConfig_t, 600
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 600
unpack_dms_SLQSSwiGetCrashInfo_t, 601
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 602
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 602
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
unpack_dms_SLQSSwiGetHostDevInfo_t, 606
unpack_dms_SLQSSwiGetOSInfo_t, 607
unpack_dms_SLQSSwiGetSerialNoExt_t, 607
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 608
unpack_dms_SLQSSwiSetHostDevInfo_t, 608
unpack_dms_SLQSSwiSetOSInfo_t, 609
unpack_dms_SLQSUIMGetState_t, 610
unpack_dms_SwiEventReportCallBack_ind_t, 610
unpack_dms_SwiSetEventReport_t, 611
unpack_dms_UIMGetControlKeyStatus_t, 612
unpack_dms_UIMGetICCID_t, 612
unpack_dms_UIMGetPINStatus_t, 614
unpack_dms_UIMSetControlKeyProtection_t, 615
unpack_dms_UIMSetPINProtection_t, 616
unpack_dms_UIMUnblockControlKey_t, 616
unpack_fms_GetImagesPreference_t, 617
unpack_fms_GetStoredImages_t, 618
unpack_fms_SetImagesPreference_t, 619
unpack_ims_SLQSGetIMSSMSConfig_t, 620
unpack_ims_SLQSGetIMSUserConfig_t, 620
unpack_ims_SLQSGetIMSVoIPConfig_t, 623
unpack_ims_SLQSGetRegMgrConfig_t, 624
unpack_ims_SLQSGetSIPConfig_t, 626
unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 627
unpack_ims_SLQSSetIMSSMSConfig_t, 627
unpack_ims_SLQSSetIMSUserConfig_t, 628
unpack_ims_SLQSSetIMSVoIPConfig_t, 628
unpack_ims_SLQSSetRegMgrConfig_t, 628
unpack_ims_SLQSSetSIPConfig_t, 629
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
unpack_ims_SLQSSMSCfgCallBack_ind_t, 631
unpack_ims_SLQSUserCfgCallBack_ind_t, 632
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
unpack_imsa_SLQSGetIMSARegStatus_t, 635
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, 638
unpack_imsa_SLQSImsaRatStatusCallBack_ind_t, 639
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 640
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 641
unpack_loc_BestAvailPos_Ind_t, 647
unpack_loc_CradleMountCallback_Ind_t, 648
unpack_loc_Delete_Assist_Data_t, 649
unpack_loc_DeleteAssistData_Ind_t, 650
unpack_loc_EngineState_Ind_t, 651
unpack_loc_EventNMEA_Ind_t, 651
unpack_loc_EventRegister_t, 652
unpack_loc_EventTimeSyncCallback_Ind_t, 652
unpack_loc_GetServer_Ind_t, 654
unpack_loc_GnssSvInfo_Ind_t, 655
unpack_loc_InjectPositionCallback_Ind_t, 655
unpack_loc_InjectSensorDataCallback_Ind_t, 657
unpack_loc_InjectTimeSyncDataCallback_Ind_t, 658
unpack_loc_InjectUTCTimeCallback_Ind_t, 658
unpack_loc_PositionRpt_Ind_t, 664
unpack_loc_SensorStreamingCallback_Ind_t, 666
unpack_loc_SetExtPowerConfig_Ind_t, 667
unpack_loc_SetExtPowerState_t, 667
unpack_loc_SetOperationMode_Ind_t, 668
unpack_loc_SetOperationMode_t, 669
unpack_loc_SLQSLOCGetBestAvailPos_t, 670
unpack_loc_Start_t, 670
unpack_loc_Stop_t, 671
unpack_nas_GetACCOLC_t, 671
unpack_nas_GetANAAAAuthenticationStatus_t, 672
unpack_nas_GetCDMANetworkParameters_t, 674
unpack_nas_GetHomeNetwork3GPP2_t, 676
unpack_nas_GetHomeNetwork_t, 678
unpack_nas_GetNetworkPreference_t, 679
unpack_nas_GetRFInfo_t, 680
unpack_nas_GetServingNetwork_t, 683
unpack_nas_GetServingNetworkCapabilities_t, 684
unpack_nas_GetSignalStrengths_t, 684
unpack_nas_PerformNetworkScan_t, 686
unpack_nas_SetDataCapabilitiesCallback_ind_t, 687
unpack_nas_SetEventReportInd_t, 688

unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 689
unpack_nas_SetNetworkPreference_t, 689
unpack_nas_SetRoamingIndicatorCallback_ind_t, 690
unpack_nas_SetServingSystemCallback_ind_t, 690
unpack_nas_SLQSGetErrorRate_t, 692
unpack_nas_SlqsGetLTECphyCAInfo_t, 692
unpack_nas_SLQSGetNetworkTime_t, 693
unpack_nas_SLQSGetOperatorNameData_t, 694
unpack_nas_SLQSGetPLMNName_t, 697
unpack_nas_SLQSGetServingSystem_t, 702
unpack_nas_SLQSGetSignalStrength_t, 705
unpack_nas_SLQSGetSysInfo_t, 708
unpack_nas_SLQSGetSysSelectionPref_t, 713
unpack_nas_SLQSNasGet3GPP2Subscription_t, 717
unpack_nas_SLQSNasGetCellLocationInfo_t, 718
unpack_nas_SLQSNASGeteDRXParams_t, 719
unpack_nas_SLQSNasGetHDRColorCode_t, 720
unpack_nas_SLQSNasGetSigInfo_t, 722
unpack_nas_SLQSNasGetTxRxInfo_t, 723
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 724
unpack_nas_SLQSNasSigInfoCallback_ind_t, 725
unpack_nas_SLQSNASSwiGetChannelLock_t, 726
unpack_nas_SLQSNasSwiModemStatus_t, 727
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 727
unpack_nas_SLQSNasTimerCallback_ind_t, 728
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 729
unpack_nas_SLQSSwiGetHDRPersonality_t, 730
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 730
unpack_nas_SLQSSwiGetHRPDStats_t, 731
unpack_nas_SLQSSwiGetLteCQI_t, 732
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 733
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t, 733
unpack_nas_SLQSSwiNetworkDebug_t, 734
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t, 735
unpack_nas_SLQSSysInfoCallback_ind_t, 738
unpack_pds_GetPDSDefaults_t, 743
unpack_pds_GetPDSState_t, 744
unpack_pds_GetPortAutomaticTracking_t, 745
unpack_pds_GetServiceAutomaticTracking_t, 745
unpack_pds_GetXTRAAutomaticDownload_t, 746
unpack_pds_GetXTRANetwork_t, 746
unpack_pds_GetXTRAValidity_t, 747
unpack_pds_SetEventReport_Ind_t, 749
unpack_pds_SetPdsState_Ind_t, 750
unpack_pds_SLQSGetAGPSConfig_t, 751
unpack_pds_SLQSGetGPSStateInfo_t, 755
unpack_qos_BindDataPort_t, 756
unpack_qos_SLQSQosGetNetworkStatus_t, 762
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 763
unpack_qos_SLQSQosSwiReadDataStats_t, 765
unpack_qos_SLQSSetQosEventCallback_ind_t, 766
unpack_qos_SLQSSetQosEventCallback_t, 766
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 767
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 767
unpack_qos_SLQSSetQosStatusCallback_ind_t, 769
unpack_result_t, 778
unpack_rms_GetSMSWake_t, 779
unpack_rms_SetSMSWake_t, 780
unpack_sar_SLQSGetRfSarState_t, 781
unpack_sms_GetSMSCAddress_t, 782
unpack_sms_SaveSMS_t, 783
unpack_sms_SendSMS_t, 783
unpack_sms_SetNewSMSCallback_ind_t, 785
unpack_sms_SetNewSMSCallback_t, 785
unpack_sms_SetSMSCAddress_t, 786
unpack_sms_SLQSDeleteSMS_t, 786
unpack_sms_SLQSGetIndicationRegister_t, 787
unpack_sms_SLQSGetMessageWaiting_t, 787
unpack_sms_SLQSGetSMS_t, 788
unpack_sms_SLQSGetSmsBroadcastConfig_t, 789
unpack_sms_SLQSGetSMSList_t, 790
unpack_sms_SLQSGetTransLayerInfo_t, 790
unpack_sms_SLQSGetTransNWRegInfo_t, 791
unpack_sms_SLQSModifySMSStatus_t, 791
unpack_sms_SLQSNWRegInfoCallback_ind_t, 792
unpack_sms_SLQSSendAsyncSMS_t, 792
unpack_sms_SLQSSetIndicationRegister_t, 793
unpack_sms_SLQSSetSmsBroadcastActivation_t, 793
unpack_sms_SLQSSetSmsBroadcastConfig_t, 794
unpack_sms_SLQSSetSmsStorage_t, 794
unpack_sms_SLQSSmsGetMaxStorageSize_t, 794
unpack_sms_SLQSSmsGetMessageProtocol_t, 795
unpack_sms_SLQSSmsSetRoutes_t, 795
unpack_sms_SLQSSwiGetSMSStorage_t, 796
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 797
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 799
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 800
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t, 800
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 802

unpack_swiaudio_SLQSGetM2MAudioVolume_t, 802
unpack_swiaudio_SLQSGetM2MAVMute_t, 803
unpack_swiaudio_SLQSGetM2MSpkrGain_t, 804
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
unpack_swiavms_SLQSAVMSSendSelection_t, 811
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 811
unpack_swiavms_SLQSAvmsSetEventReport_t, 812
unpack_swiavms_SLQSAVMSSetSettings_t, 812
unpack_swiavms_SLQSAVMSSetSettings_v2_t, 813
unpack_swiavms_SLQSAVMSStartSession_t, 813
unpack_swiavms_SLQSAVMSStopSession_t, 814
unpack_swidms_SLQSSwiDmsGetMTU_t, 815
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 815
unpack_swidms_SLQSSwiDmsSetMTU_t, 816
unpack_swiloc_SwiLocGetAutoStart_t, 818
path
uim_fileInfo, 535
pathLen
uim_fileInfo, 535
pbAuto
unpack_pds_GetPortAutomaticTracking_t, 745
unpack_pds_GetServiceAutomaticTracking_t, 745
pbEnabled
unpack_pds_GetXTRAAutomaticDownload_t, 746
pbPlatform
unpack_dms_GetOfflineReason_t, 574
pci
nas_cellParams, 154
nas_ltePCI, 201
nas_PhyCaAggPcellInfo, 223
nas_PhyCaAggScellIndType, 224
nas_PhyCaAggScellInfo, 227
nas_umtsLTENbrCell, 272
NASPhyCaAggPcellInfo, 285
NASPhyCaAggScellIndType, 287
NASPhyCaAggScellInfo, 288
pcsfFQDNAddress
wds_PCSCFFQDNAddressList, 990
PdpConnState
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, 638
PdpStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
pdpType
pack_wds_SetDefaultProfile_t, 465
pdptype
unpack_wds_GetDefaultProfile_t, 885
pds.h, 1182
pack_pds_ForceXTRADownload, 1185
pack_pds_GetPDSDefaults, 1186
pack_pds_GetPDSState, 1186
pack_pds_GetPortAutomaticTracking, 1186
pack_pds_GetServiceAutomaticTracking, 1187
pack_pds_GetXTRAAutomaticDownload, 1187
pack_pds_GetXTRANetwork, 1187
pack_pds_GetXTRAValidity, 1188
pack_pds_PDSInjectTimeReference, 1188
pack_pds_ResetPDSData, 1188
pack_pds_SLQSGetAGPSConfig, 1191
pack_pds_SLQSGetGPSStateInfo, 1192
pack_pds_SLQSPDSDeterminePosition, 1192
pack_pds_SLQSPDSInjectAbsoluteTimeReference, 1192
pack_pds_SLQSPDSInjectPositionData, 1193
pack_pds_SLQSSetAGPSConfig, 1193
pack_pds_SLQSSetPositionMethodState, 1194
pack_pds_SetEventReportCallback, 1189
pack_pds_SetPDSDefaults, 1189
pack_pds_SetPDSState, 1190
pack_pds_SetPortAutomaticTracking, 1190
pack_pds_SetServiceAutomaticTracking, 1190
pack_pds_SetXTRAAutomaticDownload, 1191
pack_pds_SetXTRANetwork, 1191
pack_pds_StartPDSTrackingSessionExt, 1194
pack_pds_StopPDSTrackingSession, 1194
unpack_pds_ForceXTRADownload, 1195
unpack_pds_ForceXTRADownload_t, 1185
unpack_pds_GetPDSDefaults, 1195
unpack_pds_GetPDSState, 1195
unpack_pds_GetPortAutomaticTracking, 1196
unpack_pds_GetServiceAutomaticTracking, 1196
unpack_pds_GetXTRAAutomaticDownload, 1196
unpack_pds_GetXTRANetwork, 1197
unpack_pds_GetXTRAValidity, 1197
unpack_pds_PDSInjectTimeReference, 1198
unpack_pds_PDSInjectTimeReference_t, 1185
unpack_pds_ResetPDSData, 1198
unpack_pds_ResetPDSData_t, 1185
unpack_pds_SLQSGetAGPSConfig, 1201
unpack_pds_SLQSGetGPSStateInfo, 1202
unpack_pds_SLQSPDSDeterminePosition, 1202
unpack_pds_SLQSPDSDeterminePosition_t, 1185
unpack_pds_SLQSPDSInjectAbsoluteTime-Reference, 1202
unpack_pds_SLQSPDSInjectAbsoluteTime-Reference_t, 1185
unpack_pds_SLQSPDSInjectPositionData, 1203
unpack_pds_SLQSPDSInjectPositionData_t, 1185
unpack_pds_SLQSSetAGPSConfig, 1203
unpack_pds_SLQSSetAGPSConfig_t, 1185
unpack_pds_SLQSSetPositionMethodState, 1203
unpack_pds_SLQSSetPositionMethodState_t, 1185
unpack_pds_SetEventReport_Ind, 1198
unpack_pds_SetEventReportCallback, 1199
unpack_pds_SetEventReportCallback_t, 1185

unpack_pds_SetPDSDefaults, 1199
unpack_pds_SetPDSDefaults_t, 1185
unpack_pds_SetPDSState, 1199
unpack_pds_SetPDSState_t, 1185
unpack_pds_SetPdsState_Ind, 1200
unpack_pds_SetPortAutomaticTracking, 1200
unpack_pds_SetPortAutomaticTracking_t, 1185
unpack_pds_SetServiceAutomaticTracking, 1200
unpack_pds_SetServiceAutomaticTracking_t, 1185
unpack_pds_SetXTRAAutomaticDownload, 1201
unpack_pds_SetXTRAAutomaticDownload_t, 1185
unpack_pds_SetXTRANetwork, 1201
unpack_pds_SetXTRANetwork_t, 1185
unpack_pds_StartPDSTrackingSessionExt, 1204
unpack_pds_StartPDSTrackingSessionExt_t, 1185
unpack_pds_StopPDSTrackingSession, 1204
unpack_pds_StopPDSTrackingSession_t, 1185
peakRate
unpack_qos_tokenBucket_t, 777
peakThroughputClass
LibPackGPRSRequestedQoS, 91
wds_GPRSQoS, 987
PeriodicUpdateTimer
unpack_dms_PSMCfgChange_ind_t, 581
periodicUpdateTimer
dms_PSMPeriodicUpdateTimerTlv, 58
PeriodicUpdateTimerInd
dms_PSMPeriodicUpdateTimerIndTlv, 57
PersistentLpm
unpack_dms_SLQSDmsSwiGetPCInfo_t, 590
PersistentTechPref
unpack_nas_GetNetworkPreference_t, 679
persoFeature
appStats, 37
uim_appStatus, 524
persoRetries
appStats, 37
uim_appStatus, 524
persoState
appStats, 37
uim_appStatus, 524
persoUnblockRetries
appStats, 37
uim_appStatus, 524
pfailureCause
unpack_voice_SLQSOriginateUSSD_t, 844
PhCtxtURI
ims_PhCtxtURIInfo, 76
phase
nas_rxInfo, 249
PhyCaAggPcellInfo
NasGetLTECphyCaInfo, 281
PhyCaAggScellDlBw
NasGetLTECphyCaInfo, 281
PhyCaAggScellIndType
NasGetLTECphyCaInfo, 281
PhyCaAggScellIndex
NasGetLTECphyCaInfo, 281
PhyCaAggScellInfo
NasGetLTECphyCaInfo, 281
PhysicalLayer
nas_protocolSubtypeElement, 232
PilotEnergy
nas_NetworkStatEVDO, 217
PilotPN
nas_PilotSetParams, 228
PilotStrength
nas_PilotSetParams, 228
PilotType
nas_PilotSetParams, 228
pin1Len
uim_encryptedPIN1, 531
pin1Retries
appStats, 38
uim_appStatus, 524
pin1State
appStats, 38
uim_appStatus, 524
pin1Val
uim_encryptedPIN1, 531
pin2Retries
appStats, 38
uim_appStatus, 524
pin2State
appStats, 38
uim_appStatus, 524
pinID
uim_changeUIMPIN, 529
uim_setPINProtection, 542
uim_unblockUIMPIN, 545
uim_verifyUIMPIN, 546
pinLen
uim_changeUIMPIN, 529
uim_verifyUIMPIN, 546
pinLength
uim_setPINProtection, 542
pinOperation
uim_setPINProtection, 542
pinProtection
pack_uim_SetPinProtection_t, 431
pack_uim_UnblockPin_t, 438
pinVal
uim_changeUIMPIN, 529
uim_verifyUIMPIN, 546
pinValue
uim_setPINProtection, 542
PkgDescLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
PkgDescription
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
PkgName
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
PkgNameLength

unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
pkgver
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603
PktErrRate
unpack_qos_swiQosFlow_t, 776
plasmaIDString
pack_dms_SLQSSwiSetHostDevInfo_t, 314
unpack_dms_SLQSSwiGetHostDevInfo_t, 606
plmn
nas_GERANInfo, 170
nas_LTEInfoIntrafreq, 194
nas_UMTSInfo, 270
polarityIncluded
voice_lineCtrlInfo, 964
port
loc_IPv4Info, 130
loc_IPv6Info, 131
unpack_qos_Port_t, 759
posDataNmea
pack_pds_SetEventReportCallback_t, 384
posSrc
unpack_pds_SetEventReport_Ind_t, 749
PositionDataNMEA
unpack_pds_SetEventReport_Ind_t, 749
positionSrc
pack_loc_SLQSLOCInjectPosition_t, 340
PowerOffMode
unpack_dms_SLQSDmsSwiGetPCInfo_t, 590
pprofileid3gpp
pack_wds_SLQSStartDataSession_t, 480
pprofileid3gpp2
pack_wds_SLQSStartDataSession_t, 480
prDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
prDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
prPCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
prPCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
Precedence
unpack_qos_swiQosFilter_t, 772
precedenceClass
LibPackGPRSRequestedQoS, 91
wds_GPRSQoS, 987
prefNetwork
unpack_wds_SLQSGetCurrDataSystemStat_t, 898
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
preference
pack_pds_SetXTRANetwork_t, 387
Preferred
nas_QmiNas3GppNetworkInfo, 234
prefixLen
unpack_qos_IPv6Addr_t, 758
presentationInd
voice_ECTNum, 960
voice_remotePartyNum, 972
priCSCFPort
ims_PCSCFPortInfo, 75
priChA
nas_CDMAChannel, 147
priChB
nas_CDMAChannel, 147
priSize
unpack_dms_GetFirmwareRevisions_t, 569
pridns
unpack_wds_GetDefaultProfile_t, 885
pridnsv6
unpack_wds_GetDefaultProfile_t, 885
primaryDNS
pack_wds_SetDefaultProfile_t, 465
PrimaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 906
PrimaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 906
primaryHA
unpack_wds_GetMobileIPProfile_t, 888
privacyPref
pack_voice_SLQSVoiceSetPreferredPrivacy_t, 455
priver
unpack_dms_SLQSSwiGetFirmwareCurr_t, 603
priversion_str
unpack_dms_GetFirmwareInfo_t, 567
Profile
pack_audio_SLQSGetAudioPathConfig_t, 295
pack_audio_SLQSGetAudioVolTLBConfig_t, 296
pack_audio_SLQSSetAudioPathConfig_t, 298
pack_audio_SLQSSetAudioProfile_t, 299
pack_audio_SLQSSetAudioVolTLBConfig_t, 300
pack_swiaudio_SLQSGetM2MAudioVolume_t, 407
pack_swiaudio_SLQSGetM2MAVMute_t, 408
pack_swiaudio_SLQSGetM2MSpkrGain_t, 408
pack_swiaudio_SLQSSetM2MAudioAVCFG_t, 409
pack_swiaudio_SLQSSetM2MAudioProfile_t, 410
pack_swiaudio_SLQSSetM2MAudioVolume_t, 411
pack_swiaudio_SLQSSetM2MAVMute_t, 412
pack_swiaudio_SLQSSetM2MSpkrGain_t, 412
unpack_audio_SLQSGetAudioProfile_t, 550
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 802
ProfileID
unpack_wds_SLQSGetRuntimeSettings_t, 906
ProfileId
pack_wds_SLQSGetProfileSettings_t, 473
profileId
wds_DHCPProfileIdTlv, 983
wds_DHCPv4ProfileId, 985
wdsDhcpv4ProfileId, 999

ProfileId3GPP2
unpack_qos_swiQosFlow_t, 776
ProfileIdTlv
unpack_wds_DHCPv4ClientLease_ind_t, 880
ProfileIndex
PackCreateProfileOut, 484
profileIndex
pack_wds_SLQSDeleteProfile_t, 470
wds_ProfileIdentifier, 992
profileList
unpack_wds_SLQSGet3GPPConfigItem_t, 898
ProfileName
unpack_wds_SLQSGetRuntimeSettings_t, 906
ProfileType
pack_wds_SLQSGetProfileSettings_t, 473
PackCreateProfileOut, 484
unpack_wds_SLQSGetProfileSettings_t, 903
profileType
pack_wds_SetDefaultProfile_t, 465
pack_wds_SLQSDeleteProfile_t, 470
wds_DHCPProfileIdTlv, 983
wds_DHCPv4ProfileId, 985
wds_ProfileIdentifier, 992
wdsDhcpv4ProfileId, 999
profiletype
pack_wds_GetDefaultProfile_t, 459
PromptFwDownload
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
PromptFwUpdate
pack_swiavms_SLQSAVMSSetSettings_t, 415
pack_swiavms_SLQSAVMSSetSettings_v2_t, 417
Protocol
unpack_nas_GetCDMANetworkParameters_t, 674
ProvisionStatus
voice_CLIPResp, 951
voice_CLIRResp, 952
voice_CNAPResp, 953
voice_COLPResp, 953
voice_COLRResp, 954
psAttachState
nas_servSystem, 254
NASServingSystemInfo, 293
psBarStatus
nas_CallBarringSysInfo, 145
nas_callBarStatus, 146
psState
nas_CommInfo, 158
psc
nas_UMTSInfo, 270
nas_wcdmaCellInfo, 274
nas_WCDMASysInfo, 279
pscValid
nas_WCDMASysInfo, 279
pscsfIPv4Addr
wds_PCSCFIPv4ServerAddressList, 991
psetting
unpack_wds_GetAutoconnect_t, 880
psid
pack_wds_SLQSStopDataSession_t, 481
unpack_wds_SLQSStartDataSession_t, 916
puk1Retries
appStats, 38
uim_appStatus, 524
puk2Retries
appStats, 38
uim_appStatus, 524
pukLen
uim_unblockUIMPIN, 545
pukVal
uim_unblockUIMPIN, 546
pukValue
pack_dms_UIMUnblockPIN_t, 319
pwrDenialTime
voice_lineCtrlInfo, 964

QMI_LITE_WDS_CURRENT_CALL_DB_MASK
wds.h, 1394
QMI_LITE_WDS_LAST_CALL_DB_MASK
wds.h, 1394
QCI
LibPackQosClassID, 115
QFlowState
unpack_qos_QosFlowInfo_t, 761
QLIC
nas_DeviceConfigDetail, 165
QMI pack/unpack (lite-qmi), 29
qaGobiApiTableBandClasses.h, 1204
qaGobiApiTableCallControlReturnReasons.h, 1207
qaGobiApiTableCallEndReasons.h, 1208
qaGobiApiTableCarrierCodes.h, 1225
qaGobiApiTableCodingScheme.h, 1227
qaGobiApiTableGpsCapabilityCodes.h, 1229
qaGobiApiTablePowerModes.h, 1230
qaGobiApiTableRadioInterfaces.h, 1231
qaGobiApiTableRegionCodes.h, 1231
qaGobiApiTableServiceOptions.h, 1232
qaGobiApiTableSupServiceInfoClasses.h, 1234
qaGobiApiTableSwiAudio.h, 1234
qaGobiApiTableSwiOMADMSessionStatus.h, 1235
qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 1236
qaGobiApiTableVoiceCallEndReasons.h, 1237
qm_wds_ds_profile_extended_err_codes
qmerrno.h, 1250
qmerrno.h
eQCWWAN_ERR_API_MUTEX_TIMEOUT, 1247
eQCWWAN_ERR_BUFFER_SZ, 1246
eQCWWAN_ERR_CANCEL_OP, 1247
eQCWWAN_ERR_DRIVER, 1247
eQCWWAN_ERR_ENUM_BEGIN, 1246
eQCWWAN_ERR_ENUM_END, 1247
eQCWWAN_ERR_FILE_COPY, 1246
eQCWWAN_ERR_FILE_OPEN, 1246
eQCWWAN_ERR_GENERAL, 1246
eQCWWAN_ERR_INTERNAL, 1246
eQCWWAN_ERR_INVALID_ARG, 1246

eQCWWAN_ERR_INVALID_DEVID, 1246
eQCWWAN_ERR_INVALID_FILE, 1246
eQCWWAN_ERR_INVALID_QMI_RSP, 1246
eQCWWAN_ERR_INVALID_XID, 1247
eQCWWAN_ERR_MALFORMED_QMI_RSP, 1246
eQCWWAN_ERR_MEMORY, 1246
eQCWWAN_ERR_MULTIPLE_DEVICES, 1247
eQCWWAN_ERR_MULTIPLE_SMS_UNSUPPORTED, 1247
eQCWWAN_ERR_NO_CANCELABLE_OP, 1247
eQCWWAN_ERR_NO_CONNECTION, 1246
eQCWWAN_ERR_NO_DEVICE, 1246
eQCWWAN_ERR_NO_SIGNAL, 1247
eQCWWAN_ERR_NONE, 1246
eQCWWAN_ERR_NULL_TLV, 1250
eQCWWAN_ERR_OFFLINE, 1247
eQCWWAN_ERR_PDU_GENERATION, 1247
eQCWWAN_ERR_QMI_ABORTED, 1247
eQCWWAN_ERR_QMI_ACCESS_DENIED, 1249
eQCWWAN_ERR_QMI_ACK_NOT_SENT, 1249
eQCWWAN_ERR_QMI_ARG_TOO_LONG, 1247
eQCWWAN_ERR_QMI_AUTHENTICATION_FAILED, 1248
eQCWWAN_ERR_QMI_AUTHENTICATION_LOCK, 1248
eQCWWAN_ERR_QMI_BUNDLING_NOT_SUPPORTED, 1249
eQCWWAN_ERR_QMI_CALL_FAILED, 1247
eQCWWAN_ERR_QMI_CARD_BUSY_RSP, 1250
eQCWWAN_ERR_QMI_CARD_CALL_CONTROL_FAILED, 1249
eQCWWAN_ERR_QMI_CAT_END, 1250
eQCWWAN_ERR_QMI_CAT_START, 1250
eQCWWAN_ERR_QMI_CAUSE_CODE, 1248
eQCWWAN_ERR_QMI_CLIENT_IDS_EXHAUSTED, 1247
eQCWWAN_ERR_QMI_CONNECT, 1246
eQCWWAN_ERR_QMI_DEVICE_IN_USE, 1247
eQCWWAN_ERR_QMI_DEVICE_NOT_READY, 1248
eQCWWAN_ERR_QMI_DEVICE_STORAGE_FULL, 1248
eQCWWAN_ERR_QMI_DISABLED, 1248
eQCWWAN_ERR_QMI_ENCODING, 1248
eQCWWAN_ERR_QMI_ENVELOPE_CMD_FAILURE, 1250
eQCWWAN_ERR_QMI_EVENT_REG_FAILED, 1250
eQCWWAN_ERR_QMI_EXTENDED_INTERNAL, 1249
eQCWWAN_ERR_QMI_FDN_RESTRICT, 1249
eQCWWAN_ERR_QMI_FLOW_SUSPENDED, 1248
eQCWWAN_ERR_QMI_GENERAL, 1248
eQCWWAN_ERR_QMI_HARDWARE_RESTRICTED, 1249
eQCWWAN_ERR_QMI_IFACE, 1246
eQCWWAN_ERR_QMI_INCOMPATIBLE_STATE, 1249
eQCWWAN_ERR_QMI_INCORRECT_FLOW_FILTER, 1248
eQCWWAN_ERR_QMI_INCORRECT_PIN, 1247
eQCWWAN_ERR_QMI_INFO_UNAVAILABLE, 1249
eQCWWAN_ERR_QMI_INJECT_TIMEOUT, 1249
eQCWWAN_ERR_QMI_INSUFFICIENT_RESOURCES, 1248
eQCWWAN_ERR_QMI_INTERFACE_NOT_FOUND, 1248
eQCWWAN_ERR_QMI_INTERNAL, 1247
eQCWWAN_ERR_QMI_INVALID_ARG, 1248
eQCWWAN_ERR_QMI_INVALID_CLIENT_ID, 1247
eQCWWAN_ERR_QMI_INVALID_DATA_FORMAT, 1248
eQCWWAN_ERR_QMI_INVALID_ENVELOPE_CMD, 1250
eQCWWAN_ERR_QMI_INVALID_HANDLE, 1247
eQCWWAN_ERR_QMI_INVALID_ID, 1248
eQCWWAN_ERR_QMI_INVALID_INDEX, 1248
eQCWWAN_ERR_QMI_INVALID_IP_FAMILY_PREF, 1248
eQCWWAN_ERR_QMI_INVALID_MCAST_HANDLE, 1248
eQCWWAN_ERR_QMI_INVALID_MESSAGE_ID, 1248
eQCWWAN_ERR_QMI_INVALID_OPERATION, 1249
eQCWWAN_ERR_QMI_INVALID_PDP_TYPE, 1247
eQCWWAN_ERR_QMI_INVALID_PINID, 1247
eQCWWAN_ERR_QMI_INVALID_PROFILE, 1247
eQCWWAN_ERR_QMI_INVALID_PROFILE_TYPE, 1247
eQCWWAN_ERR_QMI_INVALID_PS_ATTACH_ACTION, 1248
eQCWWAN_ERR_QMI_INVALID_QMI_CMD, 1249
eQCWWAN_ERR_QMI_INVALID_QOS_ID, 1248
eQCWWAN_ERR_QMI_INVALID_REGISTER_ACTION, 1248
eQCWWAN_ERR_QMI_INVALID_SERVICE_TYPE, 1247
eQCWWAN_ERR_QMI_INVALID_TECH_PREF, 1247
eQCWWAN_ERR_QMI_INVALID_TERMINAL_RSP, 1250
eQCWWAN_ERR_QMI_INVALID_TRANSITION, 1248
eQCWWAN_ERR_QMI_INVALID_TX_ID, 1247
eQCWWAN_ERR_QMI_MALFORMED_MSG, 1247
eQCWWAN_ERR_QMI_MAX, 1249
eQCWWAN_ERR_QMI_MAX_MCAST_REQUESTS_IN_USE, 1248

eQCWWAN_ERR_QMI_MAX_QOS_REQUESTS_IN_USE, 1248
eQCWWAN_ERR_QMI_MESSAGE_DELIVERY_FAILURE, 1248
eQCWWAN_ERR_QMI_MESSAGE_NOT_SENT, 1248
eQCWWAN_ERR_QMI_MISSING_ARG, 1247
eQCWWAN_ERR_QMI_MSG_BLOCKED, 1249
eQCWWAN_ERR_QMI_NETWORK_ABORTED, 1249
eQCWWAN_ERR_QMI_NETWORK_NOT_READY, 1248
eQCWWAN_ERR_QMI_NETWORK_QOS_UNAWARE, 1248
eQCWWAN_ERR_QMI_NO_EFFECT, 1247
eQCWWAN_ERR_QMI_NO_ENTRY, 1248
eQCWWAN_ERR_QMI_NO_FREE_PROFILE, 1247
eQCWWAN_ERR_QMI_NO_MEMORY, 1247
eQCWWAN_ERR_QMI_NO_NETWORK_FOUND, 1247
eQCWWAN_ERR_QMI_NO_RADIO, 1249
eQCWWAN_ERR_QMI_NO_SUBSCRIPTION, 1249
eQCWWAN_ERR_QMI_NO_THRESHOLDS, 1247
eQCWWAN_ERR_QMI_NOT_A_MCAST_IFACE, 1248
eQCWWAN_ERR_QMI_NOT_PROVISIONED, 1247
eQCWWAN_ERR_QMI_NOT_SUPPORTED, 1249
eQCWWAN_ERR_QMI_OFFSET, 1247
eQCWWAN_ERR_QMI_OP_DEVICE_UNSUPPORTED, 1247
eQCWWAN_ERR_QMI_OP_NETWORK_UNSUPPORTED, 1247
eQCWWAN_ERR_QMI_OP_PARTIAL_FAILURE, 1249
eQCWWAN_ERR_QMI_OUT_OF_CALL, 1247
eQCWWAN_ERR_QMI_PIN_BLOCKED, 1248
eQCWWAN_ERR_QMI_PIN_PERM_BLOCKED, 1248
eQCWWAN_ERR_QMI_POLICY_MISMATCH, 1249
eQCWWAN_ERR_QMI_REQ, 1246
eQCWWAN_ERR_QMI_REQ_SCH, 1246
eQCWWAN_ERR_QMI_REQ_TO, 1246
eQCWWAN_ERR_QMI_REQUESTED_NUM_UNSUPPORTED, 1248
eQCWWAN_ERR_QMI_RSP, 1246
eQCWWAN_ERR_QMI_RSP_TO, 1246
eQCWWAN_ERR_QMI_SEGMENT_ORDER, 1249
eQCWWAN_ERR_QMI_SEGMENT_TOO_LONG, 1249
eQCWWAN_ERR_QMI_SESSION_INACTIVE, 1248
eQCWWAN_ERR_QMI_SESSION_INVALID, 1248
eQCWWAN_ERR_QMI_SESSION_OWNERSHIP, 1248
eQCWWAN_ERR_QMI_SIM_FILE_NOT_FOUND, 1249
eQCWWAN_ERR_QMI_SIM_NOT_INITIALIZED, 1248
eQCWWAN_ERR_QMI_SMSC_ADDR, 1249
eQCWWAN_ERR_QMI_SUPS_FAILURE_CAUSE, 1249
eQCWWAN_ERR_QMI_TPDU_TYPE, 1249
eQCWWAN_ERR_QMI_UNABORTABLE_TRANSACTION, 1247
eQCWWAN_ERR_QMI_UNKNOWN, 1248
eQCWWAN_ERR_QMI_WIDTH, 1250
eQCWWAN_ERR_RESET, 1247
eQCWWAN_ERR_SWICM_AM_VERS_ERROR, 1249
eQCWWAN_ERR_SWICM_CALL_IN_PROGRESS, 1249
eQCWWAN_ERR_SWICM_END, 1250
eQCWWAN_ERR_SWICM_FAILED_TO_KILL_SDK_PROCESS, 1249
eQCWWAN_ERR_SWICM_INVALID_SESSION_ID, 1249
eQCWWAN_ERR_SWICM_INVALID_V4_SESSION_ID, 1249
eQCWWAN_ERR_SWICM_INVALID_V6_SESSION_ID, 1249
eQCWWAN_ERR_SWICM_NOT_IMPLEMENTED, 1249
eQCWWAN_ERR_SWICM_QMI_CLNT_NOT_SUPPORTED, 1249
eQCWWAN_ERR_SWICM_QMI_SVC_NOT_SUPPORTED, 1249
eQCWWAN_ERR_SWICM_SM_NO_AVAILABLE_SESSIONS, 1249
eQCWWAN_ERR_SWICM_SOCKET_IN_USE, 1249
eQCWWAN_ERR_SWICM_START, 1249
eQCWWAN_ERR_SWICM_TIMEOUT, 1249
eQCWWAN_ERR_SWICM_V4DWN_V6DWN, 1249
eQCWWAN_ERR_SWICM_V4DWN_V6UP, 1249
eQCWWAN_ERR_SWICM_V4UP_V6DWN, 1249
eQCWWAN_ERR_SWICM_V4UP_V6UP, 1249
eQCWWAN_ERR_SWIDCS_APP_DISCONNECTED, 1250
eQCWWAN_ERR_SWIDCS_DEVNODE_NOT_FOUND, 1250
eQCWWAN_ERR_SWIDCS_END, 1250
eQCWWAN_ERR_SWIDCS_FILEIO_ERR, 1250
eQCWWAN_ERR_SWIDCS_IOCTL_ERR, 1250
eQCWWAN_ERR_SWIDCS_START, 1250
eQCWWAN_ERR_SWIIM_CORRUPTED_FW_IMAGE, 1250
eQCWWAN_ERR_SWIIM_END, 1250

eQCWWAN_ERR_SWIIM_FILE_NOT_FOUND, 1250
eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED, 1250
eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE, 1250
eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE, 1250
eQCWWAN_ERR_SWIIM_FW_INVALID_SLOT_INDEX, 1250
eQCWWAN_ERR_SWIIM_FW_PREFERENCE_MISMATCH, 1250
eQCWWAN_ERR_SWIIM_FW_SAME_AS_CURRENT_ACTIVE_IMAGE, 1250
eQCWWAN_ERR_SWIIM_FW_SECBOOT_IMAGE_NOT_SIGNED, 1250
eQCWWAN_ERR_SWIIM_FW_UPDATE_FAIL, 1250
eQCWWAN_ERR_SWIIM_FW_UPDATE_SUCCESS, 1250
eQCWWAN_ERR_SWIIM_FW_WAIT_FOR_REBOOT, 1250
eQCWWAN_ERR_SWIIM_INVALID_CRASH_STATE, 1250
eQCWWAN_ERR_SWIIM_INVALID_PATH, 1250
eQCWWAN_ERR_SWIIM_OPENING_DIR, 1250
eQCWWAN_ERR_SWIIM_OPENING_FILE, 1250
eQCWWAN_ERR_SWIIM_START, 1250
eQCWWAN_ERR_SWISM_END, 1250
eQCWWAN_ERR_SWISMS_BEARER_DATA_NOT_FOUND, 1250
eQCWWAN_ERR_SWISMS_MSG_CORRUPTED, 1250
eQCWWAN_ERR_SWISMS_MSG_LEN_TOO_LONG, 1250
eQCWWAN_ERR_SWISMS_SMSC_NUM_CORRUPTED, 1250
eQCWWAN_ERR_SWISMS_START, 1250
eWDS_ERR_PROFILE_REG_3GPP2_ERR_INVALID_IDENT_FOR_PROFILE, 1251
eWDS_ERR_PROFILE_REG_3GPP_ACCESS_ERR, 1251
eWDS_ERR_PROFILE_REG_3GPP_CONTEXT_NOT_DEFINED, 1251
eWDS_ERR_PROFILE_REG_3GPP_ERR_OUT_OF_PROFILES, 1251
eWDS_ERR_PROFILE_REG_3GPP_INVAL_PROFILE_FAMILY, 1251
eWDS_ERR_PROFILE_REG_3GPP_READ_ONLY_FLAG_SET, 1251
eWDS_ERR_PROFILE_REG_3GPP_VALID_FLAG_NOT_SET, 1251
eWDS_ERR_PROFILE_REG_END, 1251
eWDS_ERR_PROFILE_REG_INVAL_PROFILE_FAMILY, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_HNDL, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_IDENT, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_OP, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_NUM, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_PROFILE_TYPE, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_INVAL_SUBS_ID, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_LEN_INVALID, 1251
eWDS_ERR_PROFILE_REG_RESULT_ERR_LIB_NOT_INITED, 1251
eWDS_ERR_PROFILE_REG_RESULT_FAIL, 1251
eWDS_ERR_PROFILE_REG_RESULT_LIST_END, 1251
qmerrno.h, 1244
eQCWWANError, 1246
qm_wds_ds_profile_extended_err_codes, 1250
qmiSmsMessageList, 489
messageIndex, 489
messageTag, 489
qmiWDSDataBearerTechnology, 489
currentNetwork, 490
ratMask, 490
soMask, 490
qos.h, 1251
pack_qos_BindDataPort, 1253
pack_qos_SLQSQosGetNetworkStatus, 1253
pack_qos_SLQSQosSwiReadApnExtraParams, 1254
pack_qos_SLQSQosSwiReadDataStats, 1255
pack_qos_SLQSSetQosEventCallback, 1255
unpack_qos_BindDataPort, 1256
unpack_qos_SLQSQosGetNetworkStatus, 1256
unpack_qos_SLQSQosSwiReadApnExtraParams, 1257
unpack_qos_SLQSQosSwiReadDataStats, 1257
unpack_qos_SLQSSetQosEventCallback, 1258
unpack_qos_SLQSSetQosEventCallback_ind, 1258
unpack_qos_SLQSSetQosNWStatusCallback_ind, 1259
unpack_qos_SLQSSetQosPriEventCallback_ind, 1259
unpack_qos_SLQSSetQosStatusCallback_ind, 1260
qos_BindDataPortMuxID_t, 490
MuxID, 490
qos_BindDataPortPeripheralEndPointID_t, 490
EndPointType, 491
IfaceID, 491
qos_BindDataPortSIODataPort_t, 491
SIODataPort, 491

qosDeliveryOrder
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
qosFlow
unpack_qos_SLQSQosSwiReadDataStats_t, 765
QosFlowInfo
unpack_qos_SLQSSetQosEventCallback_ind_t, 766

RAN
unpack_nas_GetServingNetwork_t, 683
RAT
nas_QmiNas3GppNetworkRAT, 235
RATMask
currNetworkInfo, 52
wds_currNetworkInfo, 979
RBTTlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
RFBandInfoElements, 491
activeBandClass, 492
activeChannel, 492
radioInterface, 492
unpack_nas_GetRFInfo_t, 680
RFTlv
unpack_nas_SetEventReportInd_t, 688
RMAutoConnect
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
RPCause
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 799
RPTlv
NASQmiCbkNasSystemSelPrefInd, 291
RRTlv
unpack_nas_SetEventReportInd_t, 688
RSRPThresListLen
nas_RSRPThresh, 246
RSRQThresListLen
nas_RSRQThresh, 247
RSSIThresListLen
nas_RSSIThresh, 248
RTIDTlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
RTTlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
RX_EC_IO
nas_NetworkStat1x, 215
RX_PWR
nas_NetworkStat1x, 215
nas_NetworkStatEVDO, 217
RXBOTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
RXChan
nas_LTEInfo, 191
rXDroppedCount
unpack_wds_GetPacketStatus_t, 892
rXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 892
rXOkBytesCount
unpack_wds_GetPacketStatus_t, 892
rXPacketErrors
unpack_wds_GetPacketStatus_t, 892
rXPacketOverflows
unpack_wds_GetPacketStatus_t, 892
rXPacketSuccesses
unpack_wds_GetPacketStatus_t, 892
radio
unpack_nas_GetSignalStrengths_t, 684
radio_if
pack_nas_SLQSNasGetTxRxInfo_t, 358
radioIf
nas_ecioListElement, 167
nas_errorRateListElement, 169
nas_rsrqInformation, 246
nas_rxSignalStrengthListElement, 251
RadioIfaces
unpack_dms_GetDeviceCap_t, 560
unpack_dms_GetDeviceCapabilities_t, 562
unpack_nas_GetServingNetwork_t, 683
RadioIfacesSize
unpack_dms_GetDeviceCap_t, 560
unpack_nas_GetServingNetwork_t, 683
radioIfacesSize
unpack_dms_GetDeviceCapabilities_t, 562
radioInterface
nas_RFBandInfoElements, 238
nas_RfBandInfoExtFormatElements, 239
nas_RfBandwidthInfoElements, 241
nas_RfDedicatedBandInfoElements, 243
nas_RFInfoTlv, 244
nas_roamIndList, 245
nas_servSystem, 254
nas_SignalStrengthTlv, 255
nas_timeInfo, 267
RFBandInfoElements, 492
radioInterfaceList
NASServingSystemInfo, 293
radioInterfaceNo
NASServingSystemInfo, 293
radioInterfaceSize
nas_RFInfoTlv, 244
randomizationWindow
dms_PSMRandomizationWindowTlv, 58
range
unpack_qos_Port_t, 759
rankIndicatorTlv
unpack_nas_SLQSSwiRandIndicatorCallback_-Ind_t, 735
rat
nas_CSGID, 159
nas_MNRInfo, 211
ratDisabledMask
nas_RatDisabledMaskTlv, 237
RatHandover
unpack_imsa_SLQSImsaRatStatusCallBack_ind_-t, 639

RatHandoverStatus
imsa_RatHandoverStatusInfo, 85
RatHandoverStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
ratMask
qmiWDSDataBearerTechnology, 490
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
rawHorConfidence
pack_loc_SLQSLOCInjectPosition_t, 340
rawHorUncCircular
pack_loc_SLQSLOCInjectPosition_t, 340
rawLen
uim_fileAttributes, 534
rawValue
uim_fileAttributes, 534
readTransparent
pack_uim_ReadTransparent_t, 430
Reason
pack_voice_SLQSVoiceGetCallForwardingStatus-_t, 447
pack_voice_SLQSVoiceSetCallBarringPassword_t, 453
reason
pack_voice_SLQSVoiceGetCallBarring_t, 446
pack_voice_SLQSVoiceSetSUPSService_t, 457
unpack_qos_SLQSSetQosStatusCallback_ind_t, 769
voice_ccSUPSType, 950
voice_redirNumInfo, 970
receiptAction
sms_routeEntry, 506
reconfigReqd
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 909
recordCount
uim_fileAttributes, 534
recordSize
uim_fileAttributes, 534
RedirPartyNum
voice_arrRedirPartyNum, 937
refData
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
refID
pack_cat_CATSendTerminalResponse_t, 302
refString
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
ReferenceID
cat_AlPhaIdentifierTlv, 45
cat_EventIDDataTlv, 47
refpn
nas_CDMAInfo, 149
refreshComplete
pack_uim_SLQSUIMRefreshComplete_t, 435
refreshEvent
unpack_uim_SLQSUIMRefreshCallback_Ind_t, 834
RefreshMode
cat_RefreshTlv, 48
RefreshStage
cat_RefreshTlv, 48
regAction
pack_nas_SLQSInitiateNetworkRegistration_t, 353
RegForeignNID
unpack_nas_GetCDMANetworkParameters_t, 674
RegForeignSID
unpack_nas_GetCDMANetworkParameters_t, 674
RegHomeSID
unpack_nas_GetCDMANetworkParameters_t, 674
regInd
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 797
regPrd
nas_AddCDMASysInfo, 143
regRefresh
pack_uim_SLQSUIMRefreshRegister_t, 436
regRejectInfoValid
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_WCDMASysInfo, 279
regState
nas_servSystem, 254
RegStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
registerFlag
uim_registerRefresh, 540
RegistrationState
unpack_nas_GetServingNetwork_t, 683
registrationState
NASServingSystemInfo, 293
rejCause
nas_GSMSysInfo, 176
nas_LTESysInfo, 208
nas_WCDMASysInfo, 279
rejectCause
nas_RejectReasonTlv, 237
rejectSrvDomain
nas_GSMSysInfo, 177
nas_LTESysInfo, 208
nas_WCDMASysInfo, 279
reliabilityClass
LibPackGPRSRequestedQoS, 91
wds_GPRSQoS, 987
remPartyNumber
voice_remotePartyNum, 972
RemotePartyName
voice_getAllCallRmtPtyName, 962
RemotePartyNum
voice_getAllCallRmtPtyNum, 962
ReqMitigationLvl
unpack_tmd_SLQSTmdGetMitigationLvl_t, 825
resBerRatio
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
ResCode

unpack_audio_SLQSGetAudioVolTLBConfig_t, 550
unpack_audio_SLQSSetAudioVolTLBConfig_t, 551
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
resetInfoInd
pack_dms_SLQSDmsSwiIndicationRegister_t, 311
resultcode
unpack_swiavms_SLQSAVMSEventReportInd_t, 805
unpack_swiavms_SLQSAVMSGetSettings_t, 808
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 810
unpack_swiavms_SLQSAVMSSendSelection_t, 811
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 812
unpack_swiavms_SLQSAvmsSetEventReport_t, 812
unpack_swiavms_SLQSAVMSSetSettings_t, 812
unpack_swiavms_SLQSAVMSSetSettings_v2_t, 813
unpack_swiavms_SLQSAVMSStartSession_t, 813
unpack_swiavms_SLQSAVMSStopSession_t, 814
RetryCount
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
revPolarity
voice_lineCtrlInfo, 964
revTunneling
unpack_wds_GetMobileIPProfile_t, 888
ReverseMac
nas_protocolSubtypeElement, 232
RfSarState
pack_sar_SLQSSetRfSarState_t, 397
rfbandInfoList
unpack_nas_SLQSNasGetRFInfo_t, 721
RingBkTmr
ims_RngBkTmrInfo, 76
RingTmr
ims_RngTmrInfo, 77
RmTrasnferStaticsReq
pack_wds_RMSetTransferStatistics_t, 462
rmTrasnferStaticsReq, 492
bResetStatistics, 493
ulMask, 493
rms.h, 1261
pack_rms_GetSMSWake, 1261
pack_rms_SetSMSWake, 1261
unpack_rms_GetSMSWake, 1262
unpack_rms_SetSMSWake, 1262
RmtPtyNum
voice_arrRemotePartyNum, 938
roamIndicator
nas_roamIndList, 245
RoamIndicatorVal
unpack_nas_SLQSGetServingSystem_t, 702
roamOrigVoiceSO
voice_prefVoiceSO, 969
RoamPref
nas_RoamPrefTlv, 245
NASRoamPreferenceTlv, 291
roamStatus
nas_sysInfoCommon, 262
roamStatusValid
nas_sysInfoCommon, 262
roamTimerValue
voice_roamTimer, 973
Roaming
nas_QmiNas3GppNetworkInfo, 234
unpack_nas_GetCDMANetworkParameters_t, 674
unpack_nas_GetServingNetwork_t, 683
roaming
unpack_nas_SetRoamingIndicatorCallback_ind_t, 690
RoamingIndicatorList
unpack_nas_SLQSGetServingSystem_t, 702
routeList
sms_setRoutesReq, 510
routeStorage
sms_routeEntry, 506
RptIMSCapability
pack_dms_SetIndicationRegister_t, 309
rptPosData
pack_pds_SetEventReportCallback_t, 384
rptRate
nas_LTESigRptCfg, 203
nas_LTESigRptConfig, 204
rscp
nas_rxInfo, 249
nas_UMTSInfo, 270
tdscdmaSigInfoExt, 519
rsrp
lteSSInfo, 139
nas_cellParams, 154
nas_rxInfo, 249
nas_RxSigInfo, 250
nas_umtsLTENbrCell, 272
rsrplevel
nas_lteRsrpinformation, 201
rsrq
lteSSInfo, 140
nas_cellParams, 154
nas_rsrqInformation, 246
nas_SccRxInfo, 252
nas_umtsLTENbrCell, 272
rsrqDelta
nas_SLQSSignalStrengthsIndReq, 257
rsrqInfo
nas_SLQSSignalStrengthsInformation, 258
unpack_nas_SLQSGetSignalStrength_t, 705
rssi
cdmaSSInfo, 49
hdrSSInfo, 69
lteSSInfo, 140
nas_cellParams, 154

nas_gsmCellInfo, 172
tdscdmaSigInfoExt, 519
unpack_nas_GetSignalStrengths_t, 684
rx_bytes
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
rx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
RxByteOKCnt
wds_RXBytesOKTlv, 992
rxChainIndex
nas_RxSigInfo, 250
RxDropConutTlv
unpack_RMTransferStatistics_ind_t, 780
rxLev
nas_GERANInfo, 170
rxOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 903
RxOkByteCountTlv
unpack_RMTransferStatistics_ind_t, 780
RxOkConutTlv
unpack_RMTransferStatistics_ind_t, 780
rxPower
nas_rxInfo, 249
nas_RxSigInfo, 250
RxQFilter
unpack_qos_QosFlowInfo_t, 761
RxQFlowGranted
unpack_qos_QosFlowInfo_t, 761
rxSignalStrength
nas_rxSignalStrengthListElement, 251
rxSignalStrengthDelta
nas_SLQSSignalStrengthsIndReq, 257
rxSignalStrengthInfo
nas_SLQSSignalStrengthsInformation, 258
rxSignalStrengthList
unpack_nas_SLQSGetSignalStrength_t, 705
rxSignalStrengthListLen
unpack_nas_SLQSGetSignalStrength_t, 705

SCI
unpack_nas_GetCDMANetworkParameters_t, 674
SCM
unpack_nas_GetCDMANetworkParameters_t, 674
SCTlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
SDPTlv
NASQmiCbkNasSystemSelPrefInd, 291
SDTlv
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 634
SDU_HDR_LEN
common.h, 1013
SFTlv
unpack_ims_SLQSSMSCfgCallBack_ind_t, 631
SHORT
SwiDataTypes.h, 1311
SI
voice_calledPartyInfo, 941
voice_callFWExtInfo, 946
voice_callingPartyInfo, 950
voice_redirNumInfo, 970
SINTlv
unpack_ims_SLQSSMSCfgCallBack_ind_t, 631
SIODataPort
qos_BindDataPortSIODataPort_t, 491
SIPLocalPort
ims_SIPPortInfo, 79
sIntraSearch
nas_LTEInfoIntrafreq, 194
SLQSSSTlv
unpack_nas_SetEventReportInd_t, 688
SMS_CONFIG_LEN
sms.h, 1268
SMS_MAX_SMS_ROUTES
sms.h, 1268
SMS_NUM_OF_SET
sms.h, 1268
SMSC_TYPE_LEN
sms.h, 1268
sMSCAddressInfo, 511
data, 511
length, 511
sMSCAddressTlv, 511
SMSCInfo, 511
TlvPresent, 511
SMSCInfo
sMSCAddressTlv, 511
SMSCTlv
unpack_sms_SetNewSMSCallback_ind_t, 785
sMSEtwsMessageInfo, 512
data, 512
length, 512
notificationType, 512
sMSEtwsMessageTlv, 512
EtwsMessageInfo, 513
TlvPresent, 513
sMSEtwsPlmnInfo, 513
mobileCountryCode, 513
mobileNetworkCode, 513
sMSMTMessageInfo, 513
messageIndex, 514
storageType, 514
sMSMessageModeInfo, 513
messageMode, 513
sMSOnIMSInfo, 514
smsOnIMS, 514
sMSOnIMSTlv, 514
IMSInfo, 515
TlvPresent, 515
SMSSupport
pack_dms_SetCustFeature_t, 307
unpack_dms_GetCustFeature_t, 558
sMSTransferRouteMTMessageInfo, 515
ackIndicator, 515
data, 515
format, 516

length, 516
transactionID, 516
SNR
nas_NetworkStatEVDO, 217
sNonIntraSearch
nas_LTEInfoIntrafreq, 194
SO
nas_NetworkStat1x, 215
SOMask
currNetworkInfo, 52
wds_currNetworkInfo, 979
SPC_SIZE
dms.h, 1025
SPTlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
sPhyCaAggPcellInfo
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 689
sPhyCaAggScellDlBw
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 689
sPhyCaAggScellIndType
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 689
sPhyCaAggScellIndex
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 689
sPhyCaAggScellInfo
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 689
SRTlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
SSInfo
unpack_nas_SetServingSystemCallback_ind_t, 691
sSLQSSignalStrengthsInfo
nas_SLQSSignalStrengthsTlv, 259
SSTlv
unpack_nas_SetEventReportInd_t, 688
STTlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
SUPSInformation
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 874
SUPSType
pack_voice_SLQSVoiceManageCalls_t, 451
SWI_API
SwiDataTypes.h, 1310
SWIWWANCMAPI.h, 1332
samplesPerBatch
loc_accelAcceptReady, 121
loc_accelTempAcceptReady, 122
loc_gyroAcceptReady, 128
loc_gyroTempAcceptReady, 128
sar.h, 1263
pack_sar_SLQSGetRfSarState, 1263
pack_sar_SLQSSetRfSarState, 1263
unpack_sar_SLQSGetRfSarState, 1264
unpack_sar_SLQSSetRfSarState, 1264
unpack_sar_SLQSSetRfSarState_t, 1263
sbas_almanac_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
sbas_ephemeris_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
sbas_health_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
sbas_visible_sv_msk
unpack_pds_SLQSGetGPSStateInfo_t, 755
scell_idx
nas_PhyCaAggScellIndex, 224
NASPhyCaAggScellIndex, 286
scell_state
nas_PhyCaAggScellIndType, 224
nas_PhyCaAggScellInfo, 227
NASPhyCaAggScellIndType, 287
NASPhyCaAggScellInfo, 288
scrAmrEnable
ims_EnabSCRAMRInfo, 73
scrAmrWBEnable
ims_EnabSCRAMRWBInfo, 73
screeningInd
voice_connectNumInfo, 956
sduErrorRatio
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
seDNSIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
seDNSIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
sePCSCFIPv4Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
sePCSCFIPv6Address
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 918
secActivate
uim_fileAttributes, 534
secActivateMask
uim_fileAttributes, 534
secChA
nas_CDMAChannel, 147
secChB
nas_CDMAChannel, 147
secDeactivate
uim_fileAttributes, 534
secDeactivateMask
uim_fileAttributes, 534
secIncrease
uim_fileAttributes, 534
secIncreaseMask
uim_fileAttributes, 534
SecProt
nas_protocolSubtypeElement, 232
secRead

uim_fileAttributes, 534
secReadMask
uim_fileAttributes, 534
secWrite
uim_fileAttributes, 534
secWriteMask
uim_fileAttributes, 534
secdns
unpack_wds_GetDefaultProfile_t, 885
secdnsv6
unpack_wds_GetDefaultProfile_t, 885
second
nas_timeInfo, 267
nas_UniversalTime, 273
secondaryDNS
pack_wds_SetDefaultProfile_t, 465
SecondaryDNSV4
unpack_wds_SLQSGetRuntimeSettings_t, 906
SecondaryDNSV6
unpack_wds_SLQSGetRuntimeSettings_t, 906
secondaryHA
unpack_wds_GetMobileIPProfile_t, 888
SectorIDLen
nas_NetworkStatEVDO, 217
selNetwork
nas_servSystem, 254
selected
sms_BroadcastConfig, 498
sms_CDMABroadcastConfig, 498
selectedNetwork
NASServingSystemInfo, 293
selection
pack_swiavms_SLQSAVMSSendSelection_t, 413
pack_swioma_SLQSOMADMSendSelection_t, 422
pack_swioma_SLQSOMADMSendSelectionExt_t, 423
sendStatus
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 799
sensorData_t, 493
flags, 494
sensorDataLen, 494
timeOfFirstSample, 494
timeOffset, 494
xAxis, 494
yAxis, 494
zAxis, 494
sensorDataLen
sensorData_t, 494
ServerAddrList
unpack_wds_SLQSGetRuntimeSettings_t, 906
serverAddrTypeMask
pack_loc_SLQSLOCGetServer_t, 335
serverStatus
unpack_loc_GetServer_Ind_t, 654
unpack_loc_SetServer_Ind_t, 669
serverType
pack_loc_SLQSLOCGetServer_t, 335
pack_loc_SLQSLOCSetServer_t, 344
unpack_loc_GetServer_Ind_t, 654
serviceCategory
sms_CDMABroadcastConfig, 498
serviceDomain
nas_RejectReasonTlv, 238
ServiceStatusConfig
pack_imsa_SLQSRegisterIMSAIndication_t, 329
servingCellId
nas_LTEInfoIntrafreq, 194
ServingSystem
unpack_nas_SLQSGetServingSystem_t, 702
sessExp
ims_SessDurInfo, 78
sessionControl
pack_pds_StartPDSTrackingSessionExt_t, 393
sessionEndReason
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 909
SessionId
pack_loc_Start_t, 346
pack_loc_Stop_t, 346
sessionId
unpack_loc_PositionRpt_Ind_t, 665
sessionInfo
pack_uim_ChangePin_t, 429
pack_uim_ReadTransparent_t, 430
pack_uim_SetPinProtection_t, 431
pack_uim_SLQSUIMAuthenticate_t, 431
pack_uim_SLQSUIMGetFileAttributes_t, 433
pack_uim_SLQSUIMRefreshComplete_t, 435
pack_uim_SLQSUIMRefreshGetLastEvent_t, 435
pack_uim_SLQSUIMRefreshOK_t, 436
pack_uim_SLQSUIMRefreshRegister_t, 437
pack_uim_UnblockPin_t, 438
pack_uim_VerifyPin_t, 439
SessionInfoConfig
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 819
SessionInfoFota
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 819
SessionInfoNotification
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 819
sessionOperation
pack_pds_StartPDSTrackingSessionExt_t, 393
sessionResponse
unpack_swiavms_SLQSAVMSStartSession_t, 814
sessionServerOption
pack_pds_StartPDSTrackingSessionExt_t, 393
SessionState
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
SessionStatus
unpack_pds_SetEventReport_Ind_t, 749
sessionStatus

unpack_loc_PositionRpt_Ind_t, 665
unpack_omaDmNotificationsTlv_t, 742
SessionType
pack_swioma_SLQSOMADMGetSessionInfo_t, 422
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
sessionType
pack_pds_StartPDSTrackingSessionExt_t, 393
pack_swiavms_SLQSAVMSStartSession_t, 417
pack_swiavms_SLQSAVMSStopSession_t, 418
pack_swioma_SLQSOMADMCancelSession_t, 421
pack_swioma_SLQSOMADMCancelSessionExt_t, 421
pack_swioma_SLQSOMADMStartSession_t, 426
pack_swioma_SLQSOMADMStartSessionExt_t, 426
uim_refreshevent, 539
uim_sessionInformation, 541
uim_UIMSessionInformation, 545
unpack_omaDmFotaTlv_t, 741
set_fix_rate
pack_swiloc_SwiLocSetAutoStart_t, 420
set_fix_type
pack_swiloc_SwiLocSetAutoStart_t, 420
set_function
pack_swiloc_SwiLocSetAutoStart_t, 420
set_max_dist
pack_swiloc_SwiLocSetAutoStart_t, 420
set_max_time
pack_swiloc_SwiLocSetAutoStart_t, 420
SetupEventList
cat_EventListTlv, 48
Severity
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
severity
unpack_omaDmFotaTlv_t, 741
shortName
nas_PLMNNetworkNameData, 230
unpack_nas_SLQSGetPLMNName_t, 697
shortNameCI
unpack_nas_SLQSGetPLMNName_t, 697
shortNameEn
unpack_nas_SLQSGetPLMNName_t, 697
shortNameLen
nas_PLMNNetworkNameData, 230
unpack_nas_SLQSGetPLMNName_t, 697
shortNameSB
unpack_nas_SLQSGetPLMNName_t, 697
shortNameSpareBits
nas_PLMNNetworkNameData, 230
sid
nas_CDMAInfo, 149
nas_sidNid, 254
unpack_nas_GetHomeNetwork_t, 678
SidNid
nas_homeSIDNID, 183
SigCompEn
ims_SigCompEnInfo, 78
SigInd
LibPackUMTSReqQoSSigInd, 120
sigInfo
nas_SccRxInfo, 252
signal
voice_signalInfo, 973
signalStrength
nas_SignalStrengthTlv, 255
signalStrengthReqMask
unpack_nas_SLQSGetSignalStrength_t, 705
signalType
voice_signalInfo, 973
SimCapability
unpack_dms_GetDeviceCap_t, 560
simCapability
unpack_dms_GetDeviceCapabilities_t, 562
sinr
hdrSSInfo, 69
nas_SLQSSignalStrengthsInformation, 258
tdscdmaSigInfoExt, 519
unpack_nas_SLQSGetSignalStrength_t, 705
sinrDelta
nas_SLQSSignalStrengthsIndReq, 257
sinrThresholdList
nas_SLQSSignalStrengthsIndReq, 257
sinrThresholdListLen
nas_SLQSSignalStrengthsIndReq, 257
sku_str
unpack_dms_GetFirmwareInfo_t, 567
slot
pack_uim_SLQSUIMPowerDown_t, 434
pack_uim_SLQSUIMPowerUp_t, 434
slot_t, 494
bICCID, 495
bICCIDLength, 495
bLogicalSlot, 495
uPhyCardStatus, 495
uPhySlotStatus, 495
slotInf, 495
AppStatus, 496
cardState, 496
errorState, 497
numApp, 497
upinRetries, 497
upinState, 497
upukRetries, 497
SlotInfo
uim_cardStatus, 528
slots_t, 497
uimSlotStatus, 497
slotsstatusChange
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 830
SlqsProfile3GPP
unpackWdsProfileParam, 928

wds_profileInfo, 992
SlqsProfile3GPP2
unpackWdsProfileParam, 928
wds_profileInfo, 992
sms.h
LITEQMI_QMI_CBK_PARAM_NOCHANGE, 1269
LITEQMI_QMI_CBK_PARAM_RESET, 1269
LITEQMI_QMI_CBK_PARAM_SET, 1269
sms.h, 1264
eqmiCbkSetStatus, 1268
MAX_MSE_TWS_MSG, 1268
MAX_SMS_LIST_SIZE, 1268
pack_sms_GetSMSCAddress, 1269
pack_sms_SLQSDeleteSMS, 1270
pack_sms_SLQSGetIndicationRegister, 1271
pack_sms_SLQSGetMessageWaiting, 1271
pack_sms_SLQSGetSMS, 1271
pack_sms_SLQSGetSMSList, 1272
pack_sms_SLQSGetSmsBroadcastConfig, 1272
pack_sms_SLQSGetTransLayerInfo, 1273
pack_sms_SLQSGetTransNWRegInfo, 1273
pack_sms_SLQSModifySMSStatus, 1273
pack_sms_SLQSSendAsyncSMS, 1274
pack_sms_SLQSSetIndicationRegister, 1274
pack_sms_SLQSSetSmsBroadcastActivation, 1274
pack_sms_SLQSSetSmsBroadcastConfig, 1275
pack_sms_SLQSSetSmsStorage, 1275
pack_sms_SLQSSmsGetMaxStorageSize, 1275
pack_sms_SLQSSmsGetMessageProtocol, 1276
pack_sms_SLQSSmsSetRoutes, 1276
pack_sms_SLQSSwiGetSMSStorage, 1277
pack_sms_SaveSMS, 1269
pack_sms_SendSMS, 1269
pack_sms_SetNewSMSCallback, 1270
pack_sms_SetSMSCAddress, 1270
SMS_CONFIG_LEN, 1268
SMS_MAX_SMS_ROUTES, 1268
SMS_NUM_OF_SET, 1268
SMSC_TYPE_LEN, 1268
unpack_sms_GetSMSCAddress, 1277
unpack_sms_SLQSDeleteSMS, 1279
unpack_sms_SLQSGetIndicationRegister, 1279
unpack_sms_SLQSGetMessageWaiting, 1280
unpack_sms_SLQSGetSMS, 1280
unpack_sms_SLQSGetSMSList, 1281
unpack_sms_SLQSGetSmsBroadcastConfig, 1280
unpack_sms_SLQSGetTransLayerInfo, 1281
unpack_sms_SLQSGetTransNWRegInfo, 1281
unpack_sms_SLQSModifySMSStatus, 1282
unpack_sms_SLQSNWRegInfoCallback_ind, 1282
unpack_sms_SLQSSendAsyncSMS, 1282
unpack_sms_SLQSSetIndicationRegister, 1283
unpack_sms_SLQSSetSmsBroadcastActivation, 1283
unpack_sms_SLQSSetSmsBroadcastConfig, 1283
unpack_sms_SLQSSetSmsStorage, 1284
unpack_sms_SLQSSmsGetMaxStorageSize, 1284
unpack_sms_SLQSSmsGetMessageProtocol, 1284
unpack_sms_SLQSSmsSetRoutes, 1285
unpack_sms_SLQSSwiGetSMSStorage, 1285
unpack_sms_SLQSTransLayerInfoCallback_ind, 1285
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind, 1286
unpack_sms_SLQSWmsMemoryFullCallBack_ind, 1286
unpack_sms_SLQSWmsMessageWaitingCallBack_ind, 1287
unpack_sms_SaveSMS, 1277
unpack_sms_SendSMS, 1278
unpack_sms_SetNewSMSCallback, 1278
unpack_sms_SetNewSMSCallback_ind, 1278
unpack_sms_SetSMSCAddress, 1279
sms_BroadcastConfig, 497
fromServiceId, 498
selected, 498
toServiceId, 498
sms_CDMABroadcastConfig, 498
language, 498
selected, 498
serviceCategory, 498
sms_getIndicationReg, 498
pRegCallStatInfoEvt, 499
pRegTransLayerInfoEvt, 499
pRegTransNWRegInfoEvt, 499
sms_getMsgWaitingInfo, 499
msgWaitInfo, 500
numInstances, 500
sms_getTransLayerInfo, 500
pRegInd, 500
pTransLayerInfo, 500
sms_getTransNWRegInfo, 500
pRegStatus, 501
sms_maxStorageSizeReq, 501
pMessageMode, 502
storageType, 502
sms_maxStorageSizeResp, 502
freeSlots, 502
maxStorageSize, 502
sms_messageWaitingInfoContent, 502
activeInd, 503
msgCount, 503
msgType, 503
sms_msgProtocolResp, 503
msgProtocol, 503
sms_qaQmi3GPP2BroadcastCfgInfo, 503
activated_ind, 504
CDMABroadcastConfig, 504
num_instances, 504
sms_qaQmi3GPPBroadcastCfgInfo, 504
activated_ind, 505
broadcastConfig, 505
num_instances, 505

sms_routeEntry, 505
messageClass, 506
messageType, 506
receiptAction, 506
routeStorage, 506
sms_sendAsyncsmsParams, 506
messageFormat, 508
messageSize, 508
pFollowOnDC, 508
pForceOnDC, 508
pLinktimer, 508
pMessage, 508
pRetryMessage, 508
pRetryMessageId, 508
pServiceOption, 508
pSmsOnIms, 508
pUserData, 508
sms_setIndicationReg, 508
pRegCallStatInfoEvt, 509
pRegTransLayerInfoEvt, 509
pRegTransNWRegInfoEvt, 509
sms_setRoutesReq, 509
numOfRoutes, 510
pTransferStatusReport, 510
routeList, 510
sms_transLayerInfo, 510
TransCap, 510
TransType, 510
smsFormat
ims_SMSFmtInfo, 80
smsOnIMS
sMSOnIMSInfo, 514
SmsRat
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 641
SmsRatVal
imsa_SmsRatInfo, 86
SmsService
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 641
SmsServiceRat
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
SmsServiceStatus
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
smsStorage
pack_sms_SLQSSetSmsStorage_t, 405
SmsSvcStatus
imsa_SmsSvcStatusInfo, 86
smsoIPNW
ims_SMSoIPNwInfo, 80
snr
loc_satelliteInfo, 135
lteSSInfo, 140
nas_SccRxInfo, 252
snrlevel
nas_lteSnrinformation, 205
soMask
qmiWDSDataBearerTechnology, 490
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
Source
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
source
altSrcInfo_t, 35
unpack_dms_GetNetworkTime_t, 573
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 591
unpack_dms_SLQSDmsSwiGetResetInfo_t, 592
sourceIPMask
LibPackTFTIDParams, 117
SourceLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
SourceRAT
imsa_RatHandoverStatusInfo, 85
spc
pack_dms_ResetToFactoryDefaults_t, 304
pack_nas_SetACCOLC_t, 347
pack_wds_SetMobileIPProfile_t, 469
spn
nas_serviceProviderName, 252
unpack_nas_SLQSGetPLMNName_t, 697
spnEncoding
unpack_nas_SLQSGetPLMNName_t, 697
spnLength
nas_serviceProviderName, 252
unpack_nas_SLQSGetPLMNName_t, 698
srcPortRangeEnd
LibPackTFTIDParams, 117
srcPortRangeStart
LibPackTFTIDParams, 117
srvCapability
nas_detailSvcInfo, 164
nas_sysInfoCommon, 262
srvCapabilityValid
nas_sysInfoCommon, 262
srvDomain
nas_sysInfoCommon, 262
SrvDomainPref
nas_SrvDomainPrefTlv, 259
NASServDomainPrefTlv, 292
srvDomainValid
nas_sysInfoCommon, 262
srvOption
voice_arrSvcOption, 938
srvStatus
nas_detailSvcInfo, 164
nas_GSMSrvStatusInfo, 174
nas_SrvStatusInfo, 260
srxlev
nas_cellParams, 154
nas_gsmCellInfo, 172
nas_umtsLTENbrCell, 272
nas_wcdmaCellInfo, 274
stage

uim_refreshevent, 539
State
nas_NetworkStat1x, 215
nas_NetworkStatEVDO, 217
state
pack_loc_SLQSLOCSetCradleMountConfig_t, 343
unpack_dms_GetActivationState_t, 553
unpack_dms_SLQSUIMGetState_t, 610
unpack_omaDmConfigTlv_t, 739
unpack_omaDmFotaTlv_t, 741
unpack_qos_QosFlowInfoState_t, 761
statmask
pack_wds_GetPacketStatus_t, 462
StatsMask
wds_transferStatInd, 993
statsMask
wds_TrStatInd, 994
StatsPeriod
wds_transferStatInd, 993
statsPeriod
wds_TrStatInd, 994
Status
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
status
nas_lteEARFCN, 188
nas_ltePCI, 201
nas_wcdmaUARFCN, 280
pack_sms_SetNewSMSCallback_t, 399
unpack_loc_BestAvailPos_Ind_t, 648
unpack_loc_DeleteAssistData_Ind_t, 650
unpack_loc_InjectPositionCallback_Ind_t, 655
unpack_loc_InjectUTCTimeCallback_Ind_t, 658
unpack_loc_SetExtPowerConfig_Ind_t, 667
unpack_loc_SetOperationMode_Ind_t, 668
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 767
unpack_qos_SLQSSetQosStatusCallback_ind_t, 769
storageIndex
FMSImageIdElement, 66
storageType
pack_sms_SaveSMS_t, 398
pack_sms_SLQSDeleteSMS_t, 400
pack_sms_SLQSGetSMS_t, 401
pack_sms_SLQSGetSMSList_t, 402
pack_sms_SLQSModifySMSStatus_t, 403
sms_maxStorageSizeReq, 502
sMSMTMessageInfo, 514
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t, 800
Streaming Download Protocol (lite-fw), 30
String
unpack_dms_GetDeviceHardwareRev_t, 562
unpack_dms_GetDeviceMfr_t, 563
unpack_dms_GetFSN_t, 570
unpack_dms_UIMGetICCID_t, 613
stringSize
unpack_dms_GetDeviceHardwareRev_t, 562
unpack_dms_GetDeviceMfr_t, 563
unpack_dms_UIMGetICCID_t, 613
subAddr
voice_calledPartySubAdd, 942
subAddrLen
voice_calledPartySubAdd, 942
subAddrType
voice_calledPartySubAdd, 942
subnetMask
unpack_qos_IPv4Addr_t, 757
SubnetMaskV4
unpack_wds_SLQSGetRuntimeSettings_t, 906
subsType
pack_voice_SLQSVoiceBindSubscription_t, 443
subscrTmr
ims_SubscrTmrInfo, 81
SupUSBComps
unpack_dms_GetUSBComp_t, 579
SuppOA
voice_CUGInfo, 956
SuppPrefCUG
voice_CUGInfo, 956
svInfoMask
loc_satelliteInfo, 135
svListLen
loc_satelliteInfo, 135
svStatus
loc_satelliteInfo, 135
svc
pack_qmi_t, 394
SvcClass
voice_callFWExtInfo, 946
voice_callFWInfo, 947
SvcStatus
voice_callFWExtInfo, 946
voice_callFWInfo, 947
svcType
voice_ccSUPSType, 951
voice_SUPSInfo, 974
sw1
uim_cardResult, 527
sw2
uim_cardResult, 527
swVerString
pack_dms_SLQSSwiSetHostDevInfo_t, 314
unpack_dms_SLQSSwiGetHostDevInfo_t, 606
swi_uint256_clear_bit
switype_256bit.h, 1331
swi_uint256_get_bit
switype_256bit.h, 1331
swi_uint256_print_mask
switype_256bit.h, 1332
swi_uint256_set_bit
switype_256bit.h, 1332
swi_uint256_t, 516
word, 516
SwiDataTypes.h, 1310

BOOL, 1311
BYTE, 1311
CHAR, 1311
FLOAT, 1311
INT32, 1311
INT8, 1311
LPCSTR, 1311
SHORT, 1311
SWI_API, 1310
ULONG, 1311
ULONGLONG, 1311
UNUSEDPARAM, 1310
USHORT, 1311
WORD, 1311
swiaudio.h, 1287
pack_swiaudio_SLQSGetM2MAVMute, 1289
pack_swiaudio_SLQSGetM2MAudioProfile, 1289
pack_swiaudio_SLQSGetM2MAudioVolume, 1289
pack_swiaudio_SLQSGetM2MSpkrGain, 1290
pack_swiaudio_SLQSSetM2MAVMute, 1292
pack_swiaudio_SLQSSetM2MAudioAVCFG, 1290
pack_swiaudio_SLQSSetM2MAudioLPBK, 1290
pack_swiaudio_SLQSSetM2MAudioNVDef, 1291
pack_swiaudio_SLQSSetM2MAudioProfile, 1291
pack_swiaudio_SLQSSetM2MAudioVolume, 1292
pack_swiaudio_SLQSSetM2MSpkrGain, 1292
unpack_swiaudio_SLQSGetM2MAVMute, 1293
unpack_swiaudio_SLQSGetM2MAudioProfile, 1293
unpack_swiaudio_SLQSGetM2MAudioVolume, 1293
unpack_swiaudio_SLQSGetM2MSpkrGain, 1294
unpack_swiaudio_SLQSSetM2MAVMute, 1296
unpack_swiaudio_SLQSSetM2MAVMute_t, 1289
unpack_swiaudio_SLQSSetM2MAudioAVCFG, 1294
unpack_swiaudio_SLQSSetM2MAudioAVCFG_t, 1288
unpack_swiaudio_SLQSSetM2MAudioLPBK, 1294
unpack_swiaudio_SLQSSetM2MAudioLPBK_t, 1288
unpack_swiaudio_SLQSSetM2MAudioNVDef, 1295
unpack_swiaudio_SLQSSetM2MAudioNVDef_t, 1288
unpack_swiaudio_SLQSSetM2MAudioProfile, 1295
unpack_swiaudio_SLQSSetM2MAudioProfile_t, 1289
unpack_swiaudio_SLQSSetM2MAudioVolume, 1295
unpack_swiaudio_SLQSSetM2MAudioVolume_t, 1289
unpack_swiaudio_SLQSSetM2MSpkrGain, 1296
unpack_swiaudio_SLQSSetM2MSpkrGain_t, 1289
swiaudio_PCMparams, 516
iFaceTab, 516
iFaceTabLen, 516
swiavms.h, 1296
pack_swiavms_SLQSAVMSGetSettings, 1299
pack_swiavms_SLQSAVMSGetSettings_v2, 1299
pack_swiavms_SLQSAVMSSendSelection, 1300
pack_swiavms_SLQSAVMSSessionGetInfo, 1300
pack_swiavms_SLQSAVMSSetSettings, 1301
pack_swiavms_SLQSAVMSSetSettings_v2, 1302
pack_swiavms_SLQSAVMSSetSettingsNoAutoRebootField, 1303
pack_swiavms_SLQSAVMSStartSession, 1303
pack_swiavms_SLQSAVMSStopSession, 1304
pack_swiavms_SLQSAvmsSetEventReport, 1301
unpack_swiavms_SLQSAVMSEventReportInd, 1305
unpack_swiavms_SLQSAVMSGetSettings, 1305
unpack_swiavms_SLQSAVMSGetSettings_v2, 1305
unpack_swiavms_SLQSAVMSSendSelection, 1306
unpack_swiavms_SLQSAVMSSessionGetInfo, 1306
unpack_swiavms_SLQSAVMSSetSettings, 1307
unpack_swiavms_SLQSAVMSSetSettings_v2, 1308
unpack_swiavms_SLQSAVMSStartSession, 1308
unpack_swiavms_SLQSAVMSStopSession, 1309
unpack_swiavms_SLQSAVMSStopSession_avc2, 1309
unpack_swiavms_SLQSAvmsSetEventReport, 1307
swidms.h, 1311
pack_swidms_SLQSSwiDmsGetMTU, 1312
pack_swidms_SLQSSwiDmsGetUsbNetNum, 1312
pack_swidms_SLQSSwiDmsSetMTU, 1312
pack_swidms_SLQSSwiDmsSetUsbNetNum, 1313
unpack_swidms_SLQSSwiDmsGetMTU, 1313
unpack_swidms_SLQSSwiDmsGetUsbNetNum, 1313
unpack_swidms_SLQSSwiDmsSetMTU, 1314
unpack_swidms_SLQSSwiDmsSetUsbNetNum, 1314
unpack_swidms_SLQSSwiDmsSetUsbNetNum_t, 1312
swidms_ehrpdMTUSizeTlv, 516
ehrpdMTUSize, 517
TlvPresent, 517
swidms_hrpdMTUSizeTlv, 517
hrpdMTUSize, 517
TlvPresent, 517
swidms_mtuSize3gppTlv, 517
MTUSize3gpp, 518
TlvPresent, 518
swidms_usbMTUSizeTlv, 518
TlvPresent, 518
UsbMTUSize, 518
swiloc.h, 1315

pack_swiloc_SwiLocGetAutoStart, 1315
pack_swiloc_SwiLocSetAutoStart, 1315
unpack_swiloc_SwiLocGetAutoStart, 1316
unpack_swiloc_SwiLocSetAutoStart, 1316
unpack_swiloc_SwiLocSetAutoStart_t, 1315
swioma.h, 1316
pack_swioma_SLQSOMADMAlertCallback, 1317
pack_swioma_SLQSOMADMCancelSession, 1318
pack_swioma_SLQSOMADMGetSessionInfo, 1319
pack_swioma_SLQSOMADMGetSettings, 1319
pack_swioma_SLQSOMADMSendSelection, 1320
pack_swioma_SLQSOMADMSetSettings, 1320
pack_swioma_SLQSOMADMStartSession, 1321
unpack_swioma_SLQSOMADMAlertCallback, 1322
unpack_swioma_SLQSOMADMAlertCallback_ind, 1322
unpack_swioma_SLQSOMADMCancelSession, 1323
unpack_swioma_SLQSOMADMGetSessionInfo, 1323
unpack_swioma_SLQSOMADMGetSettings, 1323
unpack_swioma_SLQSOMADMSendSelection, 1324
unpack_swioma_SLQSOMADMSetSettings, 1324
unpack_swioma_SLQSOMADMStartSession, 1325
swiomaext.h, 1325
pack_swioma_SLQSOMADMCancelSessionExt, 1326
pack_swioma_SLQSOMADMSendSelectionExt, 1327
pack_swioma_SLQSOMADMSetSettingsExt, 1327
pack_swioma_SLQSOMADMStartSessionExt, 1328
unpack_swioma_SLQSOMADMCancelSessionExt, 1329
unpack_swioma_SLQSOMADMCancelSessionExt_t, 1326
unpack_swioma_SLQSOMADMSendSelectionExt, 1329
unpack_swioma_SLQSOMADMSendSelectionExt_t, 1326
unpack_swioma_SLQSOMADMSetSettingsExt, 1330
unpack_swioma_SLQSOMADMSetSettingsExt_t, 1326
unpack_swioma_SLQSOMADMStartSessionExt, 1330
unpack_swioma_SLQSOMADMStartSessionExt_t, 1326
switchOption
pack_voice_SLQSVoiceALSSetLineSwitching_t, 442
switype_256bit.h, 1331
swi_uint256_clear_bit, 1331
swi_uint256_get_bit, 1331
swi_uint256_print_mask, 1332
swi_uint256_set_bit, 1332
sysInfoCDMA
nas_CDMASysInfo, 153
sysInfoGSM
nas_GSMSysInfo, 177
sysInfoHDR
nas_HDRSysInfo, 183
sysInfoLTE
nas_LTESysInfo, 208
sysInfoWCDMA
nas_WCDMASysInfo, 279
system
loc_satelliteInfo, 135
loc_SV, 137
systemDiscontinuities
pack_pds_PDSInjectTimeReference_t, 383
SystemID
unpack_nas_SLQSGetServingSystem_t, 702
systemID
nas_CDMASysInfo, 153
systemMode
nas_CommInfo, 158
systemTime
pack_pds_PDSInjectTimeReference_t, 383
szAPN
PackSwiAvmsSetSettingsAPNInfo, 486
szAlertMsg
UnpackSwiAvmsEventReportConfig, 922
szCarrier_str
_litefw_FirmwareInfo_, 33
szCarrierPriversion_str
_litefw_FirmwareInfo_, 34
szDescription
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, 920
szFwversion_str
_litefw_FirmwareInfo_, 34
szModelid_str
_litefw_FirmwareInfo_, 34
szName
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, 920
szPWD
PackSwiAvmsSetSettingsAPNInfo, 486
szPackageid_str
_litefw_FirmwareInfo_, 34
szSku_str
_litefw_FirmwareInfo_, 34
szUname
PackSwiAvmsSetSettingsAPNInfo, 486
szVersion
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, 921

t3396_apn
unpack_nas_SLQSNasTimerCallback_ind_t, 728
t3396_plmn_id
unpack_nas_SLQSNasTimerCallback_ind_t, 728

t3396_val
unpack_nas_SLQSNasTimerCallback_ind_t, 728
TCPDstPort
unpack_qos_swiQosFilter_t, 772
TCPSrcPort
unpack_qos_swiQosFilter_t, 772
TDSCDMAECIOThreshListLen
nas_TDSCDMAECIOThresh, 263
TDSCDMARSCPThreshListLen
nas_TDSCDMARSCPThresh, 263
TDSCDMARSSIThreshListLen
nas_TDSCDMARSSIThresh, 264
TDSCDMASINRThreshListLen
nas_TDSCDMASINRThresh, 265
TMD_MAX_DEV_LIST
tmd.h, 1333
TMTlv
unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 627
TPCause
unpack_sms_SLQSWmsAsyncRawSendCallBack-_ind_t, 799
TRMessageTlv
unpack_sms_SetNewSMSCallback_ind_t, 785
TT1Tlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
TT2Tlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
TTfTlv
unpack_ims_SLQSSIPCfgCallBack_ind_t, 630
TX_PWR
nas_NetworkStat1x, 215
TXBOTlv
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 896
TXChan
nas_LTEInfo, 191
tXDroppedCount
unpack_wds_GetPacketStatus_t, 892
tXOKBytesLastCall
unpack_wds_GetPacketStatus_t, 892
tXOkBytesCount
unpack_wds_GetPacketStatus_t, 892
tXPacketErrors
unpack_wds_GetPacketStatus_t, 892
tXPacketOverflows
unpack_wds_GetPacketStatus_t, 892
tXPacketSuccesses
unpack_wds_GetPacketStatus_t, 892
Tables, 31
tac
nas_LTEInfoIntrafreq, 194
nas_LTESysInfo, 208
tacValid
nas_LTESysInfo, 208
TargetRAT
imsa_RatHandoverStatusInfo, 85
TdsBandCapability
unpack_dms_SLQSGetBandCapability_t, 597
tdscdmaSigInfoExt, 518
ecio, 519
rscp, 519
rssi, 519
sinr, 519
techName
unpack_wds_SLQSSetPacketSrvStatusCallback_-t, 909
Technology
nas_DeviceConfigDetail, 165
unpack_wds_SLQSGetRuntimeSettings_t, 906
TechnologyPref
pack_nas_SetNetworkPreference_t, 349
tempData_t, 519
temperature, 520
temperatureDataLen, 520
timeOfFirstSample, 520
timeOffset, 520
timeSource, 520
TempStat
dms_TemperatureTlv, 59
TempTlv
unpack_dms_SwiEventReportCallBack_ind_t, 610
Temperature
dms_TemperatureTlv, 59
temperature
nas_CommInfo, 158
tempData_t, 520
temperatureDataLen
tempData_t, 520
threshGsmHigh
nas_lteGsmCellInfo, 189
threshGsmLow
nas_lteGsmCellInfo, 189
threshServingLow
nas_LTEInfoIntrafreq, 194
threshXHigh
nas_infoInterFreq, 184
threshXLow
nas_infoInterFreq, 185
threshXhigh
nas_lteWcdmaCellInfo, 209
threshXlow
nas_lteWcdmaCellInfo, 209
Time
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
time
NASTimeInfoTlv, 294
Time_uncert_ms
unpack_pds_SLQSGetGPSStateInfo_t, 755
timeBase
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 388
TimeLength
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
timeMsec

pack_loc_SLQSLOCInjectUTCTime_t, 342
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 388
timeOfFirstSample
sensorData_t, 494
tempData_t, 520
timeOffset
sensorData_t, 494
tempData_t, 520
timeSource
tempData_t, 520
TimeStmp_gps_week
unpack_pds_SLQSGetGPSStateInfo_t, 755
TimeStmp_tow_ms
unpack_pds_SLQSGetGPSStateInfo_t, 755
timeSyncRefCounter
unpack_loc_EventTimeSyncCallback_Ind_t, 652
timeTlv
NASQmiCbkNasSwiOTAMessageInd, 289
timeUncMsec
pack_loc_SLQSLOCInjectUTCTime_t, 342
pack_pds_SLQSPDSInjectAbsoluteTimeReference-_t, 389
timeZone
nas_timeInfo, 267
timeout
pack_pds_SetPDSDefaults_t, 385
pack_qmi_t, 394
Timers
PackSwiAvmsSetSettingsConnectionRetryTimers, 486
PackSwiAVMSSettingsConnectionRetryTimers, 488
timestamp
unpack_dms_GetNetworkTime_t, 573
timestampAge
pack_loc_SLQSLOCInjectPosition_t, 340
timestampUtc
pack_loc_SLQSLOCInjectPosition_t, 340
timingAdvance
nas_GERANInfo, 170
TlvPresent
cat_commonEventTlv, 46
dms_ActivationStatusTlv, 52
dms_OperatingModeTlv, 53
dms_PSMActiveTimerIndTlv, 54
dms_PSMActiveTimerTlv, 54
dms_PSMDurationDueToOOSTlv, 55
dms_PSMDurationThresholdTlv, 55
dms_PSMEarlyWakeupTimeTlv, 56
dms_PSMEnableStateIndTlv, 56
dms_PSMEnableStateTlv, 57
dms_PSMPeriodicUpdateTimerIndTlv, 57
dms_PSMPeriodicUpdateTimerTlv, 58
dms_PSMRandomizationWindowTlv, 58
dms_TemperatureTlv, 59
dms_VoltageTlv, 60
eTWSPLMNInfoTlv, 64
ims_AMRModeInfo, 70
ims_AMROctAlgnInfo, 70
ims_AMRWBModeInfo, 71
ims_AMRWBOctAlgnInfo, 72
ims_CSCFPortNameInfo, 72
ims_EnabAMRWBInfo, 72
ims_EnabSCRAMRInfo, 73
ims_EnabSCRAMRWBInfo, 73
ims_IMSDomainInfo, 74
ims_IMSTestModeInfo, 74
ims_MinSessExpInfo, 75
ims_PCSCFPortInfo, 75
ims_PhCtxtURIInfo, 76
ims_RngBkTmrInfo, 76
ims_RngTmrInfo, 77
ims_RTPRTCPInactTmrDurInfo, 77
ims_SessDurInfo, 78
ims_SigCompEnInfo, 78
ims_SIPPortInfo, 79
ims_SIPRegnTmrInfo, 79
ims_SMSFmtInfo, 80
ims_SMSoIPNwInfo, 80
ims_SubscrTmrInfo, 81
ims_TmrT1Info, 81
ims_TmrT2Info, 82
ims_TmrTfInfo, 82
imsa_IMSFailErrCodeTlv, 83
imsa_IMSRegStatusErrorCodeInfo, 83
imsa_IMSRegStatusInfo, 84
imsa_NewIMSRegStatusInfo, 84
imsa_RatHandoverStatusInfo, 85
imsa_SmsRatInfo, 86
imsa_SmsSvcStatusInfo, 86
imsa_UtRatInfo, 87
imsa_UtSvcStatusInfo, 87
imsa_VoipRatInfo, 88
imsa_VoipSvcStatusInfo, 88
imsa_VtRatInfo, 89
imsa_VtSvcStatusInfo, 90
loc_IPv4Info, 130
loc_IPv6Info, 131
loc_urlAddr, 138
messageModeTlv, 140
nas_AcqOrderPrefTlv, 142
nas_BandPrefTlv, 145
nas_CiotAcqOrderPrefTlv, 156
nas_CiotLteOpModePrefTlv, 156
nas_EmerModeTlv, 168
nas_GWAcqOrderPrefTlv, 177
nas_LTEBandPrefTlv, 188
nas_LteM1BandPrefTlv, 198
nas_LteNb1BandPrefTlv, 200
nas_LTEOperationMode, 200
nas_ModePrefTlv, 211
nas_NetSelPrefTlv, 213
nas_NR5gBandPrefTlv, 220
nas_PhyCaAggPcellInfo, 223
nas_PhyCaAggScellDlBw, 223

nas_PhyCaAggScellIndex, 224
nas_PhyCaAggScellIndType, 224
nas_PhyCaAggScellInfo, 227
nas_PRLPrefTlv, 231
nas_RankIndicatorTlv, 236
nas_RatDisabledMaskTlv, 237
nas_RejectReasonTlv, 238
nas_RfBandInfoExtFormat, 239
nas_RfBandwidthInfo, 241
nas_RfDedicatedBandInfo, 242
nas_RFInfoTlv, 244
nas_RoamPrefTlv, 245
nas_SccRxInfo, 252
nas_SignalStrengthTlv, 255
nas_SLQSSignalStrengthsTlv, 259
nas_SrvDomainPrefTlv, 259
nas_timeInfo, 267
NASBandPreferenceTlv, 280
NASEmergencyModeTlv, 280
NASGWAcqOrderPrefTlv, 282
NASLTEBandPreferenceTlv, 282
NASLteNasReleaseInfoTlv, 283
NASModePreferenceTlv, 283
NASNetSelPreferenceTlv, 284
NASOTAMessageTlv, 285
NASPhyCaAggPcellInfo, 285
NASPhyCaAggScellDlBw, 286
NASPhyCaAggScellIndex, 286
NASPhyCaAggScellIndType, 287
NASPhyCaAggScellInfo, 288
NASPRLPreferenceTlv, 289
NASRoamPreferenceTlv, 291
NASServDomainPrefTlv, 292
NASTimeInfoTlv, 294
newMTMessageTlv, 294
sMSCAddressTlv, 511
sMSEtwsMessageTlv, 513
sMSOnIMSTlv, 515
swidms_ehrpdMTUSizeTlv, 517
swidms_hrpdMTUSizeTlv, 517
swidms_mtuSize3gppTlv, 518
swidms_usbMTUSizeTlv, 518
transferRouteMessageTlv, 521
unpack_pds_SetPdsState_Ind_t, 750
unpack_uim_SLQSUIMRefreshCallback_Ind_t, 834
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
UnpackSwiAvmsEventReportConfig, 922
UnpackSwiAvmsEventReportConnectionRequest, 922
UnpackSwiAvmsEventReportDataSessionStatus, 923
UnpackSwiAvmsEventReportHTTPStatus, 924
UnpackSwiAvmsEventReportNotification, 926
UnpackSwiAvmsEventReportPackageID, 926
UnpackSwiAvmsEventReportRegStatus, 926
UnpackSwiAvmsEventReportSessionType, 927
UnpackSwiAvmsEventReportWAMSParam-Change, 928
wds_ChannelRateTlv, 977
wds_ConnStatusTlv, 978
wds_DataBearTechTlv, 979
wds_DataULongLongTlv, 980
wds_DataULongTlv, 980
wds_DHCPLeaseOptTlv, 981
wds_DHCPLeaseStateTlv, 981
wds_DHCPProfileIdTlv, 983
wds_DormStatTlv, 987
wds_IPv4AdTlv, 988
wds_LastMdmCallEndRsnTlv, 989
wds_RXBytesOKTlv, 993
wds_TXBytesOKTlv, 994
tlvPresent
voice_USSDNotificationNetworkInfo, 975
Tlvresult
pack_dms_GetCustFeaturesV2_t, 303
pack_dms_SetCustFeaturesV2_t, 308
pack_dms_SetPower_t, 310
pack_dms_SetUSBComp_t, 310
pack_dms_UIMGetICCID_t, 316
pack_fms_GetImagesPreference_t, 320
pack_fms_GetStoredImages_t, 320
pack_fms_SetImagesPreference_t, 321
pack_loc_Delete_Assist_Data_t, 330
pack_loc_EventRegister_t, 333
pack_loc_SetExtPowerState_t, 333
pack_loc_SetOperationMode_t, 334
pack_loc_SLQSLOCGetBestAvailPos_t, 334
pack_loc_Start_t, 346
pack_loc_Stop_t, 346
pack_nas_SetNetworkPreference_t, 349
pack_uim_ChangePin_t, 429
pack_uim_ReadTransparent_t, 430
pack_uim_SetPinProtection_t, 431
pack_uim_UnblockPin_t, 438
pack_uim_VerifyPin_t, 439
unpack_cat_SetCATEventCallback_t, 552
unpack_dms_GetBandCapability_t, 555
unpack_dms_GetCrashAction_t, 556
unpack_dms_GetCustFeature_t, 558
unpack_dms_GetCustFeaturesV2_t, 559
unpack_dms_GetDeviceCap_t, 560
unpack_dms_GetDeviceHardwareRev_t, 562
unpack_dms_GetDeviceMfr_t, 563
unpack_dms_GetDeviceSerialNumbers_t, 565
unpack_dms_GetFirmwareInfo_t, 567
unpack_dms_GetFirmwareRevision_t, 568
unpack_dms_GetFirmwareRevisions_t, 569
unpack_dms_GetFSN_t, 570
unpack_dms_GetIMSI_t, 571
unpack_dms_GetManufacturer_t, 571
unpack_dms_GetModelID_t, 572
unpack_dms_GetNetworkTime_t, 573
unpack_dms_GetOfflineReason_t, 574
unpack_dms_GetPower_t, 575

unpack_dms_GetPRLVersion_t, 576
unpack_dms_GetUSBComp_t, 579
unpack_dms_GetVoiceNumber_t, 580
unpack_dms_PSMCfgChange_ind_t, 581
unpack_dms_ResetToFactoryDefaults_t, 582
unpack_dms_SetActivationStatusCallback_t, 582
unpack_dms_SetCustFeature_t, 583
unpack_dms_SetCustFeaturesV2_t, 584
unpack_dms_SetEventReport_ind_t, 585
unpack_dms_SetEventReport_t, 585
unpack_dms_SetFirmwarePreference_t, 586
unpack_dms_SetIndicationRegister_t, 586
unpack_dms_SetPower_t, 587
unpack_dms_SetUSBComp_t, 587
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 591
unpack_dms_SLQSDmsSwiGetResetInfo_t, 592
unpack_dms_SLQSDmsSwiIndicationRegister_t, 593
unpack_dms_SLQSGetERIFile_t, 598
unpack_dms_SLQSSetPowerSaveModeConfig_t, 600
unpack_dms_SLQSSwiClearDyingGaspStatistics-_t, 600
unpack_dms_SLQSSwiGetCrashInfo_t, 601
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 602
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 602
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 605
unpack_dms_SLQSSwiGetHostDevInfo_t, 606
unpack_dms_SLQSSwiGetOSInfo_t, 607
unpack_dms_SLQSSwiGetSerialNoExt_t, 608
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 608
unpack_dms_SLQSSwiSetHostDevInfo_t, 609
unpack_dms_SLQSSwiSetOSInfo_t, 609
unpack_dms_SLQSUIMGetState_t, 610
unpack_dms_SwiSetEventReport_t, 611
unpack_dms_UIMGetControlKeyStatus_t, 612
unpack_dms_UIMGetICCID_t, 613
unpack_dms_UIMGetPINStatus_t, 614
unpack_dms_UIMSetControlKeyProtection_t, 615
unpack_dms_UIMSetPINProtection_t, 616
unpack_dms_UIMUnblockControlKey_t, 617
unpack_fms_GetImagesPreference_t, 617
unpack_fms_GetStoredImages_t, 618
unpack_fms_SetImagesPreference_t, 619
unpack_loc_BestAvailPos_Ind_t, 648
unpack_loc_Delete_Assist_Data_t, 649
unpack_loc_DeleteAssistData_Ind_t, 650
unpack_loc_EngineState_Ind_t, 651
unpack_loc_EventNMEA_Ind_t, 651
unpack_loc_EventRegister_t, 652
unpack_loc_GetServer_Ind_t, 654
unpack_loc_GnssSvInfo_Ind_t, 655
unpack_loc_PositionRpt_Ind_t, 665
unpack_loc_SetExtPowerConfig_Ind_t, 667
unpack_loc_SetExtPowerState_t, 667
unpack_loc_SetOperationMode_Ind_t, 668
unpack_loc_SetOperationMode_t, 669
unpack_loc_SetServer_Ind_t, 669
unpack_loc_SLQSLOCGetBestAvailPos_t, 670
unpack_loc_Start_t, 670
unpack_loc_Stop_t, 671
unpack_nas_GetNetworkPreference_t, 679
unpack_nas_SetNetworkPreference_t, 689
unpack_nas_SetServingSystemCallback_ind_t, 691
unpack_nas_SlqsGetLTECphyCAInfo_t, 692
unpack_nas_SLQSNasGetRFInfo_t, 721
unpack_nas_SLQSNasSwiOTAMessageCallback-_ind_t, 727
unpack_nas_SLQSSetSysSelectionPrefCallBack-_ind_t, 729
unpack_result_t, 778
unpack_swidms_SLQSSwiDmsSetMTU_t, 816
unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 824
unpack_tmd_SLQSTmdGetMitigationDevList_t, 824
unpack_tmd_SLQSTmdGetMitigationLvl_t, 825
unpack_tmd_SLQSTmdRegNotMitigationLvl_t, 826
unpack_uim_ChangePin_t, 827
unpack_uim_GetCardStatus_t, 828
unpack_uim_ReadTransparent_t, 828
unpack_uim_SetPinProtection_t, 829
unpack_uim_UnblockPin_t, 836
unpack_uim_VerifyPin_t, 836
unpack_wds_SLQSCreateProfile_t, 894
unpack_wds_SLQSGetProfileSettings_t, 903
unpack_wds_SLQSSetIPFamilyPreference_t, 907
tmd.h, 1332
pack_tmd_SLQSTmdDeRegNotMitigationLvl, 1333
pack_tmd_SLQSTmdGetMitigationDevList, 1333
pack_tmd_SLQSTmdGetMitigationLvl, 1334
pack_tmd_SLQSTmdRegNotMitigationLvl, 1334
TMD_MAX_DEV_LIST, 1333
unpack_tmd_SLQSTmdDeRegNotMitigationLvl, 1334
unpack_tmd_SLQSTmdGetMitigationDevList, 1335
unpack_tmd_SLQSTmdGetMitigationLvl, 1335
unpack_tmd_SLQSTmdMitigationLvlRptCallback_-ind, 1336
unpack_tmd_SLQSTmdRegNotMitigationLvl, 1336
tmd_mitigationDevList, 520
maxMitigationLevel, 521
mitigationDevId, 521
mitigationDevIdLen, 521
tmrSIPRegn
ims_SIPRegnTmrInfo, 79
tmrT1
ims_TmrT1Info, 81
tmrT2
ims_TmrT2Info, 82
tmrTf

ims_TmrTfInfo, 82
toServiceId
sms_BroadcastConfig, 498
toggleMode
voice_lineCtrlInfo, 964
TokenBucket
unpack_qos_swiQosFlow_t, 776
tokenRate
unpack_qos_tokenBucket_t, 777
tosMask
LibPackTFTIDParams, 117
total_rx_bytes
unpack_qos_SLQSQosSwiReadDataStats_t, 765
total_rx_pkt
unpack_qos_SLQSQosSwiReadDataStats_t, 765
total_tx_bytes
unpack_qos_SLQSQosSwiReadDataStats_t, 765
total_tx_bytes_drp
unpack_qos_SLQSQosSwiReadDataStats_t, 765
total_tx_pkt
unpack_qos_SLQSQosSwiReadDataStats_t, 765
total_tx_pkt_drp
unpack_qos_SLQSQosSwiReadDataStats_t, 765
TrackAreaCode
unpack_nas_SLQSGetServingSystem_t, 702
TrackingStatus
unpack_pds_SetPdsState_Ind_t, 750
TrafficClass
unpack_qos_swiQosFlow_t, 776
trafficClass
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
trafficPriority
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
TranDstPort
unpack_qos_swiQosFilter_t, 772
TranSrcPort
unpack_qos_swiQosFilter_t, 772
TransCap
sms_transLayerInfo, 510
TransType
sms_transLayerInfo, 510
transactionID
sMSTransferRouteMTMessageInfo, 516
transferDelay
LibPackUMTSQoS, 119
wds_UMTSMinQoS, 997
TransferRouteMTMessageInfo
transferRouteMessageTlv, 521
transferRouteMessageTlv, 521
TlvPresent, 521
TransferRouteMTMessageInfo, 521
transferStats
pack_wds_SLQSSetWdsEventCallback_t, 478
trueSrvStatus
nas_GSMSrvStatusInfo, 174
tx_bytes
unpack_QosFlowStat_t, 778
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
tx_bytes_drp
unpack_QosFlowStat_t, 778
tx_pkt
unpack_QosFlowStat_t, 778
tx_pkt_drp
unpack_QosFlowStat_t, 778
tx_pkts
unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
TxByteOKCnt
wds_TXBytesOKTlv, 994
TxDropConutTlv
unpack_RMTransferStatistics_ind_t, 780
txOKBytesCount
unpack_wds_SLQSGetDUNCallInfo_t, 903
TxOkByteCountTlv
unpack_RMTransferStatistics_ind_t, 780
TxOkConutTlv
unpack_RMTransferStatistics_ind_t, 780
txPower
nas_txInfo, 268
TxQFilter
unpack_qos_QosFlowInfo_t, 761
TxQFlowGranted
unpack_qos_QosFlowInfo_t, 761
type
pack_wds_GetDefaultProfileNum_t, 459
pack_wds_SetDefaultProfileNum_t, 465
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 591
unpack_dms_SLQSDmsSwiGetResetInfo_t, 592
unpack_qmi_t, 756
typeSize
unpack_sms_GetSMSCAddress_t, 782

u16PRLVersion
unpack_dms_GetPRLVersion_t, 576
u8PRLPreference
unpack_dms_GetPRLVersion_t, 576
UDPDstPort
unpack_qos_swiQosFilter_t, 772
UDPSrcPort
unpack_qos_swiQosFilter_t, 772
ULONG
SwiDataTypes.h, 1311
ULONGLONG
SwiDataTypes.h, 1311
UMTSGrantedQoS
unpack_wds_SLQSGetRuntimeSettings_t, 906
UMTSInstInfo
nas_UMTSInfo, 270
UMTSLTENbrCell
nas_WCDMAInfoLTENeighborCell, 275
UMTSReqQoS
LibPackUMTSReqQoSSigInd, 120
UNIQUE_ID_LEN

dms.h, 1025
UNUSEDPARAM
common.h, 1013
SwiDataTypes.h, 1310
uPhyCardStatus
slot_t, 495
uPhySlotStatus
slot_t, 495
USBComp
pack_dms_SetUSBComp_t, 310
unpack_dms_GetUSBComp_t, 579
USHORT
SwiDataTypes.h, 1311
USSDNotificationNetworkInfo
unpack_voice_USSDNotificationCallback_ind_t, 876
USSInformation
pack_voice_SLQSVoiceOrigUSSDNoWait_t, 452
UUSData
voice_UUSInfo, 976
UUSDatalen
voice_UUSInfo, 976
UUSDcs
voice_UUSInfo, 976
UUSType
voice_UUSInfo, 976
uarfcn
nas_lteWcdmaCellInfo, 209
nas_UMTSInfo, 270
nas_wcdmaUARFCN, 280
ueInIdle
nas_LTEInfoInterfreq, 192
nas_LTEInfoIntrafreq, 194
nas_LTEInfoNeighboringGSM, 195
nas_LTEInfoNeighboringWCDMA, 196
uim.h, 1336
MAX_ICCID_LENGTH, 1339
MAX_NO_OF_SLOTS, 1339
MAX_SLOTS_STATUS, 1339
pack_uim_ChangePin, 1340
pack_uim_GetCardStatus, 1340
pack_uim_ReadTransparent, 1340
pack_uim_SLQSUIMAuthenticate, 1341
pack_uim_SLQSUIMDepersonalization, 1341
pack_uim_SLQSUIMEventRegister, 1342
pack_uim_SLQSUIMGetConfiguration, 1342
pack_uim_SLQSUIMGetFileAttributes, 1342
pack_uim_SLQSUIMGetSlotsStatus, 1343
pack_uim_SLQSUIMPowerDown, 1343
pack_uim_SLQSUIMPowerUp, 1344
pack_uim_SLQSUIMRefreshComplete, 1344
pack_uim_SLQSUIMRefreshGetLastEvent, 1344
pack_uim_SLQSUIMRefreshOK, 1345
pack_uim_SLQSUIMRefreshRegister, 1345
pack_uim_SLQSUIMReset, 1345
pack_uim_SLQSUIMSwitchSlot, 1346
pack_uim_SetPinProtection, 1341
pack_uim_UnblockPin, 1346
pack_uim_VerifyPin, 1346
unpack_uim_ChangePin, 1347
unpack_uim_GetCardStatus, 1347
unpack_uim_ReadTransparent, 1348
unpack_uim_SLQSUIMAuthenticate, 1349
unpack_uim_SLQSUIMDepersonalization, 1349
unpack_uim_SLQSUIMEventRegister, 1349
unpack_uim_SLQSUIMGetConfiguration, 1350
unpack_uim_SLQSUIMGetFileAttributes, 1350
unpack_uim_SLQSUIMGetSlotsStatus, 1350
unpack_uim_SLQSUIMPowerDown, 1351
unpack_uim_SLQSUIMPowerUp, 1351
unpack_uim_SLQSUIMRefreshCallback_Ind, 1351
unpack_uim_SLQSUIMRefreshComplete, 1352
unpack_uim_SLQSUIMRefreshGetLastEvent, 1352
unpack_uim_SLQSUIMRefreshOK, 1352
unpack_uim_SLQSUIMRefreshRegister, 1353
unpack_uim_SLQSUIMReset, 1353
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind, 1353
unpack_uim_SLQSUIMSwitchSlot, 1354
unpack_uim_SetPinProtection, 1348
unpack_uim_SetUimSlotStatusChangeCallback_ind, 1348
unpack_uim_UnblockPin, 1354
unpack_uim_VerifyPin, 1354
uim_UIMSessionInformation, 544
aid, 545
aidLength, 545
sessionType, 545
uim_appStatus, 521
aidLength, 524
aidVal, 524
appState, 524
appType, 524
persoFeature, 524
persoRetries, 524
persoState, 524
persoUnblockRetries, 524
pin1Retries, 524
pin1State, 524
pin2Retries, 524
pin2State, 524
puk1Retries, 524
puk2Retries, 524
univPin, 524
uim_authenticateResult, 524
content, 525
contentLen, 525
uim_authenticationData, 525
context, 526
data, 526
dataLen, 526
uim_cardResult, 526
sw1, 527
sw2, 527
uim_cardStatus, 527

index1xPri, 528
index1xSec, 528
indexGwPri, 528
indexGwSec, 528
numSlot, 528
SlotInfo, 528
uim_changeUIMPIN, 528
oldPINLen, 529
oldPINVal, 529
pinID, 529
pinLen, 529
pinVal, 529
uim_depersonalizationInformation, 529
ckLen, 530
ckVal, 530
feature, 530
operation, 530
uim_encryptedPIN1, 530
pin1Len, 531
pin1Val, 531
uim_fileAttributes, 531
fileID, 534
fileSize, 534
fileType, 534
rawLen, 534
rawValue, 534
recordCount, 534
recordSize, 534
secActivate, 534
secActivateMask, 534
secDeactivate, 534
secDeactivateMask, 534
secIncrease, 534
secIncreaseMask, 534
secRead, 534
secReadMask, 534
secWrite, 534
secWriteMask, 534
uim_fileInfo, 534
fileID, 535
path, 535
pathLen, 535
uim_hotSwapStatus, 535
hotSwap, 535
hotSwapLength, 535
uim_personalizationStatus, 535
feature, 536
numFeatures, 536
unblockLeft, 536
verifyLeft, 536
uim_readResult, 536
content, 537
contentLen, 537
uim_readTransparentInfo, 537
length, 537
offset, 537
uim_refreshevent, 537
aid, 539
aidLength, 539
arrfileInfo, 539
mode, 539
numOfFiles, 539
sessionType, 539
stage, 539
uim_registerRefresh, 539
arrfileInfo, 540
numFiles, 540
registerFlag, 540
voteForInit, 540
uim_remainingRetries, 540
unblockLeft, 540
verifyLeft, 540
uim_sessionInformation, 541
aid, 541
aidLength, 541
sessionType, 541
uim_setPINProtection, 541
pinID, 542
pinLength, 542
pinOperation, 542
pinValue, 542
uim_slotInfo, 542
AppStatus, 544
cardState, 544
errorState, 544
numApp, 544
upinRetries, 544
upinState, 544
upukRetries, 544
uim_unblockUIMPIN, 545
newPINLen, 545
newPINVal, 545
pinID, 545
pukLen, 545
pukVal, 546
uim_verifyUIMPIN, 546
pinID, 546
pinLen, 546
pinVal, 546
uimSlotStatus
slots_t, 497
ulData
wds_DataULongTlv, 980
ulMask
rmTrasnferStaticsReq, 493
ulMax
PackSwiAvmsSetSettingsPeriodInfo, 487
ulMin
PackSwiAvmsSetSettingsPeriodInfo, 487
ulPhysicalSlot
pack_uim_SLQSUIMSwitchSlot_t, 437
ulPkgDownloadComplete
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
ulPkgDownloadSize

UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
ullData
wds_DataULongLongTlv, 980
umtsEcio
nas_UMTSinstInfo, 271
umtsInst
nas_UMTSInfo, 270
umtsLTENbrCellLen
nas_WCDMAInfoLTENeighborCell, 275
umtsPsc
nas_UMTSinstInfo, 271
umtsRscp
nas_UMTSinstInfo, 271
umtsUarfcn
nas_UMTSinstInfo, 271
UnPackGetProfileSettingOut, 918
curProfile, 919
pExtErrCode, 919
unblockLeft
uim_personalizationStatus, 536
uim_remainingRetries, 540
unblockRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 612
unpack_dms_UIMSetPINProtection_t, 616
unpack_dms_UIMUnblockControlKey_t, 617
uniqueID
image_info_t, 69
univPin
appStats, 38
uim_appStatus, 524
universalTime
unpack_nas_SLQSNasNetworkTimeCallBack_ind-_t, 724
unpack_QosFlowStat_t, 777
bearerId, 778
tx_bytes, 778
tx_bytes_drp, 778
tx_pkt, 778
tx_pkt_drp, 778
unpack_RMTransferStatistics_ind_t, 780
RxDropConutTlv, 780
RxOkByteCountTlv, 780
RxOkConutTlv, 780
TxDropConutTlv, 780
TxOkByteCountTlv, 780
TxOkConutTlv, 780
unpack_audio_SLQSGetAudioPathConfig
audio.h, 1005
unpack_audio_SLQSGetAudioPathConfig_t, 546
pCodecSTGain, 548
pECMode, 548
pMICGainSelect, 548
pNSEnable, 548
pRXAGCList, 548
pRXAVCList, 548
pTXAGCList, 548
pTXAVCSwitch, 548
pTXGain, 548
ParamPresenceMask, 548
unpack_audio_SLQSGetAudioProfile
audio.h, 1005
unpack_audio_SLQSGetAudioProfile_t, 549
EarMute, 549
MicMute, 550
ParamPresenceMask, 550
Profile, 550
Volume, 550
unpack_audio_SLQSGetAudioVolTLBConfig
audio.h, 1005
unpack_audio_SLQSGetAudioVolTLBConfig_t, 550
ParamPresenceMask, 550
ResCode, 550
unpack_audio_SLQSSetAudioPathConfig
audio.h, 1006
unpack_audio_SLQSSetAudioProfile
audio.h, 1006
unpack_audio_SLQSSetAudioVolTLBConfig
audio.h, 1006
unpack_audio_SLQSSetAudioVolTLBConfig_t, 550
ParamPresenceMask, 551
ResCode, 551
unpack_cat_CATSendEnvelopeCommand
cat.h, 1009
unpack_cat_CATSendEnvelopeCommand_t
cat.h, 1008
unpack_cat_CATSendTerminalResponse
cat.h, 1009
unpack_cat_CATSendTerminalResponse_t
cat.h, 1008
unpack_cat_SetCATEventCallback
cat.h, 1010
unpack_cat_SetCATEventCallback_t, 551
errorMask, 552
ParamPresenceMask, 552
Tlvresult, 552
unpack_cat_SetCatEventCallback_ind
cat.h, 1010
unpack_cat_SetCatEventCallback_ind_t, 551
CCETlv, 551
event_Index, 551
ParamPresenceMask, 551
unpack_dms_ActivateAutomatic
dms.h, 1049
unpack_dms_GetActivationState
dms.h, 1049
unpack_dms_GetActivationState_t, 552
ParamPresenceMask, 553
state, 553
unpack_dms_GetBandCapability
dms.h, 1050
unpack_dms_GetBandCapability_t, 553
BandCapability, 555
ParamPresenceMask, 555
Tlvresult, 555
unpack_dms_GetCrashAction

dms.h, 1050
unpack_dms_GetCrashAction_t, 555
DevCrashState, 555
ParamPresenceMask, 555
Tlvresult, 556
unpack_dms_GetCustFeature
dms.h, 1050
unpack_dms_GetCustFeature_t, 556
DHCPRelayEnabled, 558
DisableIMSI, 558
GPSLPM, 558
GPSSel, 558
GpsEnable, 558
IPFamSupport, 558
IsVoiceEnabled, 558
ParamPresenceMask, 558
RMAutoConnect, 558
SMSSupport, 558
Tlvresult, 558
unpack_dms_GetCustFeaturesV2
dms.h, 1051
unpack_dms_GetCustFeaturesV2_t, 558
GetCustomFeatureV2, 559
ParamPresenceMask, 559
Tlvresult, 559
unpack_dms_GetDeviceCap
dms.h, 1051
unpack_dms_GetDeviceCap_t, 559
DataServiceCapability, 560
MaxRXChannelRate, 560
MaxTXChannelRate, 560
ParamPresenceMask, 560
RadioIfaces, 560
RadioIfacesSize, 560
SimCapability, 560
Tlvresult, 560
unpack_dms_GetDeviceCapabilities
dms.h, 1051
unpack_dms_GetDeviceCapabilities_t, 560
dataServiceCaCapability, 561
maxRxChannelRate, 562
maxTxChannelRate, 562
ParamPresenceMask, 562
RadioIfaces, 562
radioIfacesSize, 562
simCapability, 562
unpack_dms_GetDeviceHardwareRev
dms.h, 1052
unpack_dms_GetDeviceHardwareRev_t, 562
ParamPresenceMask, 562
String, 562
stringSize, 562
Tlvresult, 562
unpack_dms_GetDeviceMfr
dms.h, 1052
unpack_dms_GetDeviceMfr_t, 563
ParamPresenceMask, 563
String, 563
stringSize, 563
Tlvresult, 563
unpack_dms_GetDeviceSerialNumbers
dms.h, 1052
unpack_dms_GetDeviceSerialNumbers_t, 563
ESNString, 565
esnSize, 565
IMEIString, 565
imeiSize, 565
imeiSvnSize, 565
ImeiSvnString, 565
MEIDString, 565
meidSize, 565
ParamPresenceMask, 565
Tlvresult, 565
unpack_dms_GetFSN
dms.h, 1054
unpack_dms_GetFSN_t, 569
ParamPresenceMask, 570
String, 570
Tlvresult, 570
unpack_dms_GetFirmwareInfo
dms.h, 1053
unpack_dms_GetFirmwareInfo_t, 565
appversion_str, 566
bootversion_str, 566
carrier_str, 566
cur_carr_name, 567
cur_carr_rev, 567
modelid_str, 567
packageid_str, 567
ParamPresenceMask, 567
priversion_str, 567
sku_str, 567
Tlvresult, 567
unpack_dms_GetFirmwareRevision
dms.h, 1053
unpack_dms_GetFirmwareRevision_t, 567
AMSSString, 568
amssSize, 568
PRIString, 568
ParamPresenceMask, 568
Tlvresult, 568
unpack_dms_GetFirmwareRevisions
dms.h, 1053
unpack_dms_GetFirmwareRevisions_t, 568
AMSSString, 569
amssSize, 569
bootSize, 569
BootString, 569
PRIString, 569
ParamPresenceMask, 569
priSize, 569
Tlvresult, 569
unpack_dms_GetHardwareRevision
dms.h, 1054
unpack_dms_GetHardwareRevision_t, 570
hwVer, 570

ParamPresenceMask, 570
unpack_dms_GetIMSI
dms.h, 1054
unpack_dms_GetIMSI_t, 570
imsi, 571
ParamPresenceMask, 571
Tlvresult, 571
unpack_dms_GetManufacturer
dms.h, 1055
unpack_dms_GetManufacturer_t, 571
manufacturer, 571
ParamPresenceMask, 571
Tlvresult, 571
unpack_dms_GetModelID
dms.h, 1055
unpack_dms_GetModelID_t, 572
modelid, 572
ParamPresenceMask, 572
Tlvresult, 572
unpack_dms_GetNetworkTime
dms.h, 1055
unpack_dms_GetNetworkTime_t, 572
ParamPresenceMask, 573
source, 573
timestamp, 573
Tlvresult, 573
unpack_dms_GetOfflineReason
dms.h, 1056
unpack_dms_GetOfflineReason_t, 573
pReasonMask, 574
ParamPresenceMask, 574
pbPlatform, 574
Tlvresult, 574
unpack_dms_GetPRLVersion
dms.h, 1056
unpack_dms_GetPRLVersion_t, 575
ParamPresenceMask, 576
Tlvresult, 576
u16PRLVersion, 576
u8PRLPreference, 576
unpack_dms_GetPower
dms.h, 1056
unpack_dms_GetPower_t, 574
HardwareControlledMode, 575
OfflineReason, 575
OperationMode, 575
ParamPresenceMask, 575
Tlvresult, 575
unpack_dms_GetSerialNumbers
dms.h, 1057
unpack_dms_GetSerialNumbers_t, 576
esn, 577
imei_no, 577
imeisv_svn, 577
meid, 577
ParamPresenceMask, 577
unpack_dms_GetUSBComp
dms.h, 1057
unpack_dms_GetUSBComp_t, 577
NumSupUSBComps, 579
ParamPresenceMask, 579
SupUSBComps, 579
Tlvresult, 579
USBComp, 579
unpack_dms_GetVoiceNumber
dms.h, 1057
unpack_dms_GetVoiceNumber_t, 579
MIN, 580
minSize, 580
ParamPresenceMask, 580
Tlvresult, 580
VoiceNumber, 580
voiceNumberSize, 580
unpack_dms_PSMCfgChange_ind
dms.h, 1058
unpack_dms_PSMCfgChange_ind_t, 581
ActiveTimer, 581
EnableState, 581
ParamPresenceMask, 581
PeriodicUpdateTimer, 581
Tlvresult, 581
unpack_dms_ResetToFactoryDefaults
dms.h, 1058
unpack_dms_ResetToFactoryDefaults_t, 581
ParamPresenceMask, 582
Tlvresult, 582
unpack_dms_SLQSDmsSwiGetPCInfo
dms.h, 1062
unpack_dms_SLQSDmsSwiGetPCInfo_t, 587
has_LpmFlag, 589
has_PersistentLpm, 589
has_Wdisable, 589
LpmFlag, 590
opMode, 590
ParamPresenceMask, 590
PersistentLpm, 590
PowerOffMode, 590
Wdisable, 590
unpack_dms_SLQSDmsSwiGetResetInfo
dms.h, 1062
unpack_dms_SLQSDmsSwiGetResetInfo_Ind
dms.h, 1063
unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t, 590
ParamPresenceMask, 591
source, 591
Tlvresult, 591
type, 591
unpack_dms_SLQSDmsSwiGetResetInfo_t, 591
ParamPresenceMask, 592
source, 592
Tlvresult, 592
type, 592
unpack_dms_SLQSDmsSwiIndicationRegister
dms.h, 1063
unpack_dms_SLQSDmsSwiIndicationRegister_t, 592
ParamPresenceMask, 593

Tlvresult, 593
unpack_dms_SLQSGetBandCapability
dms.h, 1063
unpack_dms_SLQSGetBandCapability_t, 593
bandCapability, 597
LteBandCapability, 597
ParamPresenceMask, 597
TdsBandCapability, 597
unpack_dms_SLQSGetERIFile
dms.h, 1064
unpack_dms_SLQSGetERIFile_t, 597
eriFile, 598
ParamPresenceMask, 598
Tlvresult, 598
unpack_dms_SLQSGetPowerSaveModeConfig
dms.h, 1064
unpack_dms_SLQSGetPowerSaveModeConfig_t, 598
pActiveTimer, 599
pDurationThreshold, 599
pEarlyWakeupTime, 599
pPeriodicUpdateTimer, 599
pPsmEnableState, 599
pRandomizationWindow, 599
ParamPresenceMask, 599
unpack_dms_SLQSSetPowerSaveModeConfig
dms.h, 1064
unpack_dms_SLQSSetPowerSaveModeConfig_t, 599
ParamPresenceMask, 600
Tlvresult, 600
unpack_dms_SLQSSwiClearDyingGaspStatistics
dms.h, 1065
unpack_dms_SLQSSwiClearDyingGaspStatistics_t, 600
ParamPresenceMask, 600
Tlvresult, 600
unpack_dms_SLQSSwiGetCrashInfo
dms.h, 1065
unpack_dms_SLQSSwiGetCrashInfo_t, 600
crashInfoParam, 601
ParamPresenceMask, 601
Tlvresult, 601
unpack_dms_SLQSSwiGetDyingGaspCfg
dms.h, 1066
unpack_dms_SLQSSwiGetDyingGaspCfg_t, 601
pGetDyingGaspCfg, 602
ParamPresenceMask, 602
Tlvresult, 602
unpack_dms_SLQSSwiGetDyingGaspStatistics
dms.h, 1066
unpack_dms_SLQSSwiGetDyingGaspStatistics_t, 602
pGetDyingGaspStatistics, 602
ParamPresenceMask, 602
Tlvresult, 602
unpack_dms_SLQSSwiGetFirmwareCurr
dms.h, 1066
unpack_dms_SLQSSwiGetFirmwareCurr_t, 602
carrier, 603
fwvers, 603
numEntries, 603
pCurrImgInfo, 603
ParamPresenceMask, 603
pkgver, 603
priver, 603
unpack_dms_SLQSSwiGetFwUpdateStatus
dms.h, 1067
unpack_dms_SLQSSwiGetFwUpdateStatus_t, 603
imgType, 605
logString, 605
ParamPresenceMask, 605
refData, 605
refString, 605
ResCode, 605
Tlvresult, 605
unpack_dms_SLQSSwiGetHostDevInfo
dms.h, 1067
unpack_dms_SLQSSwiGetHostDevInfo_t, 605
manString, 606
modelString, 606
ParamPresenceMask, 606
plasmaIDString, 606
swVerString, 606
Tlvresult, 606
unpack_dms_SLQSSwiGetOSInfo
dms.h, 1067
unpack_dms_SLQSSwiGetOSInfo_t, 606
nameString, 607
ParamPresenceMask, 607
Tlvresult, 607
versionString, 607
unpack_dms_SLQSSwiGetSerialNoExt
dms.h, 1068
unpack_dms_SLQSSwiGetSerialNoExt_t, 607
meidString, 607
ParamPresenceMask, 607
Tlvresult, 608
unpack_dms_SLQSSwiSetDyingGaspCfg
dms.h, 1068
unpack_dms_SLQSSwiSetDyingGaspCfg_t, 608
ParamPresenceMask, 608
Tlvresult, 608
unpack_dms_SLQSSwiSetHostDevInfo
dms.h, 1068
unpack_dms_SLQSSwiSetHostDevInfo_t, 608
ParamPresenceMask, 608
Tlvresult, 609
unpack_dms_SLQSSwiSetOSInfo
dms.h, 1069
unpack_dms_SLQSSwiSetOSInfo_t, 609
ParamPresenceMask, 609
Tlvresult, 609
unpack_dms_SLQSUIMGetState
dms.h, 1069
unpack_dms_SLQSUIMGetState_t, 609
ParamPresenceMask, 610
state, 610
Tlvresult, 610

unpack_dms_SetActivationStatusCallback
dms.h, 1058
unpack_dms_SetActivationStatusCallback_t, 582
ParamPresenceMask, 582
Tlvresult, 582
unpack_dms_SetCrashAction
dms.h, 1059
unpack_dms_SetCrashAction_t, 582
notused, 583
ParamPresenceMask, 583
unpack_dms_SetCustFeature
dms.h, 1059
unpack_dms_SetCustFeature_t, 583
ParamPresenceMask, 583
Tlvresult, 583
unpack_dms_SetCustFeaturesV2
dms.h, 1059
unpack_dms_SetCustFeaturesV2_t, 583
ParamPresenceMask, 584
Tlvresult, 584
unpack_dms_SetEventReport
dms.h, 1060
unpack_dms_SetEventReport_ind
dms.h, 1060
unpack_dms_SetEventReport_ind_t, 584
ActivationStatusTlv, 584
OperatingModeTlv, 584
ParamPresenceMask, 585
Tlvresult, 585
unpack_dms_SetEventReport_t, 585
ParamPresenceMask, 585
Tlvresult, 585
unpack_dms_SetFirmwarePreference
dms.h, 1060
unpack_dms_SetFirmwarePreference_t, 585
ParamPresenceMask, 586
Tlvresult, 586
unpack_dms_SetIndicationRegister
dms.h, 1061
unpack_dms_SetIndicationRegister_t, 586
ParamPresenceMask, 586
Tlvresult, 586
unpack_dms_SetPower
dms.h, 1061
unpack_dms_SetPower_t, 586
ParamPresenceMask, 587
Tlvresult, 587
unpack_dms_SetUSBComp
dms.h, 1061
unpack_dms_SetUSBComp_t, 587
ParamPresenceMask, 587
Tlvresult, 587
unpack_dms_SwiEventReportCallBack_ind
dms.h, 1069
unpack_dms_SwiEventReportCallBack_ind_t, 610
ParamPresenceMask, 610
TempTlv, 610
VoltTlv, 610
unpack_dms_SwiSetEventReport
dms.h, 1070
unpack_dms_SwiSetEventReport_t, 610
ParamPresenceMask, 611
Tlvresult, 611
unpack_dms_UIMChangePIN
dms.h, 1070
unpack_dms_UIMGetControlKeyStatus
dms.h, 1070
unpack_dms_UIMGetControlKeyStatus_t, 611
facilityState, 612
ParamPresenceMask, 612
Tlvresult, 612
unblockRetriesLeft, 612
verifyRetriesLeft, 612
unpack_dms_UIMGetICCID
dms.h, 1071
unpack_dms_UIMGetICCID_t, 612
ParamPresenceMask, 612
String, 613
stringSize, 613
Tlvresult, 613
unpack_dms_UIMGetPINStatus
dms.h, 1071
unpack_dms_UIMGetPINStatus_t, 613
p1Status, 614
p1UnblockRetriesLeft, 614
p1VerifyRetriesLeft, 614
p2Status, 614
p2UnblockRetriesLeft, 614
p2VerifyRetriesLeft, 614
ParamPresenceMask, 614
Tlvresult, 614
unpack_dms_UIMSetControlKeyProtection
dms.h, 1071
unpack_dms_UIMSetControlKeyProtection_t, 614
ParamPresenceMask, 615
Tlvresult, 615
verifyRetriesLeft, 615
unpack_dms_UIMSetPINProtection
dms.h, 1072
unpack_dms_UIMSetPINProtection_t, 615
ParamPresenceMask, 616
Tlvresult, 616
unblockRetriesLeft, 616
verifyRetriesLeft, 616
unpack_dms_UIMUnblockControlKey
dms.h, 1072
unpack_dms_UIMUnblockControlKey_t, 616
ParamPresenceMask, 616
Tlvresult, 617
unblockRetriesLeft, 617
unpack_dms_UIMUnblockPIN
dms.h, 1073
unpack_dms_UIMVerifyPIN
dms.h, 1073
unpack_dms_ValidateSPC
dms.h, 1073

unpack_fms_GetImagesPreference
fms.h, 1076
unpack_fms_GetImagesPreference_t, 617
ImageListSize, 617
pImageList, 617
ParamPresenceMask, 617
Tlvresult, 617
unpack_fms_GetStoredImages
fms.h, 1076
unpack_fms_GetStoredImages_t, 617
imageList, 618
imagelistSize, 618
ParamPresenceMask, 618
Tlvresult, 618
unpack_fms_SetImagesPreference
fms.h, 1076
unpack_fms_SetImagesPreference_t, 618
ImageTypes, 619
ImageTypesSize, 619
ParamPresenceMask, 619
Tlvresult, 619
unpack_ims_SLQSGetIMSSMSConfig
ims.h, 1084
unpack_ims_SLQSGetIMSSMSConfig_t, 619
pSettingResp, 620
ParamPresenceMask, 620
unpack_ims_SLQSGetIMSUserConfig
ims.h, 1084
unpack_ims_SLQSGetIMSUserConfig_t, 620
pIMSDomain, 620
pSettingResp, 620
ParamPresenceMask, 620
unpack_ims_SLQSGetIMSVoIPConfig
ims.h, 1084
unpack_ims_SLQSGetIMSVoIPConfig_t, 621
pAmrMode, 623
pRingingTimer, 623
pSettingResp, 623
ParamPresenceMask, 623
unpack_ims_SLQSGetRegMgrConfig
ims.h, 1085
unpack_ims_SLQSGetRegMgrConfig_t, 623
pIMSTestMode, 624
pPCSCFPort, 624
pSettingResp, 624
ParamPresenceMask, 624
unpack_ims_SLQSGetSIPConfig
ims.h, 1085
unpack_ims_SLQSGetSIPConfig_t, 624
pSIPLocalPort, 626
pSettingResp, 626
pSigCompEnabled, 626
pSubscribeTimer, 626
pTimerSIPReg, 626
pTimerT1, 626
pTimerT2, 626
pTimerTf, 626
ParamPresenceMask, 626
unpack_ims_SLQSImsConfigIndicationRegister
ims.h, 1085
unpack_ims_SLQSImsConfigIndicationRegister_t
ims.h, 1079
unpack_ims_SLQSRegMgrCfgCallBack_ind
ims.h, 1086
unpack_ims_SLQSRegMgrCfgCallBack_ind_t, 626
PCTlv, 627
PNTlv, 627
ParamPresenceMask, 627
TMTlv, 627
unpack_ims_SLQSSIPCfgCallBack_ind
ims.h, 1088
unpack_ims_SLQSSIPCfgCallBack_ind_t, 629
ParamPresenceMask, 630
SCTlv, 630
SPTlv, 630
SRTlv, 630
STTlv, 630
TT1Tlv, 630
TT2Tlv, 630
TTfTlv, 630
unpack_ims_SLQSSMSCfgCallBack_ind
ims.h, 1088
unpack_ims_SLQSSMSCfgCallBack_ind_t, 631
ParamPresenceMask, 631
SFTlv, 631
unpack_ims_SLQSSetIMSSMSConfig
ims.h, 1086
unpack_ims_SLQSSetIMSSMSConfig_t, 627
pSettingResp, 627
ParamPresenceMask, 627
unpack_ims_SLQSSetIMSUserConfig
ims.h, 1086
unpack_ims_SLQSSetIMSUserConfig_t, 627
pSettingResp, 628
ParamPresenceMask, 628
unpack_ims_SLQSSetIMSVoIPConfig
ims.h, 1087
unpack_ims_SLQSSetIMSVoIPConfig_t, 628
pSettingResp, 628
ParamPresenceMask, 628
unpack_ims_SLQSSetRegMgrConfig
ims.h, 1087
unpack_ims_SLQSSetRegMgrConfig_t, 628
pSettingResp, 629
ParamPresenceMask, 628
unpack_ims_SLQSSetSIPConfig
ims.h, 1087
unpack_ims_SLQSSetSIPConfig_t, 629
pSettingResp, 629
ParamPresenceMask, 629
unpack_ims_SLQSUserCfgCallBack_ind
ims.h, 1088
unpack_ims_SLQSUserCfgCallBack_ind_t, 631
IDTlv, 632
ParamPresenceMask, 632
unpack_ims_SLQSVoIPCfgCallBack_ind

ims.h, 1089
unpack_ims_SLQSVoIPCfgCallBack_ind_t, 632
AMTlv, 634
ParamPresenceMask, 634
RTTlv, 634
SDTlv, 634
unpack_imsa_SLQSGetIMSARegStatus
imsa.h, 1092
unpack_imsa_SLQSGetIMSARegStatus_t, 634
ImsRegErrCode, 635
ImsRegStatus, 635
NewImsRegStatus, 635
ParamPresenceMask, 635
unpack_imsa_SLQSGetIMSAServiceStatus
imsa.h, 1092
unpack_imsa_SLQSGetIMSAServiceStatus_t, 635
ParamPresenceMask, 637
SmsServiceRat, 637
SmsServiceStatus, 637
UtServiceRat, 637
UtServiceStatus, 637
VoipServiceRat, 637
VoipServiceStatus, 637
VsServiceRat, 637
VsServiceStatus, 637
VtServiceRat, 637
VtServiceStatus, 638
unpack_imsa_SLQSImsaPdpStatusCallBack_ind
imsa.h, 1093
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t, 638
FailErrCode, 638
ParamPresenceMask, 638
PdpConnState, 638
unpack_imsa_SLQSImsaRatStatusCallBack_ind
imsa.h, 1093
unpack_imsa_SLQSImsaRatStatusCallBack_ind_t, 638
ParamPresenceMask, 639
RatHandover, 639
unpack_imsa_SLQSImsaRegStatusCallBack_ind
imsa.h, 1093
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t, 639
IMSRegistration, 640
ParamPresenceMask, 640
unpack_imsa_SLQSImsaSvcStatusCallBack_ind
imsa.h, 1094
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 640
ParamPresenceMask, 641
SmsRat, 641
SmsService, 641
UtRat, 641
UtService, 641
VoipRat, 641
VoipService, 641
VtRat, 641
VtService, 641
unpack_imsa_SLQSRegisterIMSAIndication
imsa.h, 1094
unpack_imsa_SLQSRegisterIMSAIndication_t
imsa.h, 1090
unpack_loc_BestAvailPos_Ind
loc.h, 1116
unpack_loc_BestAvailPos_Ind_t, 641
pAltitudeWrtEllipsoid, 647
pAltitudeWrtMeanSeaLevel, 647
pGpsTime, 647
pHeading, 647
pHeadingUnc, 647
pHorCirConf, 647
pHorEllpConf, 647
pHorReliability, 647
pHorUncCircular, 647
pHorUncEllipseOrientAzimuth, 647
pHorUncEllipseSemiMajor, 647
pHorUncEllipseSemiMinor, 647
pLatitude, 647
pLongitude, 647
pMagneticDeviation, 647
pPrecisionDilution, 647
pSensorDataUsage, 647
pSpeedHorizontal, 647
pSpeedUnc, 647
pSpeedVertical, 647
pSpeedVerticalUnc, 647
pSvUsedforFix, 647
pTechnologyMask, 647
pTimeSrc, 647
pTimeUnc, 647
pTimestampUtc, 647
pVertConfidence, 647
pVertReliability, 648
pVertUnc, 648
pXid, 648
ParamPresenceMask, 647
status, 648
Tlvresult, 648
unpack_loc_CradleMountCallback_Ind
loc.h, 1117
unpack_loc_CradleMountCallback_Ind_t, 648
cradleMountConfigStatus, 648
ParamPresenceMask, 648
unpack_loc_Delete_Assist_Data_t, 648
ParamPresenceMask, 649
Tlvresult, 649
unpack_loc_DeleteAssistData
loc.h, 1117
unpack_loc_DeleteAssistData_Ind
loc.h, 1118
unpack_loc_DeleteAssistData_Ind_t, 649
ParamPresenceMask, 650
status, 650
Tlvresult, 650
unpack_loc_EngineState_Ind
loc.h, 1118
unpack_loc_EngineState_Ind_t, 650
engineState, 651

ParamPresenceMask, 651
Tlvresult, 651
unpack_loc_EventNMEA_Ind
loc.h, 1118
unpack_loc_EventNMEA_Ind_t, 651
NMEAData, 651
ParamPresenceMask, 651
Tlvresult, 651
unpack_loc_EventRegister
loc.h, 1119
unpack_loc_EventRegister_t, 652
ParamPresenceMask, 652
Tlvresult, 652
unpack_loc_EventTimeSyncCallback_Ind
loc.h, 1119
unpack_loc_EventTimeSyncCallback_Ind_t, 652
ParamPresenceMask, 652
timeSyncRefCounter, 652
unpack_loc_GetServer_Ind
loc.h, 1119
unpack_loc_GetServer_Ind_t, 652
pIPv4AddrInfo, 654
pIPv6AddrInfo, 654
pURL, 654
ParamPresenceMask, 654
serverStatus, 654
serverType, 654
Tlvresult, 654
unpack_loc_GnssSvInfo_Ind
loc.h, 1120
unpack_loc_GnssSvInfo_Ind_t, 654
altitudeAssumed, 655
pSatelliteInfo, 655
ParamPresenceMask, 655
Tlvresult, 655
unpack_loc_InjectPositionCallback_Ind
loc.h, 1120
unpack_loc_InjectPositionCallback_Ind_t, 655
ParamPresenceMask, 655
status, 655
unpack_loc_InjectSensorDataCallback_Ind
loc.h, 1120
unpack_loc_InjectSensorDataCallback_Ind_t, 655
injectSensorDataStatus, 657
pAccelSamplesAccepted, 657
pAccelTempSamplesAccepted, 657
pGyroSamplesAccepted, 657
pGyroTempSamplesAccepted, 657
pOpaqueIdentifier, 657
ParamPresenceMask, 657
unpack_loc_InjectTimeSyncDataCallback_Ind
loc.h, 1121
unpack_loc_InjectTimeSyncDataCallback_Ind_t, 657
injectTimeSyncStatus, 658
ParamPresenceMask, 658
unpack_loc_InjectUTCTimeCallback_Ind
loc.h, 1121
unpack_loc_InjectUTCTimeCallback_Ind_t, 658
ParamPresenceMask, 658
status, 658
unpack_loc_PositionRpt_Ind
loc.h, 1121
unpack_loc_PositionRpt_Ind_t, 658
pAltitudeAssumed, 664
pAltitudeWrtEllipsoid, 664
pAltitudeWrtMeanSeaLevel, 664
pFixId, 664
pGpsTime, 664
pHeading, 664
pHeadingUnc, 664
pHorConfidence, 664
pHorReliability, 664
pHorUncCircular, 664
pHorUncEllipseOrientAzimuth, 664
pHorUncEllipseSemiMajor, 664
pHorUncEllipseSemiMinor, 664
pLatitude, 664
pLeapSeconds, 664
pLongitude, 664
pMagneticDeviation, 664
pPrecisionDilution, 665
pSensorDataUsage, 665
pSpeedHorizontal, 665
pSpeedUnc, 665
pSpeedVertical, 665
pSvUsedforFix, 665
pTechnologyMask, 665
pTimeSrc, 665
pTimeUnc, 665
pTimestampUtc, 665
pVertConfidence, 665
pVertReliability, 665
pVertUnc, 665
ParamPresenceMask, 664
sessionId, 665
sessionStatus, 665
Tlvresult, 665
unpack_loc_SLQSLOCGetBestAvailPos
loc.h, 1124
unpack_loc_SLQSLOCGetBestAvailPos_t, 670
ParamPresenceMask, 670
Tlvresult, 670
unpack_loc_SLQSLOCGetServer
loc.h, 1124
unpack_loc_SLQSLOCGetServer_t
loc.h, 1111
unpack_loc_SLQSLOCInjectPosition
loc.h, 1124
unpack_loc_SLQSLOCInjectPosition_t
loc.h, 1111
unpack_loc_SLQSLOCInjectSensorData
loc.h, 1125
unpack_loc_SLQSLOCInjectSensorData_t
loc.h, 1111
unpack_loc_SLQSLOCInjectUTCTime
loc.h, 1125

unpack_loc_SLQSLOCInjectUTCTime_t
loc.h, 1111
unpack_loc_SLQSLOCSetCradleMountConfig
loc.h, 1125
unpack_loc_SLQSLOCSetCradleMountConfig_t
loc.h, 1111
unpack_loc_SLQSLOCSetServer
loc.h, 1126
unpack_loc_SensorStreamingCallback_Ind
loc.h, 1122
unpack_loc_SensorStreamingCallback_Ind_t, 665
pAccelAcceptReady, 666
pAccelTempAcceptReady, 666
pGyroAcceptReady, 666
pGyroTempAcceptReady, 666
ParamPresenceMask, 666
unpack_loc_SetExtPowerConfig_Ind
loc.h, 1122
unpack_loc_SetExtPowerConfig_Ind_t, 666
ParamPresenceMask, 667
status, 667
Tlvresult, 667
unpack_loc_SetExtPowerState
loc.h, 1122
unpack_loc_SetExtPowerState_t, 667
ParamPresenceMask, 667
Tlvresult, 667
unpack_loc_SetOperationMode
loc.h, 1123
unpack_loc_SetOperationMode_Ind
loc.h, 1123
unpack_loc_SetOperationMode_Ind_t, 668
ParamPresenceMask, 668
status, 668
Tlvresult, 668
unpack_loc_SetOperationMode_t, 668
ParamPresenceMask, 669
Tlvresult, 669
unpack_loc_SetServer_Ind
loc.h, 1123
unpack_loc_SetServer_Ind_t, 669
serverStatus, 669
Tlvresult, 669
unpack_loc_Start
loc.h, 1126
unpack_loc_Start_t, 670
ParamPresenceMask, 670
Tlvresult, 670
unpack_loc_Stop
loc.h, 1126
unpack_loc_Stop_t, 670
ParamPresenceMask, 671
Tlvresult, 671
unpack_nas_GetACCOLC
nas.h, 1157
unpack_nas_GetACCOLC_t, 671
pAccolc, 671
ParamPresenceMask, 671
unpack_nas_GetANAAAAuthenticationStatus
nas.h, 1157
unpack_nas_GetANAAAAuthenticationStatus_t, 671
pAuthStatus, 672
ParamPresenceMask, 672
unpack_nas_GetCDMANetworkParameters
nas.h, 1157
unpack_nas_GetCDMANetworkParameters_t, 672
Application, 674
Broadcast, 674
CustomSCP, 674
ForceRev0, 674
ParamPresenceMask, 674
Protocol, 674
RegForeignNID, 674
RegForeignSID, 674
RegHomeSID, 674
Roaming, 674
SCI, 674
SCM, 674
unpack_nas_GetHomeNetwork
nas.h, 1158
unpack_nas_GetHomeNetwork3GPP2
nas.h, 1158
unpack_nas_GetHomeNetwork3GPP2_t, 674
nameSize, 676
pMCC, 676
pMNC, 676
pNID, 676
pName, 676
pNw2DescDisp, 676
pNw2DescEnc, 677
pNw2DescLen, 677
pNw2MCC, 677
pNw2MNC, 677
pNw2Name, 677
pSID, 677
ParamPresenceMask, 676
unpack_nas_GetHomeNetwork_t, 677
mcc, 678
mnc, 678
name, 678
nid, 678
ParamPresenceMask, 678
sid, 678
unpack_nas_GetNetworkPreference
nas.h, 1158
unpack_nas_GetNetworkPreference_t, 678
ActiveTechPref, 679
Duration, 679
ParamPresenceMask, 679
PersistentTechPref, 679
Tlvresult, 679
unpack_nas_GetRFInfo
nas.h, 1159
unpack_nas_GetRFInfo_t, 679
instancesSize, 680
ParamPresenceMask, 680

RFBandInfoElements, 680
unpack_nas_GetServingNetwork
nas.h, 1159
unpack_nas_GetServingNetwork_t, 680
CSDomain, 682
DataCaps, 682
DataCapsLen, 682
MCC, 682
MNC, 682
Name, 682
nameSize, 683
PSDomain, 683
ParamPresenceMask, 683
RAN, 683
RadioIfaces, 683
RadioIfacesSize, 683
RegistrationState, 683
Roaming, 683
unpack_nas_GetServingNetworkCapabilities
nas.h, 1159
unpack_nas_GetServingNetworkCapabilities_t, 683
DataCaps, 684
DataCapsLen, 684
ParamPresenceMask, 684
unpack_nas_GetSignalStrengths
nas.h, 1160
unpack_nas_GetSignalStrengths_t, 684
len, 684
ParamPresenceMask, 684
radio, 684
rssi, 684
unpack_nas_InitiateDomainAttach
nas.h, 1160
unpack_nas_InitiateDomainAttach_t
nas.h, 1137
unpack_nas_PerformNetworkScan
nas.h, 1160
unpack_nas_PerformNetworkScan_t, 685
p3GppNetworkInfoInstances, 686
p3GppNetworkInstanceSize, 686
pPCSInstance, 686
pPCSInstanceSize, 686
pRATINstance, 686
pRATInstanceSize, 686
pScanResult, 686
ParamPresenceMask, 686
unpack_nas_SLQSConfigSigInfo
nas.h, 1164
unpack_nas_SLQSConfigSigInfo_t
nas.h, 1137
unpack_nas_SLQSGetErrorRate
nas.h, 1165
unpack_nas_SLQSGetErrorRate_t, 691
pCDMAFrameErrRate, 692
pGSMBER, 692
pHDRPackErrRate, 692
pWCDMABER, 692
ParamPresenceMask, 692
unpack_nas_SLQSGetNetworkTime
nas.h, 1165
unpack_nas_SLQSGetNetworkTime_t, 692
p3GPP2TimeInfo, 693
p3GPPTimeInfo, 693
ParamPresenceMask, 693
unpack_nas_SLQSGetOperatorNameData
nas.h, 1166
unpack_nas_SLQSGetOperatorNameData_t, 693
pNITZInformation, 694
pOperatorNameString, 694
pOperatorPLMNList, 694
pPLMNNetworkName, 694
pSrvcProviderName, 694
ParamPresenceMask, 694
unpack_nas_SLQSGetPLMNName
nas.h, 1166
unpack_nas_SLQSGetPLMNName_t, 694
longName, 697
longNameCI, 697
longNameEn, 697
longNameLen, 697
longNameSB, 697
ParamPresenceMask, 697
shortName, 697
shortNameCI, 697
shortNameEn, 697
shortNameLen, 697
shortNameSB, 697
spn, 697
spnEncoding, 697
spnLength, 698
unpack_nas_SLQSGetServingSystem
nas.h, 1166
unpack_nas_SLQSGetServingSystem_t, 698
BasestationID, 701
BasestationLatitude, 701
BasestationLongitude, 701
CDMASystemInfoExt, 701
CallBarStatus, 701
CellID, 701
ConcSvcInfo, 702
CurrentPLMN, 702
DTMInd, 702
DataSrvCapabilities, 702
DefaultRoamInd, 702
DetailedSvcInfo, 702
Gpp2TimeZone, 702
GppNetworkDSTAdjustment, 702
GppTimeZone, 702
HdrPersonality, 702
Lac, 702
NetworkID, 702
PRLInd, 702
ParamPresenceMask, 702
RoamIndicatorVal, 702
RoamingIndicatorList, 702
ServingSystem, 702

SystemID, 702
TrackAreaCode, 702
unpack_nas_SLQSGetSignalStrength
nas.h, 1167
unpack_nas_SLQSGetSignalStrength_t, 702
ecioList, 704
ecioListLen, 704
errorRateList, 704
errorRateListLen, 704
Io, 704
ltersrp, 705
ltesnr, 705
ParamPresenceMask, 705
rsrqInfo, 705
rxSignalStrengthList, 705
rxSignalStrengthListLen, 705
signalStrengthReqMask, 705
sinr, 705
unpack_nas_SLQSGetSysInfo
nas.h, 1167
unpack_nas_SLQSGetSysInfo_t, 705
pAddCDMASysInfo, 707
pAddGSMSysInfo, 707
pAddHDRSysInfo, 707
pAddLTESysInfo, 707
pCDMASrvStatusInfo, 708
pCDMASysInfo, 708
pGSMCallBarringSysInfo, 708
pGSMCipherDomainSysInfo, 708
pGSMSrvStatusInfo, 708
pGSMSysInfo, 708
pHDRSrvStatusInfo, 708
pHDRSysInfo, 708
pLTESrvStatusInfo, 708
pLTESysInfo, 708
pLTEVoiceSupportSysInfo, 708
pWCDMASysInfo, 708
ParamPresenceMask, 708
unpack_nas_SLQSGetSysSelectionPref
nas.h, 1167
unpack_nas_SLQSGetSysSelectionPref_t, 708
pBandPref, 713
pEmerMode, 713
pGWAcqOrderPref, 713
pLTEBandPref, 713
pModePref, 713
pNetSelPref, 713
pPRLPref, 713
pRoamPref, 713
pSrvDomainPref, 713
ParamPresenceMask, 713
unpack_nas_SLQSGetSysSelectionPrefExt
nas.h, 1168
unpack_nas_SLQSGetSysSelectionPrefExt_t, 713
pAcqOrderPref, 715
pBandPref, 715
pCiotAcqOrderPref, 715
pCiotLteOpModePref, 715
pEmerMode, 715
pGWAcqOrderPref, 715
pLTEBandPref, 715
pLteM1BandPref, 716
pLteNb1BandPref, 716
pModePref, 716
pNetSelPref, 716
pNr5gBandPref, 716
pPRLPref, 716
pRatDisabledMask, 716
pRoamPref, 716
pSrvDomainPref, 716
unpack_nas_SLQSInitiateNetworkRegistration
nas.h, 1168
unpack_nas_SLQSInitiateNetworkRegistration_t
nas.h, 1137
unpack_nas_SLQSNASGeteDRXParams
nas.h, 1170
unpack_nas_SLQSNASGeteDRXParams_t, 719
pCycleLen, 719
pEdrxEnable, 719
ParamPresenceMask, 719
unpack_nas_SLQSNASSeteDRXParams
nas.h, 1172
unpack_nas_SLQSNASSeteDRXParams_t
nas.h, 1137
unpack_nas_SLQSNASSwiGetChannelLock
nas.h, 1173
unpack_nas_SLQSNASSwiGetChannelLock_t, 725
ParamPresenceMask, 726
unpack_nas_SLQSNASSwiSetChannelLock
nas.h, 1174
unpack_nas_SLQSNASSwiSetChannelLock_t
nas.h, 1137
unpack_nas_SLQSNasConfigSigInfo2
nas.h, 1169
unpack_nas_SLQSNasConfigSigInfo2_t
nas.h, 1137
unpack_nas_SLQSNasGet3GPP2Subscription
nas.h, 1169
unpack_nas_SLQSNasGet3GPP2Subscription_t, 716
pDirNum, 717
ParamPresenceMask, 717
unpack_nas_SLQSNasGetCellLocationInfo
nas.h, 1169
unpack_nas_SLQSNasGetCellLocationInfo_t, 717
pCDMAInfo, 718
pLTEInfoInterfreq, 718
pLTEInfoIntrafreq, 718
pUMTSInfo, 719
ParamPresenceMask, 718
unpack_nas_SLQSNasGetHDRColorCode
nas.h, 1170
unpack_nas_SLQSNasGetHDRColorCode_t, 720
pColorCode, 720
ParamPresenceMask, 720
unpack_nas_SLQSNasGetRFInfo
nas.h, 1170

unpack_nas_SLQSNasGetRFInfo_t, 720
pLTEOperationMode, 721
pRfBandInfoExtFormat, 721
pRfBandwidthInfo, 721
pRfDedicatedBandInfo, 721
rfbandInfoList, 721
Tlvresult, 721
unpack_nas_SLQSNasGetSigInfo
nas.h, 1171
unpack_nas_SLQSNasGetSigInfo_t, 721
CDMASSInfo, 722
GSMSSInfo, 722
HDRSSInfo, 722
LTESSInfo, 722
ParamPresenceMask, 722
unpack_nas_SLQSNasGetTxRxInfo
nas.h, 1171
unpack_nas_SLQSNasGetTxRxInfo_t, 722
pRXChain0Info, 723
pRXChain1Info, 723
pTXInfo, 723
ParamPresenceMask, 723
unpack_nas_SLQSNasIndicationRegisterExt
nas.h, 1171
unpack_nas_SLQSNasIndicationRegisterExt_t
nas.h, 1137
unpack_nas_SLQSNasNetworkTimeCallBack_ind
nas.h, 1172
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t, 723
pDayltSavAdj, 724
pRadioInterface, 724
pTimeZone, 724
ParamPresenceMask, 724
universalTime, 724
unpack_nas_SLQSNasSigInfoCallback_ind
nas.h, 1173
unpack_nas_SLQSNasSigInfoCallback_ind_t, 724
pRscp, 725
ParamPresenceMask, 725
unpack_nas_SLQSNasSwiIndicationRegister
nas.h, 1173
unpack_nas_SLQSNasSwiIndicationRegister_t
nas.h, 1137
unpack_nas_SLQSNasSwiModemStatus
nas.h, 1174
unpack_nas_SLQSNasSwiModemStatus_t, 726
commonInfo, 727
pLTEInfo, 727
ParamPresenceMask, 727
unpack_nas_SLQSNasSwiOTAMessageCallback_ind
nas.h, 1174
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t, 727
Info, 727
Tlvresult, 727
unpack_nas_SLQSNasSysInfoCallback_ind
nas.h, 1175
unpack_nas_SLQSNasTimerCallback_ind
nas.h, 1175
unpack_nas_SLQSNasTimerCallback_ind_t, 728
ParamPresenceMask, 728
t3396_apn, 728
t3396_plmn_id, 728
t3396_val, 728
unpack_nas_SLQSSetBandPreference
nas.h, 1175
unpack_nas_SLQSSetBandPreference_t
nas.h, 1137
unpack_nas_SLQSSetSignalStrengthsCallback
nas.h, 1176
unpack_nas_SLQSSetSignalStrengthsCallback_t
nas.h, 1137
unpack_nas_SLQSSetSysSelectionPref
nas.h, 1176
unpack_nas_SLQSSetSysSelectionPref_t
nas.h, 1137
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind
nas.h, 1176
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t, 728
Info, 729
Tlvresult, 729
unpack_nas_SLQSSetSysSelectionPrefExt
nas.h, 1177
unpack_nas_SLQSSetSysSelectionPrefExt_t
nas.h, 1137
unpack_nas_SLQSSwiGetHDRPersonality
nas.h, 1177
unpack_nas_SLQSSwiGetHDRPersonality_t, 729
pCurrentPersonality, 730
pPersonalityListLength, 730
pProtocolSubtypeElement, 730
ParamPresenceMask, 730
unpack_nas_SLQSSwiGetHDRProtSubtype
nas.h, 1177
unpack_nas_SLQSSwiGetHDRProtSubtype_t, 730
pAppSubType, 730
pCurrentPrsnlty, 731
pPersonalityListLength, 731
ParamPresenceMask, 730
unpack_nas_SLQSSwiGetHRPDStats
nas.h, 1178
unpack_nas_SLQSSwiGetHRPDStats_t, 731
pPilotSetData, 731
pUATI, 731
ParamPresenceMask, 731
unpack_nas_SLQSSwiGetLteCQI
nas.h, 1178
unpack_nas_SLQSSwiGetLteCQI_t, 731
ParamPresenceMask, 732
ValidityCW0, 732
ValidityCW1, 732
unpack_nas_SLQSSwiGetLteSccRxInfo
nas.h, 1178
unpack_nas_SLQSSwiGetLteSccRxInfo_t, 733

pSccRxInfo, 733
ParamPresenceMask, 733
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind
nas.h, 1179
unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t, 733
unpack_nas_SLQSSwiNetworkDebug
nas.h, 1179
unpack_nas_SLQSSwiNetworkDebug_t, 734
pDataStatusDetail, 734
pDeviceConfigDetail, 734
pNetworkStat1x, 734
pNetworkStatEVDO, 734
pObjectVer, 735
ParamPresenceMask, 734
unpack_nas_SLQSSwiPSDetach
nas.h, 1179
unpack_nas_SLQSSwiPSDetach_t
nas.h, 1137
unpack_nas_SLQSSwiRandIndicatorCallback_Ind
nas.h, 1180
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t, 735
ParamPresenceMask, 735
rankIndicatorTlv, 735
unpack_nas_SLQSSysInfoCallback_ind_t, 735
pGSMSysInfo, 738
pHDRSysInfo, 738
pLTESysInfo, 738
pSysInfoNoChange, 738
ParamPresenceMask, 738
unpack_nas_SetACCOLC
nas.h, 1161
unpack_nas_SetACCOLC_t
nas.h, 1137
unpack_nas_SetCDMANetworkParameters
nas.h, 1161
unpack_nas_SetCDMANetworkParameters_t
nas.h, 1137
unpack_nas_SetDataCapabilitiesCallback_ind
nas.h, 1161
unpack_nas_SetDataCapabilitiesCallback_ind_t, 686
dataCaps, 687
dataCapsSize, 687
ParamPresenceMask, 687
unpack_nas_SetEventReportInd
nas.h, 1162
unpack_nas_SetEventReportInd_t, 687
ParamPresenceMask, 688
RFTlv, 688
RRTlv, 688
SLQSSSTlv, 688
SSTlv, 688
unpack_nas_SetLURejectCallback
nas.h, 1162
unpack_nas_SetLURejectCallback_t
nas.h, 1137
unpack_nas_SetNasLTECphyCaIndCallback_ind
nas.h, 1163
unpack_nas_SetNasLTECphyCaIndCallback_ind_t, 688
ParamPresenceMask, 689
unpack_nas_SetNetworkPreference
nas.h, 1163
unpack_nas_SetNetworkPreference_t, 689
ParamPresenceMask, 689
Tlvresult, 689
unpack_nas_SetRFInfoCallback
nas.h, 1163
unpack_nas_SetRFInfoCallback_t
nas.h, 1137
unpack_nas_SetRoamingIndicatorCallback_ind
nas.h, 1164
unpack_nas_SetRoamingIndicatorCallback_ind_t, 689
ParamPresenceMask, 690
roaming, 690
unpack_nas_SetServingSystemCallback_ind
nas.h, 1164
unpack_nas_SetServingSystemCallback_ind_t, 690
ParamPresenceMask, 690
SSInfo, 691
Tlvresult, 691
unpack_nas_SlqsGetLTECphyCAInfo
nas.h, 1165
unpack_nas_SlqsGetLTECphyCAInfo_t, 692
LTECphyCaInfo, 692
ParamPresenceMask, 692
Tlvresult, 692
unpack_omaDmConfigTlv_t, 739
alertmsg, 739
alertmsglength, 739
state, 739
userInputReq, 739
userInputTimeout, 739
unpack_omaDmFotaTlv_t, 739
description, 741
descriptionlength, 741
fwdloadsize, 741
fwloadComplete, 741
namelength, 741
package_name, 741
sessionType, 741
severity, 741
state, 741
updateCompleteStatus, 741
userInputReq, 741
userInputTimeout, 741
version, 741
versionlength, 741
unpack_omaDmNotificationsTlv_t, 742
notification, 742
sessionStatus, 742
unpack_pds_ForceXTRADownload
pds.h, 1195
unpack_pds_ForceXTRADownload_t
pds.h, 1185
unpack_pds_GetPDSDefaults

pds.h, 1195
unpack_pds_GetPDSDefaults_t, 742
pAccuracy, 743
pInterval, 743
pOperation, 743
pTimeout, 743
ParamPresenceMask, 743
unpack_pds_GetPDSState
pds.h, 1195
unpack_pds_GetPDSState_t, 743
pEnabledStatus, 744
pTrackingStatus, 744
ParamPresenceMask, 744
unpack_pds_GetPortAutomaticTracking
pds.h, 1196
unpack_pds_GetPortAutomaticTracking_t, 744
ParamPresenceMask, 745
pbAuto, 745
unpack_pds_GetServiceAutomaticTracking
pds.h, 1196
unpack_pds_GetServiceAutomaticTracking_t, 745
ParamPresenceMask, 745
pbAuto, 745
unpack_pds_GetXTRAAutomaticDownload
pds.h, 1196
unpack_pds_GetXTRAAutomaticDownload_t, 745
pInterval, 746
ParamPresenceMask, 746
pbEnabled, 746
unpack_pds_GetXTRANetwork
pds.h, 1197
unpack_pds_GetXTRANetwork_t, 746
pPreference, 746
ParamPresenceMask, 746
unpack_pds_GetXTRAValidity
pds.h, 1197
unpack_pds_GetXTRAValidity_t, 747
pDuration, 747
pGPSWeek, 747
pGPSWeekOffset, 747
ParamPresenceMask, 747
unpack_pds_PDSInjectTimeReference
pds.h, 1198
unpack_pds_PDSInjectTimeReference_t
pds.h, 1185
unpack_pds_ResetPDSData
pds.h, 1198
unpack_pds_ResetPDSData_t
pds.h, 1185
unpack_pds_SLQSGetAGPSConfig
pds.h, 1201
unpack_pds_SLQSGetAGPSConfig_t, 750
pServerAddress, 751
pServerPort, 751
pServerURL, 751
ParamPresenceMask, 751
unpack_pds_SLQSGetGPSStateInfo
pds.h, 1202
unpack_pds_SLQSGetGPSStateInfo_t, 751
Altitude, 755
EngineState, 755
HorizontalUncertainty, 755
Iono_valid, 755
Latitude, 755
Longitude, 755
ParamPresenceMask, 755
Time_uncert_ms, 755
ValidMask, 755
VerticalUncertainty, 755
unpack_pds_SLQSPDSDeterminePosition
pds.h, 1202
unpack_pds_SLQSPDSDeterminePosition_t
pds.h, 1185
unpack_pds_SLQSPDSInjectAbsoluteTimeReference
pds.h, 1202
unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t
pds.h, 1185
unpack_pds_SLQSPDSInjectPositionData
pds.h, 1203
unpack_pds_SLQSPDSInjectPositionData_t
pds.h, 1185
unpack_pds_SLQSSetAGPSConfig
pds.h, 1203
unpack_pds_SLQSSetAGPSConfig_t
pds.h, 1185
unpack_pds_SLQSSetPositionMethodState
pds.h, 1203
unpack_pds_SLQSSetPositionMethodState_t
pds.h, 1185
unpack_pds_SetEventReport_Ind
pds.h, 1198
unpack_pds_SetEventReport_Ind_t, 747
dLatitude, 749
dLongitude, 749
has_PositionDataNMEA, 749
has_SessionStatus, 749
has_dLatitude, 749
has_dLongitude, 749
has_posSrc, 749
ParamPresenceMask, 749
posSrc, 749
PositionDataNMEA, 749
SessionStatus, 749
unpack_pds_SetEventReportCallback
pds.h, 1199
unpack_pds_SetEventReportCallback_t
pds.h, 1185
unpack_pds_SetPDSDefaults
pds.h, 1199
unpack_pds_SetPDSDefaults_t
pds.h, 1185
unpack_pds_SetPDSState
pds.h, 1199
unpack_pds_SetPDSState_t
pds.h, 1185
unpack_pds_SetPdsState_Ind

pds.h, 1200
unpack_pds_SetPdsState_Ind_t, 749
EnabledStatus, 750
ParamPresenceMask, 750
TlvPresent, 750
TrackingStatus, 750
unpack_pds_SetPortAutomaticTracking
pds.h, 1200
unpack_pds_SetPortAutomaticTracking_t
pds.h, 1185
unpack_pds_SetServiceAutomaticTracking
pds.h, 1200
unpack_pds_SetServiceAutomaticTracking_t
pds.h, 1185
unpack_pds_SetXTRAAutomaticDownload
pds.h, 1201
unpack_pds_SetXTRAAutomaticDownload_t
pds.h, 1185
unpack_pds_SetXTRANetwork
pds.h, 1201
unpack_pds_SetXTRANetwork_t
pds.h, 1185
unpack_pds_StartPDSTrackingSessionExt
pds.h, 1204
unpack_pds_StartPDSTrackingSessionExt_t
pds.h, 1185
unpack_pds_StopPDSTrackingSession
pds.h, 1204
unpack_pds_StopPDSTrackingSession_t
pds.h, 1185
unpack_qmi_t, 756
msgid, 756
type, 756
xid, 756
unpack_qos_BindDataPort
qos.h, 1256
unpack_qos_BindDataPort_t, 756
ParamPresenceMask, 756
unpack_qos_IPv4Addr_t, 757
addr, 757
subnetMask, 757
unpack_qos_IPv6Addr_t, 757
addr, 758
prefixLen, 758
unpack_qos_IPv6TrafCls_t, 758
mask, 758
val, 758
unpack_qos_Port_t, 759
port, 759
range, 759
unpack_qos_QosFlowInfo_t, 759
BearerID, 760
is_RxQFlowGranted_Available, 760
is_TxQFlowGranted_Available, 760
NumRxFilters, 760
NumTxFilters, 761
QFlowState, 761
RxQFilter, 761
RxQFlowGranted, 761
TxQFilter, 761
TxQFlowGranted, 761
unpack_qos_QosFlowInfoState_t, 761
id, 761
isNewFlow, 761
state, 761
unpack_qos_SLQSQosGetNetworkStatus
qos.h, 1256
unpack_qos_SLQSQosGetNetworkStatus_t, 761
NWQoSStatus, 762
ParamPresenceMask, 762
unpack_qos_SLQSQosSwiReadApnExtraParams
qos.h, 1257
unpack_qos_SLQSQosSwiReadApnExtraParams_t, 762
ambr_dl, 763
ambr_dl_ext, 763
ambr_dl_ext2, 763
ambr_ul, 763
ambr_ul_ext, 763
ambr_ul_ext2, 763
apnId, 763
ParamPresenceMask, 763
unpack_qos_SLQSQosSwiReadDataStats
qos.h, 1257
unpack_qos_SLQSQosSwiReadDataStats_t, 764
apnId, 765
numQosFlow, 765
ParamPresenceMask, 765
qosFlow, 765
total_rx_bytes, 765
total_rx_pkt, 765
total_tx_bytes, 765
total_tx_pkt, 765
unpack_qos_SLQSSetQosEventCallback
qos.h, 1258
unpack_qos_SLQSSetQosEventCallback_ind
qos.h, 1258
unpack_qos_SLQSSetQosEventCallback_ind_t, 765
NumFlows, 766
ParamPresenceMask, 766
QosFlowInfo, 766
unpack_qos_SLQSSetQosEventCallback_t, 766
ParamPresenceMask, 766
unpack_qos_SLQSSetQosNWStatusCallback_ind
qos.h, 1259
unpack_qos_SLQSSetQosNWStatusCallback_ind_t, 766
status, 767
unpack_qos_SLQSSetQosPriEventCallback_ind
qos.h, 1259
unpack_qos_SLQSSetQosPriEventCallback_ind_t, 767
event, 767
ParamPresenceMask, 767
unpack_qos_SLQSSetQosStatusCallback_ind
qos.h, 1260
unpack_qos_SLQSSetQosStatusCallback_ind_t, 767

event, 769
id, 769
ParamPresenceMask, 769
reason, 769
status, 769
unpack_qos_Tos_t, 777
mask, 777
val, 777
unpack_qos_dataRate_t, 757
dataRateMax, 757
guaranteedRate, 757
unpack_qos_pktErrRate_t, 758
exponent, 759
multiplier, 759
unpack_qos_swiQosFilter_t, 769
EspSpi, 771
IPv4DstAddr, 771
IPv4SrcAddr, 771
IPv4Tos, 771
IPv6DstAddr, 771
IPv6Label, 771
IPv6SrcAddr, 771
IPv6TrafCls, 771
Id, 771
index, 771
is_EspSpi_Available, 771
is_IPv4DstAddr_Available, 771
is_IPv4SrcAddr_Available, 771
is_IPv4Tos_Available, 771
is_IPv6DstAddr_Available, 771
is_IPv6Label_Available, 771
is_IPv6SrcAddr_Available, 771
is_IPv6TrafCls_Available, 771
is_Id_Available, 771
is_NxtHdrProto_Available, 772
is_Precedence_Available, 772
is_TCPDstPort_Available, 772
is_TCPSrcPort_Available, 772
is_TranDstPort_Available, 772
is_TranSrcPort_Available, 772
is_UDPDstPort_Available, 772
is_UDPSrcPort_Available, 772
NxtHdrProto, 772
Precedence, 772
TCPDstPort, 772
TCPSrcPort, 772
TranDstPort, 772
TranSrcPort, 772
UDPDstPort, 772
UDPSrcPort, 772
version, 772
unpack_qos_swiQosFlow_t, 772
DataRate, 775
index, 775
is_DataRate_Available, 775
is_Jitter_Available, 775
is_Latency_Available, 775
is_LteQci_Available, 775
is_MaxAllowedPktSz_Available, 775
is_MinPolicedPktSz_Available, 775
is_PktErrRate_Available, 775
is_ProfileId3GPP2_Available, 775
is_TokenBucket_Available, 775
is_TrafficClass_Available, 775
is_val_3GPP2Pri_Available, 775
is_val_3GPPImCn_Available, 775
is_val_3GPPSigInd_Available, 776
Jitter, 776
Latency, 776
LteQci, 776
MaxAllowedPktSz, 776
MinPolicedPktSz, 776
PktErrRate, 776
ProfileId3GPP2, 776
TokenBucket, 776
TrafficClass, 776
val_3GPP2Pri, 776
val_3GPPImCn, 776
val_3GPPResResidualBER, 776
val_3GPPSigInd, 776
val_3GPPTraHdlPri, 776
unpack_qos_tokenBucket_t, 776
bucketSz, 777
peakRate, 777
tokenRate, 777
unpack_result_code_only
common.h, 1016
unpack_result_t, 778
ParamPresenceMask, 778
Tlvresult, 778
unpack_rms_GetSMSWake
rms.h, 1262
unpack_rms_GetSMSWake_t, 779
enabled, 779
ParamPresenceMask, 779
wake_mask, 779
unpack_rms_SetSMSWake
rms.h, 1262
unpack_rms_SetSMSWake_t, 779
ParamPresenceMask, 780
unpack_sar_SLQSGetRfSarState
sar.h, 1264
unpack_sar_SLQSGetRfSarState_t, 781
pSarRFState, 781
ParamPresenceMask, 781
unpack_sar_SLQSSetRfSarState
sar.h, 1264
unpack_sar_SLQSSetRfSarState_t
sar.h, 1263
unpack_sms_GetSMSCAddress
sms.h, 1277
unpack_sms_GetSMSCAddress_t, 781
addressSize, 782
pSMSCAddress, 782
pSMSCType, 782
ParamPresenceMask, 782

typeSize, 782
unpack_sms_SLQSDeleteSMS
sms.h, 1279
unpack_sms_SLQSDeleteSMS_t, 786
ParamPresenceMask, 786
unpack_sms_SLQSGetIndicationRegister
sms.h, 1279
unpack_sms_SLQSGetIndicationRegister_t, 786
pGetIndicationRegInfo, 787
ParamPresenceMask, 787
unpack_sms_SLQSGetMessageWaiting
sms.h, 1280
unpack_sms_SLQSGetMessageWaiting_t, 787
pGetMsgWaitingInfoResp, 787
ParamPresenceMask, 787
unpack_sms_SLQSGetSMS
sms.h, 1280
unpack_sms_SLQSGetSMS_t, 787
message, 788
messageFormat, 788
messageSize, 788
messageTag, 788
ParamPresenceMask, 788
unpack_sms_SLQSGetSMSList
sms.h, 1281
unpack_sms_SLQSGetSMSList_t, 789
messageList, 790
messageListSize, 790
ParamPresenceMask, 790
unpack_sms_SLQSGetSmsBroadcastConfig
sms.h, 1280
unpack_sms_SLQSGetSmsBroadcastConfig_t, 788
pBroadcastConfig, 789
pCDMABroadcastConfig, 789
ParamPresenceMask, 789
unpack_sms_SLQSGetTransLayerInfo
sms.h, 1281
unpack_sms_SLQSGetTransLayerInfo_t, 790
pGetTransLayerInfo, 790
ParamPresenceMask, 790
unpack_sms_SLQSGetTransNWRegInfo
sms.h, 1281
unpack_sms_SLQSGetTransNWRegInfo_t, 790
ParamPresenceMask, 791
unpack_sms_SLQSModifySMSStatus
sms.h, 1282
unpack_sms_SLQSModifySMSStatus_t, 791
ParamPresenceMask, 791
unpack_sms_SLQSNWRegInfoCallback_ind
sms.h, 1282
unpack_sms_SLQSNWRegInfoCallback_ind_t, 791
NWRegStat, 792
ParamPresenceMask, 792
unpack_sms_SLQSSendAsyncSMS
sms.h, 1282
unpack_sms_SLQSSendAsyncSMS_t, 792
ParamPresenceMask, 792
unpack_sms_SLQSSetIndicationRegister
sms.h, 1283
unpack_sms_SLQSSetIndicationRegister_t, 792
ParamPresenceMask, 793
unpack_sms_SLQSSetSmsBroadcastActivation
sms.h, 1283
unpack_sms_SLQSSetSmsBroadcastActivation_t, 793
ParamPresenceMask, 793
unpack_sms_SLQSSetSmsBroadcastConfig
sms.h, 1283
unpack_sms_SLQSSetSmsBroadcastConfig_t, 793
ParamPresenceMask, 794
unpack_sms_SLQSSetSmsStorage
sms.h, 1284
unpack_sms_SLQSSetSmsStorage_t, 794
ParamPresenceMask, 794
unpack_sms_SLQSSmsGetMaxStorageSize
sms.h, 1284
unpack_sms_SLQSSmsGetMaxStorageSize_t, 794
pMaxStorageSizeResp, 795
ParamPresenceMask, 794
unpack_sms_SLQSSmsGetMessageProtocol
sms.h, 1284
unpack_sms_SLQSSmsGetMessageProtocol_t, 795
pMessageProtocol, 795
ParamPresenceMask, 795
unpack_sms_SLQSSmsSetRoutes
sms.h, 1285
unpack_sms_SLQSSmsSetRoutes_t, 795
ParamPresenceMask, 795
unpack_sms_SLQSSwiGetSMSStorage
sms.h, 1285
unpack_sms_SLQSSwiGetSMSStorage_t, 796
pSmsStorage, 796
ParamPresenceMask, 796
unpack_sms_SLQSTransLayerInfoCallback_ind
sms.h, 1285
unpack_sms_SLQSTransLayerInfoCallback_ind_t, 796
pTransLayerInfo, 797
ParamPresenceMask, 797
regInd, 797
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind
sms.h, 1286
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind-
_t, 797
causeCode, 799
errorClass, 799
messageID, 799
RPCause, 799
sendStatus, 799
TPCause, 799
userData, 799
unpack_sms_SLQSWmsMemoryFullCallBack_ind
sms.h, 1286
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t,
799
messageMode, 800
ParamPresenceMask, 800
storageType, 800

unpack_sms_SLQSWmsMessageWaitingCallBack_ind
sms.h, 1287
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t, 800
msgWaitInfo, 800
numInstances, 800
ParamPresenceMask, 800
unpack_sms_SaveSMS
sms.h, 1277
unpack_sms_SaveSMS_t, 782
pMessageIndex, 783
ParamPresenceMask, 783
unpack_sms_SendSMS
sms.h, 1278
unpack_sms_SendSMS_t, 783
messageFailureCode, 783
messageID, 783
ParamPresenceMask, 783
unpack_sms_SetNewSMSCallback
sms.h, 1278
unpack_sms_SetNewSMSCallback_ind
sms.h, 1278
unpack_sms_SetNewSMSCallback_ind_t, 783
ETWSTlv, 785
IMSTlv, 785
MMTlv, 785
NewMMTlv, 785
ParamPresenceMask, 785
SMSCTlv, 785
TRMessageTlv, 785
unpack_sms_SetNewSMSCallback_t, 785
ParamPresenceMask, 785
unpack_sms_SetSMSCAddress
sms.h, 1279
unpack_sms_SetSMSCAddress_t, 785
ParamPresenceMask, 786
unpack_swiaudio_SLQSGetM2MAVMute
swiaudio.h, 1293
unpack_swiaudio_SLQSGetM2MAVMute_t, 802
CwtMute, 803
EarMute, 803
MicMute, 803
ParamPresenceMask, 803
unpack_swiaudio_SLQSGetM2MAudioProfile
swiaudio.h, 1293
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 800
CwtMute, 801
EarMute, 801
Generator, 801
MicMute, 801
ParamPresenceMask, 802
Profile, 802
Volume, 802
unpack_swiaudio_SLQSGetM2MAudioVolume
swiaudio.h, 1293
unpack_swiaudio_SLQSGetM2MAudioVolume_t, 802
Level, 802
ParamPresenceMask, 802
unpack_swiaudio_SLQSGetM2MSpkrGain
swiaudio.h, 1294
unpack_swiaudio_SLQSGetM2MSpkrGain_t, 803
ParamPresenceMask, 804
Value, 804
unpack_swiaudio_SLQSSetM2MAVMute
swiaudio.h, 1296
unpack_swiaudio_SLQSSetM2MAVMute_t
swiaudio.h, 1289
unpack_swiaudio_SLQSSetM2MAudioAVCFG
swiaudio.h, 1294
unpack_swiaudio_SLQSSetM2MAudioAVCFG_t
swiaudio.h, 1288
unpack_swiaudio_SLQSSetM2MAudioLPBK
swiaudio.h, 1294
unpack_swiaudio_SLQSSetM2MAudioLPBK_t
swiaudio.h, 1288
unpack_swiaudio_SLQSSetM2MAudioNVDef
swiaudio.h, 1295
unpack_swiaudio_SLQSSetM2MAudioNVDef_t
swiaudio.h, 1288
unpack_swiaudio_SLQSSetM2MAudioProfile
swiaudio.h, 1295
unpack_swiaudio_SLQSSetM2MAudioProfile_t
swiaudio.h, 1289
unpack_swiaudio_SLQSSetM2MAudioVolume
swiaudio.h, 1295
unpack_swiaudio_SLQSSetM2MAudioVolume_t
swiaudio.h, 1289
unpack_swiaudio_SLQSSetM2MSpkrGain
swiaudio.h, 1296
unpack_swiaudio_SLQSSetM2MSpkrGain_t
swiaudio.h, 1289
unpack_swiavms_SLQSAVMSEventReportInd
swiavms.h, 1305
unpack_swiavms_SLQSAVMSEventReportInd_t, 804
pConfig, 805
pConnectionRequest, 805
pDataSessionStatus, 805
pNotification, 805
pPackageID, 805
pRegStatus, 805
pSessionType, 805
ParamPresenceMask, 805
resultcode, 805
unpack_swiavms_SLQSAVMSGetSettings
swiavms.h, 1305
unpack_swiavms_SLQSAVMSGetSettings_t, 805
AutoConnect, 807
AutoReboot, 807
FwAutoSDM, 807
FwPromptUpdate, 808
FwPromptdownload, 807
pAPNInfo, 808
pConnectionRetryTimers, 808
pNotificationStore, 808
pPeroidsInfo, 808
pPollingTimer, 808

ParamPresenceMask, 808
resultcode, 808
unpack_swiavms_SLQSAVMSGetSettings_v2
swiavms.h, 1305
unpack_swiavms_SLQSAVMSGetSettings_v2_t, 808
AutoConnect, 810
FwPromptUpdate, 810
FwPromptdownload, 810
pAutoReboot, 810
pNotificationStore, 810
pPeroidsInfo, 810
pPollingTimer, 810
ParamPresenceMask, 810
resultcode, 810
unpack_swiavms_SLQSAVMSSendSelection
swiavms.h, 1306
unpack_swiavms_SLQSAVMSSendSelection_t, 810
ParamPresenceMask, 811
resultcode, 811
unpack_swiavms_SLQSAVMSSessionGetInfo
swiavms.h, 1306
unpack_swiavms_SLQSAVMSSessionGetInfo_t, 811
pConfig, 811
pNotification, 812
pPackageID, 812
ParamPresenceMask, 811
resultcode, 812
unpack_swiavms_SLQSAVMSSetSettings
swiavms.h, 1307
unpack_swiavms_SLQSAVMSSetSettings_t, 812
ParamPresenceMask, 812
resultcode, 812
unpack_swiavms_SLQSAVMSSetSettings_v2
swiavms.h, 1308
unpack_swiavms_SLQSAVMSSetSettings_v2_t, 813
ParamPresenceMask, 813
resultcode, 813
unpack_swiavms_SLQSAVMSStartSession
swiavms.h, 1308
unpack_swiavms_SLQSAVMSStartSession_t, 813
ParamPresenceMask, 813
resultcode, 813
sessionResponse, 814
unpack_swiavms_SLQSAVMSStopSession
swiavms.h, 1309
unpack_swiavms_SLQSAVMSStopSession_avc2
swiavms.h, 1309
unpack_swiavms_SLQSAVMSStopSession_t, 814
ParamPresenceMask, 814
resultcode, 814
unpack_swiavms_SLQSAvmsSetEventReport
swiavms.h, 1307
unpack_swiavms_SLQSAvmsSetEventReport_t, 812
ParamPresenceMask, 812
resultcode, 812
unpack_swidms_SLQSSwiDmsGetMTU
swidms.h, 1313
unpack_swidms_SLQSSwiDmsGetMTU_t, 814
pMTUSize3gpp, 815
ParamPresenceMask, 815
unpack_swidms_SLQSSwiDmsGetUsbNetNum
swidms.h, 1313
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 815
ParamPresenceMask, 815
usbNetNum, 815
unpack_swidms_SLQSSwiDmsSetMTU
swidms.h, 1314
unpack_swidms_SLQSSwiDmsSetMTU_t, 816
ParamPresenceMask, 816
Tlvresult, 816
unpack_swidms_SLQSSwiDmsSetUsbNetNum
swidms.h, 1314
unpack_swidms_SLQSSwiDmsSetUsbNetNum_t
swidms.h, 1312
unpack_swiloc_SwiLocGetAutoStart
swiloc.h, 1316
unpack_swiloc_SwiLocGetAutoStart_t, 816
fix_rate, 818
fix_rate_reported, 818
fix_type, 818
fix_type_reported, 818
function, 818
function_reported, 818
max_dist, 818
max_dist_reported, 818
max_time, 818
max_time_reported, 818
ParamPresenceMask, 818
unpack_swiloc_SwiLocSetAutoStart
swiloc.h, 1316
unpack_swiloc_SwiLocSetAutoStart_t
swiloc.h, 1315
unpack_swioma_SLQSOMADMAlertCallback
swioma.h, 1322
unpack_swioma_SLQSOMADMAlertCallback_ind
swioma.h, 1322
unpack_swioma_SLQSOMADMAlertCallback_ind_t, 818
eventType, 819
unpack_swioma_SLQSOMADMCancelSession
swioma.h, 1323
unpack_swioma_SLQSOMADMCancelSessionExt
swiomaext.h, 1329
unpack_swioma_SLQSOMADMCancelSessionExt_t
swiomaext.h, 1326
unpack_swioma_SLQSOMADMGetSessionInfo
swioma.h, 1323
unpack_swioma_SLQSOMADMGetSessionInfo_t, 819
Date, 821
DateLength, 821
PkgDescLength, 821
PkgDescription, 821
PkgName, 821
PkgNameLength, 821
RetryCount, 821
SessionState, 821

SessionType, 821
Severity, 821
Source, 821
SourceLength, 821
Status, 821
Time, 821
TimeLength, 821
UpdateCompleteStatus, 821
unpack_swioma_SLQSOMADMGetSettings
swioma.h, 1323
unpack_swioma_SLQSOMADMGetSettings_t, 821
Autosdm, 823
FOTAdownload, 823
FwAutoCheck, 823
unpack_swioma_SLQSOMADMSendSelection
swioma.h, 1324
unpack_swioma_SLQSOMADMSendSelectionExt
swiomaext.h, 1329
unpack_swioma_SLQSOMADMSendSelectionExt_t
swiomaext.h, 1326
unpack_swioma_SLQSOMADMSetSettings
swioma.h, 1324
unpack_swioma_SLQSOMADMSetSettingsExt
swiomaext.h, 1330
unpack_swioma_SLQSOMADMSetSettingsExt_t
swiomaext.h, 1326
unpack_swioma_SLQSOMADMStartSession
swioma.h, 1325
unpack_swioma_SLQSOMADMStartSession_t, 823
FwAvailability, 823
unpack_swioma_SLQSOMADMStartSessionExt
swiomaext.h, 1330
unpack_swioma_SLQSOMADMStartSessionExt_t
swiomaext.h, 1326
unpack_tmd_SLQSTmdDeRegNotMitigationLvl
tmd.h, 1334
unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t, 823
Tlvresult, 824
unpack_tmd_SLQSTmdGetMitigationDevList
tmd.h, 1335
unpack_tmd_SLQSTmdGetMitigationDevList_t, 824
MitigationDevList, 824
MitigationDevListLen, 824
Tlvresult, 824
unpack_tmd_SLQSTmdGetMitigationLvl
tmd.h, 1335
unpack_tmd_SLQSTmdGetMitigationLvl_t, 824
CurrentmitigationLvl, 825
ReqMitigationLvl, 825
Tlvresult, 825
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind
tmd.h, 1336
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t, 825
deviceID, 826
deviceIdLen, 826
lvl, 826
unpack_tmd_SLQSTmdRegNotMitigationLvl
tmd.h, 1336
unpack_tmd_SLQSTmdRegNotMitigationLvl_t, 826
Tlvresult, 826
unpack_uim_ChangePin
uim.h, 1347
unpack_uim_ChangePin_t, 826
pEncryptedPIN1, 827
pIndicationToken, 827
pRemainingRetries, 827
Tlvresult, 827
unpack_uim_GetCardStatus
uim.h, 1347
unpack_uim_GetCardStatus_t, 827
pCardStatus, 827
pHotSwapStatus, 827
Tlvresult, 828
unpack_uim_ReadTransparent
uim.h, 1348
unpack_uim_ReadTransparent_t, 828
pCardResult, 828
pEncryptedData, 828
pIndicationToken, 828
pReadResult, 828
Tlvresult, 828
unpack_uim_SLQSUIMAuthenticate
uim.h, 1349
unpack_uim_SLQSUIMAuthenticate_t, 830
pAuthenticateResult, 830
pCardResult, 831
pIndicationToken, 831
unpack_uim_SLQSUIMDepersonalization
uim.h, 1349
unpack_uim_SLQSUIMDepersonalization_t, 831
pRemainingRetries, 831
unpack_uim_SLQSUIMEventRegister
uim.h, 1349
unpack_uim_SLQSUIMEventRegister_t, 831
eventMask, 831
unpack_uim_SLQSUIMGetConfiguration
uim.h, 1350
unpack_uim_SLQSUIMGetConfiguration_t, 832
pAutoSelection, 832
pHaltSubscription, 832
pPersonalizationStatus, 832
unpack_uim_SLQSUIMGetFileAttributes
uim.h, 1350
unpack_uim_SLQSUIMGetFileAttributes_t, 832
pCardResult, 833
pFileAttributes, 833
pIndicationToken, 833
unpack_uim_SLQSUIMGetSlotsStatus
uim.h, 1350
unpack_uim_SLQSUIMGetSlotsStatus_t, 833
pNumberOfPhySlot, 834
pUimSlotsStatus, 834
unpack_uim_SLQSUIMPowerDown
uim.h, 1351
unpack_uim_SLQSUIMPowerUp

uim.h, 1351
unpack_uim_SLQSUIMRefreshCallback_Ind
uim.h, 1351
unpack_uim_SLQSUIMRefreshCallback_Ind_t, 834
refreshEvent, 834
TlvPresent, 834
unpack_uim_SLQSUIMRefreshComplete
uim.h, 1352
unpack_uim_SLQSUIMRefreshGetLastEvent
uim.h, 1352
unpack_uim_SLQSUIMRefreshGetLastEvent_t, 834
pRefreshEvent, 834
unpack_uim_SLQSUIMRefreshOK
uim.h, 1352
unpack_uim_SLQSUIMRefreshRegister
uim.h, 1353
unpack_uim_SLQSUIMReset
uim.h, 1353
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind
uim.h, 1353
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t, 835
unpack_uim_SLQSUIMSwitchSlot
uim.h, 1354
unpack_uim_SetPinProtection
uim.h, 1348
unpack_uim_SetPinProtection_t, 829
pEncryptedPIN1, 829
pIndicationToken, 829
pRemainingRetries, 829
Tlvresult, 829
unpack_uim_SetUimSlotStatusChangeCallback_ind
uim.h, 1348
unpack_uim_SetUimSlotStatusChangeCallback_ind_t, 829
bNumberOfPhySlots, 830
slotsstatusChange, 830
unpack_uim_UnblockPin
uim.h, 1354
unpack_uim_UnblockPin_t, 835
pEncryptedPIN1, 836
pIndicationToken, 836
pRemainingRetries, 836
Tlvresult, 836
unpack_uim_VerifyPin
uim.h, 1354
unpack_uim_VerifyPin_t, 836
pEncryptedPIN1, 836
pIndicationToken, 836
pRemainingRetries, 836
Tlvresult, 836
unpack_valid_nas_GetCDMANetworkParameters
nas.h, 1180
unpack_valid_nas_SLQSGetServingSystem
nas.h, 1180
unpack_valid_nas_SLQSGetSignalStrength
nas.h, 1181
unpack_valid_nas_SLQSNasGetSigInfo
nas.h, 1182
unpack_voice_AnswerUSSD
voice.h, 1373
unpack_voice_CancelUSSD
voice.h, 1373
unpack_voice_DTMFEventCallback_ind
voice.h, 1373
unpack_voice_DTMFEventCallback_ind_t, 839
DTMFInformation, 840
pOffLength, 840
pOnLength, 840
unpack_voice_OTASPStatusCallback_ind
voice.h, 1374
unpack_voice_OTASPStatusCallback_ind_t, 840
callID, 841
OTASPStatus, 841
unpack_voice_OriginateUSSD
voice.h, 1374
unpack_voice_SLQSOriginateUSSD
voice.h, 1374
unpack_voice_SLQSOriginateUSSD_t, 841
pAlphaIDInfo, 844
pCCSuppsType, 844
pCallId, 844
pCcResultType, 844
pUSSDInfo, 844
pfailureCause, 844
unpack_voice_SLQSVoiceALSSelectLine
voice.h, 1375
unpack_voice_SLQSVoiceALSSetLineSwitching
voice.h, 1375
unpack_voice_SLQSVoiceAnswerCall
voice.h, 1375
unpack_voice_SLQSVoiceAnswerCall_t, 844
pCallId, 845
unpack_voice_SLQSVoiceBindSubscription
voice.h, 1376
unpack_voice_SLQSVoiceBurstDTMF
voice.h, 1376
unpack_voice_SLQSVoiceBurstDTMF_t, 845
pCallID, 845
unpack_voice_SLQSVoiceDialCall
voice.h, 1376
unpack_voice_SLQSVoiceDialCall_t, 845
pAlphaIDInfo, 846
pCCResultType, 846
pCCSUPSType, 846
pCallID, 846
unpack_voice_SLQSVoiceEndCall
voice.h, 1377
unpack_voice_SLQSVoiceEndCall_t, 846
pCallId, 847
unpack_voice_SLQSVoiceGetAllCallInfo
voice.h, 1377
unpack_voice_SLQSVoiceGetAllCallInfo_t, 847
pArrAlertingPattern, 849
pArrAlertingType, 849
pArrAlphaID, 849

pArrCallEndReason, 849
pArrCallInfo, 849
pArrCalledPartyNum, 849
pArrConnectPartyNum, 849
pArrDiagInfo, 849
pArrRedirPartyNum, 849
pArrRemotePartyName, 849
pArrRemotePartyNum, 849
pArrSvcOption, 849
pArrUUSInfo, 849
pVoicePrivacy, 849
unpack_voice_SLQSVoiceGetCLIP
voice.h, 1379
unpack_voice_SLQSVoiceGetCLIP_t, 857
pAlphaIDInfo, 858
pCCResType, 858
pCallID, 858
pFailCause, 858
unpack_voice_SLQSVoiceGetCLIR
voice.h, 1379
unpack_voice_SLQSVoiceGetCLIR_t, 858
pAlphaIDInfo, 859
pCCResType, 859
pCallID, 859
pFailCause, 860
unpack_voice_SLQSVoiceGetCNAP
voice.h, 1379
unpack_voice_SLQSVoiceGetCNAP_t, 860
pAlphaIDInfo, 861
pCCResType, 861
pCallID, 861
pFailCause, 861
unpack_voice_SLQSVoiceGetCOLP
voice.h, 1380
unpack_voice_SLQSVoiceGetCOLP_t, 861
pAlphaIDInfo, 862
pCCResType, 862
pCallID, 862
pFailCause, 862
unpack_voice_SLQSVoiceGetCOLR
voice.h, 1380
unpack_voice_SLQSVoiceGetCOLR_t, 862
pAlphaIDInfo, 864
pCCResType, 864
pCallID, 864
pFailCause, 864
unpack_voice_SLQSVoiceGetCallBarring
voice.h, 1377
unpack_voice_SLQSVoiceGetCallBarring_t, 849
pAlphaIDInfo, 851
pCCResType, 851
pCallID, 851
pFailCause, 851
pSvcClass, 851
unpack_voice_SLQSVoiceGetCallForwardingStatus
voice.h, 1378
unpack_voice_SLQSVoiceGetCallForwardingStatus_t, 851
pAlphaIDInfo, 852
pCCResType, 852
pCallID, 852
pFailCause, 852
unpack_voice_SLQSVoiceGetCallInfo
voice.h, 1378
unpack_voice_SLQSVoiceGetCallInfo_t, 852
pAlertType, 855
pAlertingPattern, 855
pAlphaIDInfo, 855
pCallInfo, 855
pConnectNumInfo, 855
pDiagInfo, 855
pOTASPStatus, 855
pRemotePartyName, 855
pRemotePartyNum, 855
pSrvOpt, 855
pUUSInfo, 855
pVoicePrivacy, 855
unpack_voice_SLQSVoiceGetCallWaiting
voice.h, 1378
unpack_voice_SLQSVoiceGetCallWaiting_t, 856
pAlphaIDInfo, 857
pCCResType, 857
pCallID, 857
pFailCause, 857
pSvcClass, 857
unpack_voice_SLQSVoiceGetConfig
voice.h, 1380
unpack_voice_SLQSVoiceGetConfig_t, 864
pAirTimerCnt, 866
pAutoAnswerStat, 866
pCurAMRConfig, 866
pCurPrefVoiceSO, 866
pCurVoiceDomainPref, 866
pCurVoicePrivacyPref, 866
pCurrTTYMode, 866
pRoamTimerCnt, 866
unpack_voice_SLQSVoiceIndicationRegister
voice.h, 1381
unpack_voice_SLQSVoiceManageCalls
voice.h, 1381
unpack_voice_SLQSVoiceManageCalls_t, 866
pFailCause, 866
unpack_voice_SLQSVoiceOrigUSSDNoWait
voice.h, 1381
unpack_voice_SLQSVoiceSUPSCallback_ind
voice.h, 1384
unpack_voice_SLQSVoiceSUPSCallback_ind_t, 872
pDataSrc, 874
pFailCause, 874
pReason, 874
pSvcClass, 874
unpack_voice_SLQSVoiceSendFlash
voice.h, 1382
unpack_voice_SLQSVoiceSendFlash_t, 866
pCallID, 867
unpack_voice_SLQSVoiceSetCallBarringPassword

voice.h, 1382
unpack_voice_SLQSVoiceSetCallBarringPassword_t, 867
pAlphaIDInfo, 868
pCCResType, 868
pCallID, 868
pFailCause, 868
unpack_voice_SLQSVoiceSetConfig
voice.h, 1382
unpack_voice_SLQSVoiceSetConfig_t, 868
pAirTimerStatus, 869
pAutoAnsStatus, 869
pPrefVoiceSOStatus, 869
pRoamTimerStatus, 870
pTTYConfigStatus, 870
pVoiceDomainPrefStatus, 870
unpack_voice_SLQSVoiceSetPreferredPrivacy
voice.h, 1383
unpack_voice_SLQSVoiceSetSUPSService
voice.h, 1383
unpack_voice_SLQSVoiceSetSUPSService_t, 870
pCallID, 871
pFailCause, 871
unpack_voice_SLQSVoiceStartContDTMF
voice.h, 1383
unpack_voice_SLQSVoiceStartContDTMF_t, 871
pCallID, 871
unpack_voice_SLQSVoiceStopContDTMF
voice.h, 1384
unpack_voice_SLQSVoiceStopContDTMF_t, 871
callID, 872
unpack_voice_SUPSNotificationCallback_ind
voice.h, 1384
unpack_voice_SUPSNotificationCallback_ind_t, 874
callID, 876
notifType, 876
pCUGIndex, 876
pECTNum, 876
unpack_voice_USSDNotificationCallback_ind
voice.h, 1385
unpack_voice_USSDNotificationCallback_ind_t, 876
notification_Type, 876
unpack_voice_VoiceInfoRecCallback_ind
voice.h, 1385
unpack_voice_VoiceInfoRecCallback_ind_t, 876
callID, 878
pCLIRCause, 878
pCallWaitInd, 878
pCalledPartyInfo, 878
pCallerIDInfo, 878
pCallerNameInfo, 878
pCallingPartyInfo, 878
pConnectNumInfo, 878
pDispInfo, 878
pExtDispInfo, 878
pExtDispRecInfo, 878
pLineCtrlInfo, 878
pNSSAudioCtrl, 878
pNSSRelease, 879
pRedirNumInfo, 879
pSignalInfo, 879
unpack_voice_allCallStatusCallback_ind
voice.h, 1372
unpack_voice_allCallStatusCallback_ind_t, 837
arrCallInfomation, 839
pArrAlertingPattern, 839
pArrAlertingType, 839
pArrAlphaID, 839
pArrCallEndReason, 839
pArrCalledPartyNum, 839
pArrConnectPartyNum, 839
pArrDiagInfo, 839
pArrRedirPartyNum, 839
pArrRemotePartyName, 839
pArrRemotePartyNum, 839
pArrSvcOption, 839
unpack_voice_voicePrivacyChangeCallback_ind
voice.h, 1385
unpack_voice_voicePrivacyChangeCallback_ind_t, 879
callID, 879
voicePrivacy, 879
unpack_wds_DHCPv4ClientLease_ind
wds.h, 1413
unpack_wds_DHCPv4ClientLease_ind_t, 879
DHCPv4LeaseOptTlv, 880
DHCPv4LeaseStateTlv, 880
IPv4AddrTlv, 880
ProfileIdTlv, 880
unpack_wds_DHCPv4ClientLeaseChange
wds.h, 1414
unpack_wds_GetAutoconnect
wds.h, 1414
unpack_wds_GetAutoconnect_t, 880
psetting, 880
unpack_wds_GetByteTotals
wds.h, 1414
unpack_wds_GetByteTotals_t, 880
pRXTotalBytes, 881
pTXTotalBytes, 881
unpack_wds_GetConnectionRate
wds.h, 1415
unpack_wds_GetConnectionRate_t, 881
currentChannelRXRate, 881
currentChannelTXRate, 881
maxChannelRXRate, 881
maxChannelTXRate, 881
unpack_wds_GetDataBearerTechnology
wds.h, 1415
unpack_wds_GetDataBearerTechnology_t, 882
pDataBearer, 882
unpack_wds_GetDefaultProfile
wds.h, 1415
unpack_wds_GetDefaultProfile_t, 882
apnname, 884
apnsize, 884
auth, 884

ipaddr, 884
ipaddrv6, 884
name, 884
namesize, 884
pdptype, 885
pridns, 885
pridnsv6, 885
secdns, 885
secdnsv6, 885
username, 885
usersize, 885
unpack_wds_GetDefaultProfileNum
wds.h, 1416
unpack_wds_GetDefaultProfileNum_t, 885
index, 885
unpack_wds_GetDormancyState
wds.h, 1416
unpack_wds_GetDormancyState_t, 885
dormancyState, 886
unpack_wds_GetLastMobileIPError
wds.h, 1417
unpack_wds_GetLastMobileIPError_t, 886
error, 886
unpack_wds_GetMobileIP
wds.h, 1417
unpack_wds_GetMobileIP_t, 886
mipMode, 886
unpack_wds_GetMobileIPProfile
wds.h, 1417
unpack_wds_GetMobileIPProfile_t, 887
AAASPI, 888
AAAState, 888
address, 888
enabled, 888
HASPI, 888
HAState, 888
NAI, 888
naiSize, 888
primaryHA, 888
revTunneling, 888
secondaryHA, 888
unpack_wds_GetPacketStatistics
wds.h, 1418
unpack_wds_GetPacketStatistics_t, 888
pRXDroppedCount, 890
pRXOKBytesLastCall, 890
pRXOkBytesCount, 890
pRXPacketErrors, 890
pRXPacketOverflows, 890
pRXPacketSuccesses, 890
pTXDroppedCount, 890
pTXOKBytesLastCall, 890
pTXOkBytesCount, 890
pTXPacketErrors, 890
pTXPacketOverflows, 890
pTXPacketSuccesses, 890
unpack_wds_GetPacketStatus
wds.h, 1418
unpack_wds_GetPacketStatus_t, 890
rXDroppedCount, 892
rXOKBytesLastCall, 892
rXOkBytesCount, 892
rXPacketErrors, 892
rXPacketOverflows, 892
rXPacketSuccesses, 892
tXDroppedCount, 892
tXOKBytesLastCall, 892
tXOkBytesCount, 892
tXPacketErrors, 892
tXPacketOverflows, 892
tXPacketSuccesses, 892
unpack_wds_GetSessionDuration
wds.h, 1418
unpack_wds_GetSessionDuration_t, 892
callDuration, 893
unpack_wds_GetSessionState
wds.h, 1419
unpack_wds_GetSessionState_t, 893
connectionStatus, 893
unpack_wds_RMSetTransferStatistics
wds.h, 1419
unpack_wds_RMSetTransferStatistics_t, 893
unpack_wds_RMTransferStatistics_ind
wds.h, 1419
unpack_wds_SLQSCreateProfile
wds.h, 1422
unpack_wds_SLQSCreateProfile_t, 894
pCreateProfileOut, 894
pProfileID, 894
Tlvresult, 894
unpack_wds_SLQSDUNCallInfoCallBack_ind
wds.h, 1423
unpack_wds_SLQSDUNCallInfoCallBack_ind_t, 895
unpack_wds_SLQSDeleteProfile
wds.h, 1422
unpack_wds_SLQSDeleteProfile_t, 894
extendedErrorCode, 895
unpack_wds_SLQSGet3GPPConfigItem
wds.h, 1423
unpack_wds_SLQSGet3GPPConfigItem_t, 896
_3gppRelease, 897
defaultPDNEnabled, 897
LTEAttachProfile, 897
profileList, 898
unpack_wds_SLQSGetCurrDataSystemStat
wds.h, 1423
unpack_wds_SLQSGetCurrDataSystemStat_t, 898
currNetworkInfo, 898
networkInfoLen, 898
prefNetwork, 898
unpack_wds_SLQSGetCurrentChannelRate
wds.h, 1424
unpack_wds_SLQSGetCurrentChannelRate_t, 898
max_channel_rx_rate, 899
max_channel_tx_rate, 899
unpack_wds_SLQSGetDUNCallInfo

wds.h, 1424
unpack_wds_SLQSGetDUNCallInfo_t, 900
callEndReason, 902
channelRate, 902
connectionStatus, 902
dataBearerTech, 902
dormancyStatus, 903
lastCallDataBearerTech, 903
mdmCallDurationActive, 903
rxOKBytesCount, 903
txOKBytesCount, 903
unpack_wds_SLQSGetDataBearerTechnology
wds.h, 1424
unpack_wds_SLQSGetDataBearerTechnology_t, 899
curDataBearerTechnology, 900
dataBearerMask, 900
lastCallDataBearerTechnology, 900
unpack_wds_SLQSGetProfileSettings
wds.h, 1425
unpack_wds_SLQSGetProfileSettings_t, 903
pProfileSettings, 903
ProfileType, 903
Tlvresult, 903
unpack_wds_SLQSGetRuntimeSettings
wds.h, 1425
unpack_wds_SLQSGetRuntimeSettings_t, 903
APNName, 905
Authentication, 905
DomainList, 905
GPRSGrantedQoS, 905
GWAddressV4, 906
IMCNflag, 906
IPFamilyPreference, 906
IPV6AddrInfo, 906
IPV6GWAddrInfo, 906
IPv4, 906
Mtu, 906
PDPType, 906
PrimaryDNSV4, 906
PrimaryDNSV6, 906
ProfileID, 906
ProfileName, 906
SecondaryDNSV4, 906
SecondaryDNSV6, 906
ServerAddrList, 906
SubnetMaskV4, 906
Technology, 906
UMTSGrantedQoS, 906
Username, 906
unpack_wds_SLQSModifyProfile
wds.h, 1425
unpack_wds_SLQSModifyProfile_t, 906
pExtErrorCode, 907
unpack_wds_SLQSResetPacketStatics
wds.h, 1426
unpack_wds_SLQSSGetDHCPv4ClientConfig
wds.h, 1427
unpack_wds_SLQSSGetDHCPv4ClientConfig_t, 914
pHwConfig, 915
unpack_wds_SLQSSGetLoopback
wds.h, 1428
unpack_wds_SLQSSGetLoopback_t, 915
ByteLoopbackMode, 915
ByteLoopbackMultiplier, 915
unpack_wds_SLQSSSetDHCPv4ClientConfig
wds.h, 1428
unpack_wds_SLQSSSetLoopback
wds.h, 1428
unpack_wds_SLQSSet3GPPConfigItem
wds.h, 1426
unpack_wds_SLQSSetIPFamilyPreference
wds.h, 1426
unpack_wds_SLQSSetIPFamilyPreference_t, 907
Tlvresult, 907
unpack_wds_SLQSSetPacketSrvStatusCallback
wds.h, 1426
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 907
bearerID, 909
conn_status, 909
ipFamily, 909
reconfigReqd, 909
sessionEndReason, 909
techName, 909
verboseSessnEndReason, 909
verboseSessnEndReasonType, 909
unpack_wds_SLQSSetWdsEventCallback
wds.h, 1427
unpack_wds_SLQSSetWdsEventCallback_ind
wds.h, 1427
unpack_wds_SLQSSetWdsEventCallback_ind_t, 909
currDBTechAvail, 913
currNWInfo, 913
dBTechAvail, 913
dBTechnology, 914
dBtechExtAvail, 913
dBtechnologyExt, 914
dataSysStatAvail, 913
dormancyStatAvail, 914
dormancyStatus, 914
mipStatus, 914
mipstatAvail, 914
netInfoLen, 914
prefNetwork, 914
ratMask, 914
rx_bytes, 914
rx_pkts, 914
soMask, 914
tx_bytes, 914
tx_pkts, 914
xferStatAvail, 914
unpack_wds_SLQSStartDataSession
wds.h, 1429
unpack_wds_SLQSStartDataSession_t, 915
pFailureReason, 916
pVerboseFailReasonType, 916
pVerboseFailureReason, 916

psid, 916
unpack_wds_SLQSStopDataSession
wds.h, 1429
unpack_wds_SLQSWdsGoActive
wds.h, 1429
unpack_wds_SLQSWdsGoDormant
wds.h, 1430
unpack_wds_SLQSWdsSetEventReport
wds.h, 1430
unpack_wds_SLQSWdsSwiPDPRuntimeSettings
wds.h, 1430
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t, 916
apnName, 918
bearerId, 918
contextId, 918
ipv4Address, 918
ipv4GWAddress, 918
ipv6Address, 918
ipv6GWAddress, 918
unpack_wds_SetAutoconnect
wds.h, 1420
unpack_wds_SetDefaultProfile
wds.h, 1420
unpack_wds_SetDefaultProfileNum
wds.h, 1420
unpack_wds_SetMobileIP
wds.h, 1421
unpack_wds_SetMobileIPParameters
wds.h, 1421
unpack_wds_SetMobileIPProfile
wds.h, 1421
unpack_wds_SetMobileIPProfile_t, 893
unpack_wds_SetMuxID
wds.h, 1422
UnpackQmiProfileInfo
wds.h, 1393
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo, 919
bBinaryType, 920
bSerity, 920
bState, 920
bUserInputRequest, 920
szDescription, 920
szName, 920
szVersion, 921
TlvPresent, 921
ulPkgDownloadComplete, 921
ulPkgDownloadSize, 921
wDescriptionLength, 921
wNameLength, 921
wUpdateCompeteStatus, 921
wUserInputTimeout, 921
wVersionLength, 921
UnpackSwiAvmsEventReportConfig, 921
bState, 922
bUserInputRequest, 922
szAlertMsg, 922
TlvPresent, 922
wAlertMsgLength, 922
wUserInputTimeout, 922
UnpackSwiAvmsEventReportConnectionRequest, 922
bUserInputRequest, 922
TlvPresent, 922
wUserInputTimeout, 922
UnpackSwiAvmsEventReportDataSessionStatus, 922
bType, 923
TlvPresent, 923
wErrorCode, 923
UnpackSwiAvmsEventReportHTTPStatus, 923
TlvPresent, 924
wHTTPStatus, 924
UnpackSwiAvmsEventReportNotification, 925
bNotification, 925
TlvPresent, 926
wSessionStatus, 926
UnpackSwiAvmsEventReportPackageID, 926
bPackageID, 926
TlvPresent, 926
UnpackSwiAvmsEventReportRegStatus, 926
bRegStatus, 926
TlvPresent, 926
UnpackSwiAvmsEventReportSessionType, 927
bType, 927
TlvPresent, 927
UnpackSwiAvmsEventReportWAMSParamChange, 927
TlvPresent, 928
wWamsChangeMask, 928
unpackWdsProfileParam, 928
SlqsProfile3GPP, 928
SlqsProfile3GPP2, 928
upLink
voice_NSSAudioCtrl, 965
UpdateCompleteStatus
unpack_swioma_SLQSOMADMGetSessionInfo_t, 821
updateCompleteStatus
unpack_omaDmFotaTlv_t, 741
upinRetries
slotInf, 497
uim_slotInfo, 544
upinState
slotInf, 497
uim_slotInfo, 544
upukRetries
slotInf, 497
uim_slotInfo, 544
urlAddr
loc_URLAddrInfo, 139
usageMask
loc_sensorDataUsage, 136
UsbMTUSize
swidms_usbMTUSizeTlv, 518
usbNetNum
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t, 815

userData
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t, 799
userInputReq
unpack_omaDmConfigTlv_t, 739
unpack_omaDmFotaTlv_t, 741
userInputTimeout
unpack_omaDmConfigTlv_t, 739
unpack_omaDmFotaTlv_t, 741
Username
unpack_wds_SLQSGetRuntimeSettings_t, 906
username
unpack_wds_GetDefaultProfile_t, 885
usersize
unpack_wds_GetDefaultProfile_t, 885
ussDCS
pack_voice_SLQSOriginateUSSD_t, 441
voice_USSInfo, 975
ussData
pack_voice_SLQSOriginateUSSD_t, 441
voice_USSInfo, 975
ussLen
pack_voice_SLQSOriginateUSSD_t, 441
voice_USSInfo, 975
UtRat
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 641
UtRatVal
imsa_UtRatInfo, 87
UtService
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 641
UtServiceRat
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
UtServiceStatus
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
UtSvcStatus
imsa_UtSvcStatusInfo, 87
uusInfo
voice_allCallsUUSInfo, 930

VDOP
loc_precisionDilution, 133
val
unpack_qos_IPv6TrafCls_t, 758
unpack_qos_Tos_t, 777
val_3GPP2Pri
unpack_qos_swiQosFlow_t, 776
val_3GPPImCn
unpack_qos_swiQosFlow_t, 776
val_3GPPResResidualBER
unpack_qos_swiQosFlow_t, 776
val_3GPPSigInd
unpack_qos_swiQosFlow_t, 776
val_3GPPTraHdlPri
unpack_qos_swiQosFlow_t, 776
ValidMask
unpack_pds_SLQSGetGPSStateInfo_t, 755
validMask
loc_satelliteInfo, 135
ValidityCW0
unpack_nas_SLQSSwiGetLteCQI_t, 732
ValidityCW1
unpack_nas_SLQSSwiGetLteCQI_t, 732
Value
pack_swiaudio_SLQSSetM2MSpkrGain_t, 412
unpack_swiaudio_SLQSGetM2MSpkrGain_t, 804
value
pack_dms_UIMSetPINProtection_t, 318
pack_dms_UIMVerifyPIN_t, 319
value_length
DMScustSettingInfo, 61
pack_dms_SetCustFeaturesV2_t, 308
verboseSessnEndReason
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 909
verboseSessnEndReasonType
unpack_wds_SLQSSetPacketSrvStatusCallback_t, 909
verifyLeft
uim_personalizationStatus, 536
uim_remainingRetries, 540
verifyPIN
pack_uim_VerifyPin_t, 439
verifyRetriesLeft
unpack_dms_UIMGetControlKeyStatus_t, 612
unpack_dms_UIMSetControlKeyProtection_t, 615
unpack_dms_UIMSetPINProtection_t, 616
version
unpack_omaDmFotaTlv_t, 741
unpack_qos_swiQosFilter_t, 772
versionString
pack_dms_SLQSSwiSetOSInfo_t, 314
unpack_dms_SLQSSwiGetOSInfo_t, 607
versionlength
unpack_omaDmFotaTlv_t, 741
vertConfidence
pack_loc_SLQSLOCInjectPosition_t, 340
vertReliability
pack_loc_SLQSLOCInjectPosition_t, 340
vertUnc
pack_loc_SLQSLOCInjectPosition_t, 340
VerticalUncertainty
unpack_pds_SLQSGetGPSStateInfo_t, 755
VirtStream
nas_protocolSubtypeElement, 232
voice.h
LITE_VOICE_SUPS_SRV_CLASS_DATA, 1360
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITASYNC, 1360
LITE_VOICE_SUPS_SRV_CLASS_DATACIRCUITSYNC, 1360
LITE_VOICE_SUPS_SRV_CLASS_FAX, 1360
LITE_VOICE_SUPS_SRV_CLASS_NONE, 1360
LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS, 1360

LITE_VOICE_SUPS_SRV_CLASS_PADACCESS, 1360
LITE_VOICE_SUPS_SRV_CLASS_SMS, 1360
LITE_VOICE_SUPS_SRV_CLASS_VOICE, 1360
voice.h, 1355
liteServiceClassInformation, 1360
MAXVOICEUSSDLENGTH, 1360
pack_voice_AnswerUSSD, 1361
pack_voice_CancelUSSD, 1361
pack_voice_OriginateUSSD, 1361
pack_voice_SLQSOriginateUSSD, 1362
pack_voice_SLQSVoiceALSSelectLine, 1362
pack_voice_SLQSVoiceALSSetLineSwitching, 1362
pack_voice_SLQSVoiceAnswerCall, 1363
pack_voice_SLQSVoiceBindSubscription, 1363
pack_voice_SLQSVoiceBurstDTMF, 1363
pack_voice_SLQSVoiceDialCall, 1364
pack_voice_SLQSVoiceEndCall, 1364
pack_voice_SLQSVoiceGetAllCallInfo, 1365
pack_voice_SLQSVoiceGetCLIP, 1366
pack_voice_SLQSVoiceGetCLIR, 1367
pack_voice_SLQSVoiceGetCNAP, 1367
pack_voice_SLQSVoiceGetCOLP, 1367
pack_voice_SLQSVoiceGetCOLR, 1368
pack_voice_SLQSVoiceGetCallBarring, 1365
pack_voice_SLQSVoiceGetCallForwardingStatus, 1365
pack_voice_SLQSVoiceGetCallInfo, 1366
pack_voice_SLQSVoiceGetCallWaiting, 1366
pack_voice_SLQSVoiceGetConfig, 1368
pack_voice_SLQSVoiceIndicationRegister, 1368
pack_voice_SLQSVoiceManageCalls, 1369
pack_voice_SLQSVoiceOrigUSSDNoWait, 1369
pack_voice_SLQSVoiceSendFlash, 1370
pack_voice_SLQSVoiceSetCallBarringPassword, 1370
pack_voice_SLQSVoiceSetConfig, 1370
pack_voice_SLQSVoiceSetPreferredPrivacy, 1371
pack_voice_SLQSVoiceSetSUPSService, 1371
pack_voice_SLQSVoiceStartContDTMF, 1371
pack_voice_SLQSVoiceStopContDTMF, 1372
unpack_voice_AnswerUSSD, 1373
unpack_voice_CancelUSSD, 1373
unpack_voice_DTMFEventCallback_ind, 1373
unpack_voice_OTASPStatusCallback_ind, 1374
unpack_voice_OriginateUSSD, 1374
unpack_voice_SLQSOriginateUSSD, 1374
unpack_voice_SLQSVoiceALSSelectLine, 1375
unpack_voice_SLQSVoiceALSSetLineSwitching, 1375
unpack_voice_SLQSVoiceAnswerCall, 1375
unpack_voice_SLQSVoiceBindSubscription, 1376
unpack_voice_SLQSVoiceBurstDTMF, 1376
unpack_voice_SLQSVoiceDialCall, 1376
unpack_voice_SLQSVoiceEndCall, 1377
unpack_voice_SLQSVoiceGetAllCallInfo, 1377
unpack_voice_SLQSVoiceGetCLIP, 1379
unpack_voice_SLQSVoiceGetCLIR, 1379
unpack_voice_SLQSVoiceGetCNAP, 1379
unpack_voice_SLQSVoiceGetCOLP, 1380
unpack_voice_SLQSVoiceGetCOLR, 1380
unpack_voice_SLQSVoiceGetCallBarring, 1377
unpack_voice_SLQSVoiceGetCallForwarding-Status, 1378
unpack_voice_SLQSVoiceGetCallInfo, 1378
unpack_voice_SLQSVoiceGetCallWaiting, 1378
unpack_voice_SLQSVoiceGetConfig, 1380
unpack_voice_SLQSVoiceIndicationRegister, 1381
unpack_voice_SLQSVoiceManageCalls, 1381
unpack_voice_SLQSVoiceOrigUSSDNoWait, 1381
unpack_voice_SLQSVoiceSUPSCallback_ind, 1384
unpack_voice_SLQSVoiceSendFlash, 1382
unpack_voice_SLQSVoiceSetCallBarringPassword, 1382
unpack_voice_SLQSVoiceSetConfig, 1382
unpack_voice_SLQSVoiceSetPreferredPrivacy, 1383
unpack_voice_SLQSVoiceSetSUPSService, 1383
unpack_voice_SLQSVoiceStartContDTMF, 1383
unpack_voice_SLQSVoiceStopContDTMF, 1384
unpack_voice_SUPSNotificationCallback_ind, 1384
unpack_voice_USSDNotificationCallback_ind, 1385
unpack_voice_VoiceInfoRecCallback_ind, 1385
unpack_voice_allCallStatusCallback_ind, 1372
unpack_voice_voicePrivacyChangeCallback_ind, 1385
voice_CLIPResp, 951
ActiveStatus, 951
ProvisionStatus, 951
voice_CLIRResp, 951
ActiveStatus, 952
ProvisionStatus, 952
voice_CNAPResp, 952
ActiveStatus, 953
ProvisionStatus, 953
voice_COLPResp, 953
ActiveStatus, 953
ProvisionStatus, 953
voice_COLRResp, 953
ActiveStatus, 954
ProvisionStatus, 954
voice_CUGInfo, 956
CUGIndex, 956
SuppOA, 956
SuppPrefCUG, 956
voice_DTMFInfo, 958
callID, 958
DTMFEvent, 958
digitBuff, 958
digitCnt, 958
voice_DTMFLengths, 958
DTMFInterdigitInterval, 959

DTMFPulseWidth, 959
voice_ECTNum, 959
ECTCallState, 960
number, 960
presentationInd, 960
voice_NSSAudioCtrl, 965
downLink, 965
upLink, 965
voice_SUPSInfo, 974
isModByCC, 974
svcType, 974
voice_USSDNotificationNetworkInfo, 974
networkInfo, 975
tlvPresent, 975
voice_USSInfo, 975
ussDCS, 975
ussData, 975
ussLen, 975
voice_UUSInfo, 975
UUSData, 976
UUSDatalen, 976
UUSDcs, 976
UUSType, 976
voice_airTimer, 928
airTimerValue, 929
namID, 929
voice_allCallsAlphaIDInfo, 929
AlphaIDInfo, 929
callID, 929
voice_allCallsDiagInfo, 929
callID, 930
DiagInfo, 930
voice_allCallsUUSInfo, 930
callID, 930
uusInfo, 930
voice_alphaIDInfo, 930
alphaDcs, 931
alphaLen, 931
alphaText, 931
voice_arrAlertingPattern, 931
alertingPattern, 932
callID, 932
numInstances, 932
voice_arrAlertingType, 932
AlertingType, 933
callID, 933
numInstances, 933
voice_arrAlphaID, 933
allCallsAlphaIDInfoArr, 933
numInstances, 933
voice_arrCallEndReason, 934
callEndReason, 934
callID, 934
numInstances, 935
voice_arrCallInfo, 935
getAllCallInfo, 935
numInstances, 935
voice_arrCalledPartyNum, 933
CalledPartyNum, 934
numInstances, 934
voice_arrConnectPartyNum, 935
ConnectedPartyNum, 936
numInstances, 936
voice_arrDiagInfo, 936
DiagInfo, 936
numInstances, 936
voice_arrRedirPartyNum, 936
numInstances, 937
RedirPartyNum, 937
voice_arrRemotePartyName, 937
GetAllCallRmtPtyName, 937
numInstances, 937
voice_arrRemotePartyNum, 937
numInstances, 938
RmtPtyNum, 938
voice_arrSvcOption, 938
callID, 938
numInstances, 938
srvOption, 938
voice_arrUUSInfo, 938
AllCallsUUSInfo, 939
numInstances, 939
voice_burstDTMFInfo, 939
digitCnt, 940
pCallID, 940
pDigitBuff, 940
voice_callFWExtInfo, 944
noReplyTimer, 945
numLen, 945
numPlan, 946
numType, 946
number, 945
PI, 946
SI, 946
SvcClass, 946
SvcStatus, 946
voice_callFWInfo, 946
noReplyTimer, 946
numLen, 947
number, 946
SvcClass, 947
SvcStatus, 947
voice_callFwdTypeAndPlan, 943
numberPlan, 944
numberType, 944
voice_callInfo, 947
callID, 948
callState, 948
callType, 948
direction, 948
mode, 948
voice_calledPartyInfo, 940
numLen, 941
numPlan, 941
numType, 941
number, 941

PI, 941
SI, 941
voice_calledPartySubAdd, 941
extBit, 942
oddEvenInd, 942
subAddr, 942
subAddrLen, 942
subAddrType, 942
voice_callerIDInfo, 942
callerID, 943
callerIDLen, 943
PI, 943
voice_callingPartyInfo, 948
numLen, 950
numPlan, 950
numType, 950
number, 950
PI, 950
SI, 950
voice_ccSUPSType, 950
reason, 950
svcType, 951
voice_connectNumInfo, 954
callerID, 955
callerIDLen, 955
numPlan, 955
numPresInd, 955
numType, 956
screeningInd, 956
voice_curAMRConfig, 956
gsmAmrStat, 957
wcdmaAmrStat, 957
voice_diagInfo, 957
diagInfoLen, 957
diagnosticInfo, 957
voice_extDispRecInfo, 960
dispType, 960
extDispInfo, 960
extDispInfoLen, 960
voice_getAllCallInformation, 961
ALS, 961
CallInfo, 961
isEmpty, 961
voice_getAllCallRmtPtyName, 961
callID, 962
RemotePartyName, 962
voice_getAllCallRmtPtyNum, 962
callID, 962
RemotePartyNum, 962
voice_getCallFWExtInfo, 962
CallFWExtInfo, 963
numInstances, 963
voice_getCallFWInfo, 963
CallFWInfo, 963
numInstances, 963
voice_lineCtrlInfo, 963
polarityIncluded, 964
pwrDenialTime, 964
revPolarity, 964
toggleMode, 964
voice_newPwdData, 964
newPwd, 965
newPwdAgain, 965
voice_peerNumberInfo, 965
callID, 966
numLen, 967
numPI, 967
numPlan, 967
numSI, 967
numType, 967
number, 967
voice_prefVoiceSO, 967
evrcCapability, 969
homeOrigVoiceSO, 969
homePageVoiceSO, 969
namID, 969
roamOrigVoiceSO, 969
voice_redirNumInfo, 969
numLen, 970
numPlan, 970
numType, 970
number, 970
PI, 970
reason, 970
SI, 970
voice_remotePartyName, 971
callerName, 971
codingScheme, 971
nameLen, 971
namePI, 971
voice_remotePartyNum, 971
numLen, 972
presentationInd, 972
remPartyNumber, 972
voice_roamTimer, 972
namID, 973
roamTimerValue, 973
voice_signalInfo, 973
alertPitch, 973
signal, 973
signalType, 973
VoiceNumber
unpack_dms_GetVoiceNumber_t, 580
voiceNumberSize
unpack_dms_GetVoiceNumber_t, 580
voicePrivacy
unpack_voice_voicePrivacyChangeCallback_ind_t, 879
voiceSvc
pack_voice_SLQSVoiceSetSUPSService_t, 457
VoipRat
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t, 641
VoipRatVal
imsa_VoipRatInfo, 88
VoipService

unpack_imsa_SLQSImsaSvcStatusCallBack_ind_-t, 641
VoipServiceRat
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
VoipServiceStatus
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
VoipSvcStatus
imsa_VoipSvcStatusInfo, 88
VolValue
pack_audio_SLQSSetAudioVolTLBConfig_t, 300
VoltStat
dms_VoltageTlv, 60
VoltTlv
unpack_dms_SwiEventReportCallBack_ind_t, 610
Voltage
dms_VoltageTlv, 60
Volume
pack_audio_SLQSGetAudioVolTLBConfig_t, 296
pack_audio_SLQSSetAudioProfile_t, 299
pack_audio_SLQSSetAudioVolTLBConfig_t, 300
unpack_audio_SLQSGetAudioProfile_t, 550
unpack_swiaudio_SLQSGetM2MAudioProfile_t, 802
voteForInit
uim_registerRefresh, 540
VsServiceRat
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
VsServiceStatus
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
VtRat
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_-t, 641
VtRatVal
imsa_VtRatInfo, 89
VtService
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_-t, 641
VtServiceRat
unpack_imsa_SLQSGetIMSAServiceStatus_t, 637
VtServiceStatus
unpack_imsa_SLQSGetIMSAServiceStatus_t, 638
VtSvcStatus
imsa_VtSvcStatusInfo, 90

wAlertMsgLength
UnpackSwiAvmsEventReportConfig, 922
WCDMACellInfo
nas_lteWcdmaCellInfo, 209
WCDMAECIOThreshListLen
nas_WCDMAECIOThresh, 274
WCDMARSSIThreshListLen
nas_WCDMARSSIThresh, 276
WCDMASSInfo
unpack_nas_SLQSNasGetSigInfo_t, 722
wDescriptionLength
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
wErrorCode
UnpackSwiAvmsEventReportDataSessionStatus, 923
wHTTPStatus
UnpackSwiAvmsEventReportHTTPStatus, 924
wNameLength
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
WORD
SwiDataTypes.h, 1311
wSessionStatus
UnpackSwiAvmsEventReportNotification, 926
wUpdateCompeteStatus
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
wUserInputTimeout
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
UnpackSwiAvmsEventReportConfig, 922
UnpackSwiAvmsEventReportConnectionRequest, 922
wVersionLength
UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 921
wWamsChangeMask
UnpackSwiAvmsEventReportWAMSParam-Change, 928
wake_mask
pack_rms_SetSMSWake_t, 396
unpack_rms_GetSMSWake_t, 779
wcdmaAmrStat
voice_curAMRConfig, 957
wcdmaRRCState
nas_WCDMAInfoLTENeighborCell, 275
Wdisable
unpack_dms_SLQSDmsSwiGetPCInfo_t, 590
wds.h
QMI_LITE_WDS_CURRENT_CALL_DB_MASK, 1394
QMI_LITE_WDS_LAST_CALL_DB_MASK, 1394
wds.h, 1386
BYT_STAT_STAT_MASK, 1393
liteQmiDataBearerMasks, 1394
PACK_WDS_IPV4, 1393
PACK_WDS_IPV6, 1393
pack_wds_DHCPv4ClientLeaseChange, 1394
pack_wds_GetAutoconnect, 1394
pack_wds_GetByteTotals, 1395
pack_wds_GetConnectionRate, 1395
pack_wds_GetDataBearerTechnology, 1395
pack_wds_GetDefaultProfile, 1396
pack_wds_GetDefaultProfileNum, 1396
pack_wds_GetDormancyState, 1397
pack_wds_GetLastMobileIPError, 1397
pack_wds_GetMobileIP, 1397
pack_wds_GetMobileIPProfile, 1398
pack_wds_GetPacketStatistics, 1398
pack_wds_GetPacketStatus, 1399
pack_wds_GetSessionDuration, 1399

pack_wds_GetSessionState, 1400
pack_wds_RMSetTransferStatistics, 1400
pack_wds_SLQSCreateProfile, 1403
pack_wds_SLQSDeleteProfile, 1404
pack_wds_SLQSGet3GPPConfigItem, 1404
pack_wds_SLQSGetCurrDataSystemStat, 1405
pack_wds_SLQSGetCurrentChannelRate, 1405
pack_wds_SLQSGetDUNCallInfo, 1406
pack_wds_SLQSGetDataBearerTechnology, 1405
pack_wds_SLQSGetProfileSettings, 1406
pack_wds_SLQSGetRuntimeSettings, 1407
pack_wds_SLQSModifyProfile, 1407
pack_wds_SLQSResetPacketStatics, 1408
pack_wds_SLQSSGetDHCPv4ClientConfig, 1409
pack_wds_SLQSSGetLoopback, 1410
pack_wds_SLQSSSetDHCPv4ClientConfig, 1410
pack_wds_SLQSSSetLoopback, 1410
pack_wds_SLQSSet3GPPConfigItem, 1408
pack_wds_SLQSSetIPFamilyPreference, 1408
pack_wds_SLQSSetWdsEventCallback, 1409
pack_wds_SLQSStartDataSession, 1411
pack_wds_SLQSStopDataSession, 1411
pack_wds_SLQSWdsGoActive, 1412
pack_wds_SLQSWdsGoDormant, 1412
pack_wds_SLQSWdsSetEventReport, 1412
pack_wds_SLQSWdsSwiPDPRuntimeSettings, 1413
pack_wds_SetAutoconnect, 1400
pack_wds_SetDefaultProfile, 1401
pack_wds_SetDefaultProfileNum, 1401
pack_wds_SetMobileIP, 1402
pack_wds_SetMobileIPParameters, 1402
pack_wds_SetMobileIPProfile, 1402
pack_wds_SetMuxID, 1403
unpack_wds_DHCPv4ClientLease_ind, 1413
unpack_wds_DHCPv4ClientLeaseChange, 1414
unpack_wds_GetAutoconnect, 1414
unpack_wds_GetByteTotals, 1414
unpack_wds_GetConnectionRate, 1415
unpack_wds_GetDataBearerTechnology, 1415
unpack_wds_GetDefaultProfile, 1415
unpack_wds_GetDefaultProfileNum, 1416
unpack_wds_GetDormancyState, 1416
unpack_wds_GetLastMobileIPError, 1417
unpack_wds_GetMobileIP, 1417
unpack_wds_GetMobileIPProfile, 1417
unpack_wds_GetPacketStatistics, 1418
unpack_wds_GetPacketStatus, 1418
unpack_wds_GetSessionDuration, 1418
unpack_wds_GetSessionState, 1419
unpack_wds_RMSetTransferStatistics, 1419
unpack_wds_RMTransferStatistics_ind, 1419
unpack_wds_SLQSCreateProfile, 1422
unpack_wds_SLQSDUNCallInfoCallBack_ind, 1423
unpack_wds_SLQSDeleteProfile, 1422
unpack_wds_SLQSGet3GPPConfigItem, 1423
unpack_wds_SLQSGetCurrDataSystemStat, 1423
unpack_wds_SLQSGetCurrentChannelRate, 1424
unpack_wds_SLQSGetDUNCallInfo, 1424
unpack_wds_SLQSGetDataBearerTechnology, 1424
unpack_wds_SLQSGetProfileSettings, 1425
unpack_wds_SLQSGetRuntimeSettings, 1425
unpack_wds_SLQSModifyProfile, 1425
unpack_wds_SLQSResetPacketStatics, 1426
unpack_wds_SLQSSGetDHCPv4ClientConfig, 1427
unpack_wds_SLQSSGetLoopback, 1428
unpack_wds_SLQSSSetDHCPv4ClientConfig, 1428
unpack_wds_SLQSSSetLoopback, 1428
unpack_wds_SLQSSet3GPPConfigItem, 1426
unpack_wds_SLQSSetIPFamilyPreference, 1426
unpack_wds_SLQSSetPacketSrvStatusCallback, 1426
unpack_wds_SLQSSetWdsEventCallback, 1427
unpack_wds_SLQSSetWdsEventCallback_ind, 1427
unpack_wds_SLQSStartDataSession, 1429
unpack_wds_SLQSStopDataSession, 1429
unpack_wds_SLQSWdsGoActive, 1429
unpack_wds_SLQSWdsGoDormant, 1430
unpack_wds_SLQSWdsSetEventReport, 1430
unpack_wds_SLQSWdsSwiPDPRuntimeSettings, 1430
unpack_wds_SetAutoconnect, 1420
unpack_wds_SetDefaultProfile, 1420
unpack_wds_SetDefaultProfileNum, 1420
unpack_wds_SetMobileIP, 1421
unpack_wds_SetMobileIPParameters, 1421
unpack_wds_SetMobileIPProfile, 1421
unpack_wds_SetMuxID, 1422
UnpackQmiProfileInfo, 1393
wds_ChannelRateTlv, 977
ChannelRate, 977
TlvPresent, 977
wds_ConnStatusTlv, 978
MDMConnStatus, 978
TlvPresent, 978
wds_DHCPLeaseOptTlv, 980
numOpt, 981
optList, 981
optListData, 981
TlvPresent, 981
wds_DHCPLeaseStateTlv, 981
leaseState, 981
TlvPresent, 981
wds_DHCPOpt, 982
optCode, 982
optValLen, 982
pOptVal, 982
wds_DHCPProfileIdTlv, 982
profileId, 983
profileType, 983
TlvPresent, 983

wds_DHCPv4HWConfig, 983
chaddr, 983
chaddrLen, 983
hwType, 983
wds_DHCPv4Option, 983
optCode, 984
optVal, 984
optValLen, 984
wds_DHCPv4OptionList, 984
numOpt, 984
pOptList, 984
wds_DHCPv4ProfileId, 985
profileId, 985
profileType, 985
wds_DataBearTechTlv, 979
DataBearerTech, 979
TlvPresent, 979
wds_DataULongLongTlv, 980
TlvPresent, 980
ullData, 980
wds_DataULongTlv, 980
TlvPresent, 980
ulData, 980
wds_Domain, 985
domainLen, 985
domainName, 985
wds_DomainNameList, 986
domain, 986
numInstances, 986
wds_DormStatTlv, 986
DormancyStat, 986
TlvPresent, 987
wds_GPRSQoS, 987
delayClass, 987
meanThroughputClass, 987
peakThroughputClass, 987
precedenceClass, 987
reliabilityClass, 987
wds_IPV6AddressInfo, 988
IPAddressV6, 988
IPV6PrefixLen, 988
wds_IPV6GWAddressInfo, 988
gwAddressV6, 989
gwV6PrefixLen, 989
wds_IPv4AdTlv, 987
IPv4Addr, 988
TlvPresent, 988
wds_LastMdmCallEndRsnTlv, 989
CallEndReason, 989
TlvPresent, 989
wds_PCSCFFQDNAddress, 990
fqdnAddr, 990
fqdnLen, 990
wds_PCSCFFQDNAddressList, 990
numInstances, 990
pcsfFQDNAddress, 990
wds_PCSCFIPv4ServerAddressList, 991
numInstances, 991
pscsfIPv4Addr, 991
wds_ProfileIdentifier, 991
profileIndex, 992
profileType, 992
wds_RXBytesOKTlv, 992
RxByteOKCnt, 992
TlvPresent, 993
wds_TXBytesOKTlv, 994
TlvPresent, 994
TxByteOKCnt, 994
wds_TrStatInd, 993
statsMask, 994
statsPeriod, 994
wds_UMTSMinQoS, 994
deliveryErrSDU, 996
grntDownlinkBitrate, 996
grntUplinkBitrate, 997
maxDownlinkBitrate, 997
maxSDUSize, 997
maxUplinkBitrate, 997
qosDeliveryOrder, 997
resBerRatio, 997
sduErrorRatio, 997
trafficClass, 997
trafficPriority, 997
transferDelay, 997
wds_channelRate, 977
CurrChanRxRate, 977
CurrChanTxRate, 977
wds_currNetworkInfo, 978
NetworkType, 979
RATMask, 979
SOMask, 979
wds_profileInfo, 992
SlqsProfile3GPP, 992
SlqsProfile3GPP2, 992
wds_transferStatInd, 993
StatsMask, 993
StatsPeriod, 993
wdsDhcpv4HwConfig, 997
chaddr, 997
chaddrLen, 997
hwType, 997
wdsDhcpv4Option, 998
optCode, 998
optVal, 998
optValLen, 998
wdsDhcpv4OptionList, 998
numOpt, 998
pOptList, 999
wdsDhcpv4ProfileId, 999
profileId, 999
profileType, 999
word
swi_uint256_t, 516

xAxis
sensorData_t, 494
xferStatAvail

unpack_wds_SLQSSetWdsEventCallback_ind_t, 914
xid
pack_loc_SLQSLOCGetBestAvailPos_t, 334
pack_qmi_t, 394
unpack_qmi_t, 756
xtra_start_gps_minutes
unpack_pds_SLQSGetGPSStateInfo_t, 755
xtra_start_gps_week
unpack_pds_SLQSGetGPSStateInfo_t, 756
xtra_valid_duration_hours
unpack_pds_SLQSGetGPSStateInfo_t, 756

yAxis
sensorData_t, 494
year
nas_timeInfo, 267
nas_UniversalTime, 273

zAxis
sensorData_t, 494